University Library



A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Shen, Aili

Title:
A multi-faceted approach to document quality assessment

Date:
2020

Persistent Link:
https://hdl.handle.net/11343/258655

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.

A Multi-faceted Approach to Document Quality Assessment

Aili Shen

ORCID: 0000-0002-5816-3465

School of Computing and Information Systems

The University of Melbourne

Submitted in total fulfilment of the requirements of the degree of

Doctor of Philosophy

July 2020

Dedicated to my loving sister, brothers, and parents.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 100,000 words, including footnotes but excluding bibliographies, tables and equations and has fewer than 150 figures.

Aili Shen

July 2020

Abstract

Document quality assessment, due to its complexity and subjectivity, requires considering information from multiple sources and aspects, to capture quality indicators. Grammaticality, readability, stylistics, structure, correctness, and expertise depth reflect the quality of documents from different aspects, with varying importance across different domains.

Automatic quality assessment has obvious benefits in terms of time saving and tractability in contexts where the volume of documents is large. In the case of dynamic documents (possibly with multiple authors), such as in the case of Wikipedia, it is particularly pertinent, as any edit potentially has implications for the quality label of that document. In this thesis, we focusing on improving the performance of document quality assessment systems and measure the uncertainty of document quality assessment systems. This thesis addresses four research questions: (1) How can we capture visual features not present in the document text, such as images and visual layout, to enhance representations learned from text content? (2) How can we make use of hand-crafted features widely adopted in traditional machine learning approaches in the context of neural networks, to generate a more accurate document quality assessment system? (3) How can we model the inherent subjectivity of quality assessment in evaluating the performance of quality assessment systems? and (4) Can a quality assessment system detect whether there are intruder sentences in documents and identify the span of any such intruder sentences, given that they interrupt the coherence of documents, thereby lowering their quality?

To address the first research question, we propose to use Inception V3 (Szegedy et al., 2016), a widely used visual model in computer vision, to capture visual features from visual renderings of documents, based on the observation that visual renderings of documents

can capture these visual features. Inception V3 compares favourably to textual-based models over the Wikipedia and academic paper reviewing datasets. We further propose a joint model to predict document quality by combining visual and textual features. We observe further improvements over both Wikipedia and academic paper reviewing datasets, indicating complementary between visual and textual features, and the general applicability of our proposed method.

Next, we propose two methods to enhance the capacity of neural models in predicting the quality of documents by utilising hand-crafted features. In the first method, we propose to concatenate hand-crafted features with neural learned high-level representations, assuming that neural model-learned features may not have captured all the information carried by these hand-crafted features. The second method, on the other hand, utilises hand-crafted features to guide neural model learning by explicitly attending to feature indicators when learning the relationship between the input and target variables, rather than simply concatenating hand-crafted features. Experimental results demonstrate the superiority of our proposed methods over baselines.

To imitate people's disagreement over the inherently subjective task of document quality assessment, we propose to measure the uncertainty in document quality predictions. We investigate two methods: Gaussian processes (GPs) (Rasmussen and Williams, 2006) and random forests (RFs) (Breiman, 2001), which provide not only a prediction of the document quality but also the uncertainty over their predictions. We also propose an asymmetric cost, considering the prediction uncertainty, which is used to measure the performance of two methods in the scenario, where decision-making processes based on model predictions can lead to different costs.

Lastly, we propose a new task of detecting whether there is an intruder sentence in a document, generated by replacing an original sentence with a similar sentence from a second document. Existing datasets in coherence detection are not suitable for our task as they are either too small for training current data-hungry models on or do not specify the span of incoherent text. To benchmark model performance over this task, we construct a large-scale dataset consisting of documents from English Wikipedia and

CNN news articles. Experimental results show that pre-trained language models which incorporate larger document contexts in pretraining perform remarkably well in-domain, but experience a substantial drop cross-domain. In follow-up analysis based on human annotations, substantial divergences from human intuitions were observed, pointing to limitations in their ability to model document coherence. Further results over a linguistic probe dataset show that pre-trained models fail to identify some linguistic characteristics that affect document coherence, suggesting room to improve for them to truly capture document coherence, and motivating the construction of a dataset with intruder text at the intra-sentential level.

Citations to Previously Published Work

Large portions of Chapter 3 have appeared in the following papers:

> Aili Shen, Bahar Salehi, Timothy Baldwin and Jianzhong Qi. 2019. A Joint Model for Multimodal Document Quality Assessment. In *Proceedings of the 18th Joint Conference on Digital Libraries (JCDL 2019)*, pages 107–110, Urbana-Champaign, USA.
>
> Aili Shen, Bahar Salehi, Jianzhong Qi and Timothy Baldwin. 2020. A General Approach to Multimodal Document Quality Assessment. *Journal of Artificial Intelligence Research (JAIR)* (accepted in June 2020).

Large portions of Chapter 4 have appeared in the following papers:

> Aili Shen, Jianzhong Qi and Timothy Baldwin. 2017. A Hybrid Model for Quality Assessment of Wikipedia Articles. In *Proceedings of the 2017 Australasian Language Technology Workshop (ALTW 2017)*, pages 43–52, Brisbane, Australia.
>
> Aili Shen, Bahar Salehi, Jianzhong Qi and Timothy Baldwin. 2019. Feature-guided Neural Model Training for Supervised Document Representation Learning. In *Proceedings of the 2019 Australasian Language Technology Workshop (ALTW 2019)*, pages 47–51, Sydney, Australia.

Large portions of Chapter 5 have appeared in the following paper:

> Aili Shen, Daniel Beck, Bahar Salehi, Jianzhong Qi and Timothy Baldwin. 2019. Modelling Uncertainty in Collaborative Document Quality Assessment. In

Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pages 191–201, Hong Kong, China.

Large portions of Chapter 6 have appeared in the following paper:

Aili Shen, Meladel Mistica, Bahar Salehi, Hang Li, Timothy Baldwin and Jianzhong Qi. 2020. Evaluating Document Coherence Modelling. In *Transactions of the Association for Computational Linguistics (TACL)* (under review).

Acknowledgements

Firstly, I would like to express my deep gratitude to my supervisors Jianzhong Qi, Bahar Salehi, and Timothy Baldwin for their continuous support and encouragement of my PhD, for their great patience, professional knowledge in research, and friendly supervision style. I would like to thank them enormously for weekly meetings with me to discuss ideas and issues that arose during my research. I would feel energised and confident after each weekly meeting, making me enjoy the PhD journey more. I highly appreciate their attitude to treat me as a colleague not just a student, their willingness to hear my opinions, their patience to explain stuff whenever I had a different opinion. The lessons I have learned from them, be it academic or otherwise, will be a lifelong gift to me. I could not expect better supervisors during my PhD candidature.

I would like to extend my gratitude to the co-author of my work on uncertainty modelling (in Chapter 5), Daniel Beck, who helped me in understanding the workings of Gaussian Processes. I am also grateful for the effort of Meladel Mistica and Hang Li, either devoting their time or financial support, in constructing a large-scale dataset to benchmark our proposed task (in Chapter 6).

I feel very lucky and proud to be part of a friendly and talented NLP team, where I met brilliant, dedicated, and friendly colleagues. I have learned a lot from them with regard to both research and personal life. Thank you to all colleague students at my lab for the great company, support and fun. I have a pleasant and memorable PhD journey.

I would like to thank The University of Melbourne for covering the costs of my PhD studies and living allowance throughout the candidature. Without it, I would not be able to finish my PhD. I would like to acknowledge the travel support of the School of

Engineering, School of Computing and Information Systems, and Google PhD Travel Scholarship, making it possible to present my work to a broader audience.

My deep and sincere gratitude to my family: my parents and elder brothers and sister, for their continuous and unconditional love, help, and support. I am grateful to my sister for always being there for me whenever I'm in need of help. I am forever indebted to my parents for allowing me enormous freedom to choose a life I want to live.

Last but not least, I want to thank all cats in the world. They are so cute that they can always cheer me up even when I am down.

Table of contents

List of figures **xxiii**

List of tables **xxvii**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Aim and Research Questions 6

1.3 Contributions 8

1.4 Thesis Structure 10

2 Literature Review **15**

2.1 Task Definition 15

2.1.1 Quality Assessment of Wikipedia articles 16

2.1.2 Academic Paper Rating 17

2.1.3 Essay Scoring 19

2.1.4 Content Quality Assessment in cQA 21

2.1.5 Discussion 23

2.2 Task Modelling 26

2.2.1 Classification 26

2.2.2 Regression 26

2.2.3 Ordinal Regression 27

2.2.4 Ranking 27

2.2.5 Domain-specific Modelling 27

2.3 Evaluation Metrics . . . 28
2.3.1 Discrete Evaluation Metrics . . . 29
2.3.2 Rank-based Evaluation Metrics . . . 30
2.3.3 Continuous Evaluation Metrics . . . 32
2.3.4 Evaluation Metrics Adopted For Each Task . . . 33
2.4 Hand-crafted Features . . . 34
2.4.1 Quality Assessment of Wikipedia articles . . . 35
2.4.1.1 Article Internal Features . . . 35
2.4.1.2 Metadata Features . . . 38
2.4.1.3 Article-Editor Mutual Dependency Features . . . 39
2.4.2 Academic Paper Rating . . . 40
2.4.3 Essay Scoring . . . 42
2.4.3.1 Persuasiveness and Topic Relevance Features . . . 43
2.4.3.2 Organisation and Coherence Features . . . 44
2.4.3.3 Sentence Complexity and Word Usage Features . . . 47
2.4.3.4 Grammar and Mechanics Features . . . 48
2.4.4 Content Quality Assessment in cQA . . . 49
2.4.4.1 Question and Answer Features . . . 50
2.4.4.2 User Features . . . 58
2.4.4.3 Thread Level Features . . . 61
2.4.4.4 Usage Features . . . 64
2.4.5 Discussion . . . 65
2.5 End-to-End Models . . . 66
2.5.1 Quality Assessment of Wikipedia Articles . . . 66
2.5.2 Academic Paper Rating . . . 67
2.5.3 Essay Scoring . . . 68
2.5.4 Content Quality Assessment in cQA . . . 70
2.5.5 Discussion . . . 72
2.6 Document Coherence Measurement . . . 73

2.6.1 Task Definition and Modelling . . . 74
2.6.2 Evaluation Metrics . . . 77
2.6.3 Hand-crafted Features . . . 77
2.6.4 End-to-End Models . . . 79
2.7 Summary . . . 81

3 Multimodal Document Quality Assessment 83
3.1 Motivation . . . 84
3.2 Related Work . . . 87
3.2.1 Wikipedia Article Quality Assessment . . . 87
3.2.2 Academic Paper Rating . . . 88
3.2.3 Multimodal Document Processing . . . 89
3.3 The Proposed JOINT Model . . . 90
3.3.1 Visual Embedding Learning . . . 90
3.3.2 Textual Embedding Learning . . . 91
3.3.3 The Joint Model . . . 91
3.4 Experiments . . . 92
3.4.1 Experimental Setting . . . 93
3.4.2 Wikipedia . . . 94
3.4.2.1 Dataset . . . 94
3.4.2.2 Baseline Approaches . . . 96
3.4.2.3 Experimental Results . . . 97
3.4.3 arXiv . . . 99
3.4.3.1 Dataset . . . 99
3.4.3.2 Baseline Approaches . . . 100
3.4.3.3 Experimental Results . . . 100
3.4.4 CVPG . . . 101
3.4.4.1 Dataset . . . 101
3.4.4.2 Baseline Approaches . . . 101
3.4.4.3 Experimental Results . . . 102

3.4.5 Summary of Results across Datasets . . . 102
3.4.6 Analysis . . . 103
3.4.6.1 INCEPTION . . . 103
3.4.6.2 JOINT . . . 105
3.5 Model Interpretability . . . 107
3.5.1 Hand-crafted Features . . . 107
3.5.2 Experimental Settings . . . 109
3.5.3 Experimental Results . . . 110
3.5.4 Experiments Incorporating Hand-crafted Features . . . 112
3.6 Applicability to Other Domains . . . 112
3.7 Summary . . . 113

4 Enhancing Neural Models with Hand-crafted Features 115

4.1 Enhancing Neural Models via Concatenating Hand-crafted Features with Learned Representations . . . 116
4.1.1 Motivation . . . 116
4.1.2 Related Work . . . 117
4.1.3 The Proposed Hybrid Model . . . 119
4.1.3.1 High-level Representation . . . 119
4.1.3.2 Hand-crafted Features . . . 119
4.1.3.3 The Proposed Hybrid Model BI-LSTM$^+$ . . . 121
4.1.4 Experiments . . . 121
4.1.4.1 Experimental Setting . . . 121
4.1.4.2 Datasets . . . 122
4.1.4.3 Experimental Results . . . 123
4.1.4.4 Analysis . . . 123
4.1.4.5 Discussion . . . 125
4.1.5 Applicability to Other Domains . . . 126
4.2 Enhancing Neural Models via Feature-guiding . . . 127
4.2.1 Motivation . . . 127

4.2.2 Related Work . . . 128
4.2.3 Methodology . . . 128
4.2.4 Experiments on Quality Assessment of Wikipedia Articles . . . 130
4.2.4.1 Dataset . . . 131
4.2.4.2 Model Configuration . . . 131
4.2.4.3 Results . . . 133
4.2.4.4 Further Results . . . 133
4.2.5 Experiments on Online Petition Signature Prediction . . . 134
4.2.5.1 Dataset . . . 135
4.2.5.2 Model Configuration . . . 135
4.2.5.3 Results . . . 136
4.2.6 Applicability to Other Domains . . . 136
4.3 Summary . . . 137

5 Modelling Uncertainty in Collaborative Document Quality Assessment 139
5.1 Motivation . . . 140
5.2 Related Work . . . 142
5.3 Preliminaries . . . 143
5.3.1 Problem Definition . . . 143
5.3.2 Data Collection . . . 144
5.4 Methods . . . 145
5.4.1 Gaussian Processes . . . 147
5.4.2 Random Forests . . . 148
5.4.3 Features and Preprocessing . . . 148
5.5 Experimental Study . . . 148
5.5.1 Intrinsic Evaluation . . . 149
5.5.1.1 GP Settings . . . 149
5.5.1.2 RF Settings . . . 149
5.5.1.3 Evaluation Metrics . . . 149
5.5.1.4 Results . . . 150

5.5.1.5 Feature Analysis . . . 151
5.5.2 Point Estimate Comparison . . . 151
5.5.2.1 Experimental Settings . . . 151
5.5.2.2 Results . . . 152
5.5.3 Prediction of Secondary Labels . . . 152
5.5.3.1 Evaluation Metrics . . . 153
5.5.3.2 Results . . . 153
5.5.4 Case Study: Quality Prediction as Filtering for Manual Revision . . . 154
5.5.4.1 Discretized Distribution . . . 155
5.5.4.2 Asymmetric Loss Function . . . 156
5.5.4.3 Evaluation . . . 156
5.5.4.4 Results . . . 156
5.6 Applicability to Other Domains . . . 157
5.7 Summary . . . 157

6 Evaluating Document Coherence Modelling 159
6.1 Motivation . . . 160
6.2 Related Work . . . 162
6.2.1 Document Coherence Measurement . . . 162
6.2.2 Document Coherence Datasets . . . 164
6.2.3 Dataset Artefacts . . . 165
6.2.4 Linguistic Probes . . . 165
6.3 Dataset Construction . . . 166
6.3.1 Dataset Desiderata . . . 166
6.3.2 Data Sources . . . 167
6.3.3 Generating Candidate Positive Samples . . . 167
6.4 Dataset Analysis . . . 169
6.4.1 Statistics of the Dataset . . . 169
6.4.2 Types of Incoherence . . . 169
6.4.3 Evaluation Metrics . . . 170

6.4.4 Testing for Dataset Artefacts . . . 171
6.4.5 Human Verification . . . 171
6.5 Models . . . 173
6.6 Experiments . . . 174
6.6.1 Preliminary Results . . . 174
6.6.2 Results over the Existing Dataset . . . 177
6.6.3 Performance on Documents of Different Difficulty Levels . . . 178
6.6.4 Analysis over Documents with High Human Agreement . . . 179
6.6.5 Question Revisited . . . 180
6.7 Linguistic Probes . . . 181
6.7.1 Linguistic Probe Dataset Construction . . . 181
6.7.2 Experimental Results . . . 183
6.8 Summary . . . 184

7 Conclusions **187**
7.1 Research Question Revisited . . . 187
7.2 Future Directions . . . 190
7.2.1 Incorporating Discourse Knowledge . . . 190
7.2.2 End-to-End Models in a Generative-Discriminative Fashion . . . 191
7.2.3 Document Quality Assessment via Multitask Learning . . . 193
7.3 Summary . . . 194

References **195**

List of figures

1.1 Visual renderings of two example documents from Wikipedia (not intended to be readable). . . . 3
1.2 Visual renderings of two example documents from cQA (not intended to be readable). These are two answers to the question "How do I check what version of Python is running my script?". . . . 4
1.3 A screenshot of the "Warden Head Light" Talk page. Wikipedia Project *Lighthouses* assigns a B-class quality label to this article, while Wikipedia Project *Australia* assigns a Start-class quality label. . . . 5

2.1 A screenshot of a Wikipedia article with the Featured Article quality label, not intended to be readable. . . . 17
2.2 An example TOEFL essay with the highest score, in response to the given prompt, which is from the book entitled "The Official Guide to the TOEFL Test". . . . 20
2.3 An example of a cQA thread on Stack Overflow, where only one of 19 answers is shown in this figure. . . . 22

3.1 Visual renderings of two example Wikipedia documents with different quality labels (not intended to be readable). . . . 85
3.2 Illustration of the textual model (BI-LSTM), a component of our JOINT model. 92
3.3 Illustration of our proposed model (JOINT), which consists of textual and visual models. . . . 92

3.4 Heatmap overlapped onto screenshots of each Wikipedia quality class. Best viewed in color. 104
3.5 t-SNE scatter plot of Wikipedia article representations (representations from the penultimate layer of each model, based on 200 random samples from each quality class; best viewed in colour). 107

4.1 Performance of BI-LSTM and BI-LSTM$^+$ on the 60K dataset, varying the number of sentences. 124
4.2 Illustration of our proposed method (`Feature-guided`). Here: $\otimes$ denotes cosine similarity between the feature indicator and word embeddings; f_1 and f_2 denote BI-LSTM layers; f_3 denotes a dense layer; $\mathcal{Z}_1$ and $\mathcal{Z}_2$ denote sentence and document representations, respectively; $\boldsymbol{V}$ and $\boldsymbol{F}$ are the document input and feature indicators, respectively; and $\boldsymbol{y}$ is the target output. 130

5.1 A screenshot of the "Warden Head Light" Talk page. Wikipedia Project *Lighthouses* assigns a B-class quality label to this article, while Wikipedia Project *Australia* assigns a Start-class quality label. 141
5.2 Model application and decision-making procedure. 144
5.3 Feature importance values in the GP model (left) and RF model (right). A full description of all features can be found in Dang and Ignat (2016a) and Table 3.7. 151
5.4 Discretisation of a continuous predictive distribution. The shaded area shows the probability of quality label Start (-1.5). 155
5.5 Asymmetric risk with varying values of α (best viewed in colour). Here, GP_{con} and RF_{con} denote risk values achieved by GP and RF over *consistent* articles, respectively; GP_{inc} and RF_{inc} denote risk values achieved by GP and RF over *inconsistent* articles. 157

6.1 An excerpt of an incoherent document, with the "intruder" sentence indicated in **bold**. 161

6.2 Illustration of generating candidate incoherent documents. 167
6.3 Distribution of instances where different numbers of humans produce correct answers. Note that the red bars indicate distributions over all documents and the blue bars indicate distributions over incoherent documents. 172
6.4 ALBERT-xxLarge vs. humans. 178

7.1 Illustration of a simple discourse tree, from Hernault et al. (2010). 191

List of tables

2.1 Descriptions and examples of Wikipedia quality classes. 18

3.1 Statistics of the Wikipedia dataset used to benchmark our proposed model (JOINT). 95

3.2 Performance comparisons between our proposed model (JOINT) and baselines. The best result for each dataset is indicated in **bold**, and marked with "†" if it is significantly better than the second best result (based on a two-tailed Wilcoxon signed-rank test; $p < 0.05$). The results of BENCHMARK on the arXiv dataset are from the original paper, where the standard deviation values were not reported. All neural models except for $\text{INCEPTION}_{\text{FIXED}}$ have larger standard deviation values on arXiv than Wikipedia and CVPG, which can be explained by the small size of the arXiv test set. 98

3.3 Statistics of the arXiv dataset. "Accepted" indicates the proportion of accepted papers in the given subject. 99

3.4 Statistics of the CVPG dataset, in terms of the number of documents that were accepted or rejected by a conference. 101

3.5 Confusion matrix of the JOINT model on Wikipedia. Rows are the actual quality classes and columns are the predicted quality classes. The diagonal (gray cells) indicates correct predictions. 105

3.6 Precision ("$\mathscr{P}$"), recall ("$\mathscr{R}$"), and F1 ("$\mathscr{F}_{\beta=1}$") of BI-LSTM, INCEPTION, and JOINT on Wikipedia. 106

3.7 A list of hand-crafted features explored in the multi-task setting. Here, the text in parentheses (e.g., "*Length*") is a descriptor for each feature, which will be used hereafter. 108

3.8 Pearson correlation (r) for each predicted feature, for BI-LSTM and INCEPTION. The best result for each feature is indicated in bold, and marked with "†" if the difference is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$). 111

3.9 Performance comparisons between BI-LSTM and INCEPTION over the Wikipedia dataset in the multi-task setting. Results achieved by BI-LSTM and INCEPTION in the multi-task setting which are better than those for the corresponding single task setting are indicated in bold, and marked with "†" if the difference is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$). 111

3.10 Performance comparisons between BI-LSTM, INCEPTION, and JOINT after incorporating hand-crafted features. BI-LSTM$^+$, INCEPTION$^+$, and JOINT$^+$ are the models that concatenate hand-crafted features with features learned by BI-LSTM, INCEPTION, and JOINT. Results achieved by BI-LSTM$^+$, INCEPTION$^+$, and JOINT$^+$ are indicated in **bold**, if they are better than those achieved by BI-LSTM, INCEPTION, and JOINT, respectively, and additionally with "†" if the difference is statistically better (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$). 113

4.1 A list of hand-crafted features concatenated with learned representations. . 120

4.2 Statistics of Wikipedia datasets used to benchmark our proposed model (BI-LSTM$^+$). 122

4.3 Performance comparisons between our proposed model (BI-LSTM$^+$) and baselines. 123

4.4 Accuracy (%) over Wikipedia dataset for our proposed method (`Feature-guided`) and various baselines. The best result is in **bold**, and marked with "†" if the improvement is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$). 133
4.5 Accuracy (%) over Wikipedia dataset for our proposed method `Feature-guided` with different pooling strategies. The best result is in **bold**. 134
4.6 Performance of our proposed method (`Feature-guided`) over online petitions. The best result is indicated in **bold**, and marked with "†" if the improvement is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$). 136

5.1 Statistics of the Wikipedia dataset consisting of *consistent* and *inconsistent* articles, which are used to benchmark models' performance in providing uncertainty estimates of their predictions. 146
5.2 Performance comparisons between RF and GPs with different kernels in predicting the primary quality label. GP_{arc0}, GP_{arc1}, and GP_{arc2} denote GP using an Arccosine kernel with orders of 0, 1, and 2. $GP_{Matérn32}$, $GP_{Matérn52}$, and GP_{RQ} denote GP using Matérn32, Matérn52, and RQ, respectively. The best result (the lower the better) is indicated in **bold**, and marked with "†" if the improvement is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$). 150
5.3 Point estimate comparison results. 152
5.4 Performance comparisons between RF and GP with the RBF kernel in predicting the secondary quality labels. 154

6.1 Dataset statistics for INSteD. Numbers in parentheses are percentages of incoherent documents. 169
6.2 Types of document incoherence in Wikipedia. Text in bold indicates the intruder sentence. 170

6.3 Experimental results over Wikipedia and CNN, in both in-domain and cross-domain settings. 175
6.4 Acc for the dataset of Chen et al. (2019). 177
6.5 Statistics over documents where all 5 humans agree. 179
6.6 Results over documents annotated consistently by all 5 annotators, where annotations can be the same as or different from gold-standard. 179
6.7 Results over language probes in incoherent Wikipedia test documents. BERT-Large$_{CNN}$, XLNet-Large$_{CNN}$, RoBERTa-Large$_{CNN}$, and ALBERT-xxLarge$_{CNN}$ are trained over CNN, while BERT-LARGE, XLNET-LARGE, RoBERTa-Large, and ALBERT-xxLarge are trained over Wikipedia. Here, F_1 is over the original incoherent documents (excluding linguistic probes), and ΔF_1 indicates the absolute performance difference resulting from incorporating linguistic probes. 183

Chapter 1

Introduction

1.1 Background and Motivation

With the advent of Web 2.0, it is much easier for communities to write and distribute documents, such as web pages in Wikipedia,[1] questions and answers in community question answering (cQA), and academic papers. However, the quality of such documents varies greatly, ranging from well-written documents, which are easy to read, conform to community policies, and convey useful information that users are looking for, to poorly written documents, which are difficult to read, disobey community guidelines, and are unable to satisfy users' information needs. Additionally, there are cases where the quality of a document is worsened maliciously, such as vandalism in Wikipedia and inserting commercial advertisements into a high quality document, resulting in incorrect and incoherent documents. Documents of high quality give more confidence to users to trust the documents, while those of low quality need to be identified and improved. For example, users can confidently trust high quality answers to solve their questions in Stack Overflow;[2] and Wikipedia contributors can further improve the article when notified of the low quality of an article. In this thesis, we focus on evaluating a single document in isolation, different from document summary evaluation (Radev et al., 2003,

[1] https://www.wikipedia.org/
[2] https://stackoverflow.com/

Zhang et al., 2008, Campr and Ježek, 2015) and machine translation evaluation (Papineni et al., 2002, Han, 2016), where the quality of document is based on the content overlap between summaries/translations and human references rather than the properties of summaries/translations. On the other hand, quality estimation (Specia et al., 2010), which is orthogonal to machine translation evaluation, evaluates the quality of a translation for a given document, without access to reference translation(s). Translation quality estimation evaluates the quality of machine-generated documents, while our focus in this thesis is on evaluating human-written documents.

Although there is no universal agreement on which factors define document quality, we consider the following six dimensions: (1) grammaticality, (2) readability, (3) stylistics, (4) structure, (5) correctness, and (6) expertise depth. Grammaticality indicates whether a sentence follows the rules and constraints of a given language. For example, *revolutionary new ideas appear infrequently* is grammatical in English, while *ideas green furiously colourless sleep* is ungrammatical (Briscoe, 2011). Readability reflects the usage of language and comprehension difficulty of the document. For example, Smog index (McLaughlin, 1969) measures the comprehension difficulty by estimating years of formal education a person needs to understand the text. Writing style reflects the choice of words, sentence structure, and paragraph structure used to convey information. For example, the writing style of *These are the times that try men's souls.* is different from that of *These are trying times for men's souls.* (Strunk, 2007), although both sentences convey the same idea. Structure describes the organisation of a document into sections, paragraphs, and sentences, which are connected with appropriate discourse relationships. Correctness indicates that information provided in the document should be accurate and free of errors.[3] Depth of expertise requires the document to provide expertise level content to satisfy users' information needs. In summary, the first four dimensions measure how well the document is written and the last two dimensions capture how much useful information the document provides. Although all the six dimensions we have mentioned are important factors

[3]Content-density detection (Yang and Nenkova, 2017) is relevant to correctness measurement, in evaluating whether the lead section of a news document provides important factual information in a direct and succinct manner.



A high-quality article



A low-quality article

Figure 1.1 Visual renderings of two example documents from Wikipedia (not intended to be readable).

affecting the quality of documents, measuring correctness and expertise depth requires complex information, such as commonsense, knowledge base, and expertise background, which may not be present in the document of interest. In this thesis, correctness and expertise depth are not automatically measured, and are left to future work.

There is also prior work on assessing other dimensions of documents. Louis (2013) has measured three individual dimensions of document quality: content, organisation, and reader interest. Content is measured via specificity and verbosity, where specificity measures the amount of detail in the documents and verbosity measures which details are important to include. Organisation is measured by the regularities of the intentional structure in the documents, where regularities of communicative goals are captured by syntax regularities between adjacent sentences. Reader interest is targeted at identifying interesting articles, as captured by the use of visual language, the use of sentiment and emotions, etc.. The usage of each dimension in predicting text quality is examined individually and together with features from other dimensions across documents in academic writing, science journalism, and automatically generated summaries. Louis (2013) focuses on the development of features to represent these three dimensions, and examination of how important these three dimensions are in document quality assessment. In comparison, our focus in this thesis is on assessing the quality of documents in an end-to-end fashion, and improving model performance in predicting the overall quality of documents.



A high-quality answer A low-quality answer

Figure 1.2 Visual renderings of two example documents from cQA (not intended to be readable). These are two answers to the question "How do I check what version of Python is running my script?".

People have different interpretations of document quality with regard to documents from different domains. For example, a Wikipedia article is considered to be a featured article (FA, the highest quality) if it is well-written, comprehensive, neutral, stable, and follows Wikipedia structure guidelines,[4] while an answer to a technical question in Stack Overflow[5] is considered to be good if it provides code examples and accompanying explanations (Nasehi et al., 2012). Figures 1.1 and 1.2 provide examples of high-quality documents and low-quality documents from Wikipedia and Stack Overflow, respectively. Another example is academic papers, where a good document is not only well-written, but also proposes novel ideas supported by theoretical or empirical evidence.

Furthermore, different users can disagree on the quality of a document, even given the same grading scheme. For example, as shown in Figure 1.3, the Wikipedia article *Warden Head Light*[6] is assigned different labels by different Wikipedia Projects:[7] B-class (in the green block) by Wikipedia Project *Lighthouses*, and Start (in the orange block) by Wikipedia Project *Australia*. Here, Start is a lower quality class than B-class according to Wikipedia grading scheme.

[4] https://en.wikipedia.org/wiki/Template:Grading_scheme
[5] https://stackoverflow.com/
[6] https://en.wikipedia.org/w/index.php?title=Warden_Head_Light&oldid=759074867
[7] A Wikipedia Project is a group of Wikipedia contributors who work together to improve Wikipedia articles that they are interested in.



Figure 1.3 A screenshot of the "Warden Head Light" Talk page. Wikipedia Project *Lighthouses* assigns a B-class quality label to this article, while Wikipedia Project *Australia* assigns a Start-class quality label.

An automatic document quality assessment system is needed for the following reasons. First, it is time-consuming and expensive to assess document quality manually, given the large volume of documents available online and the dynamics of documents. For example, there are only 32.3% among 13,525,636 articles assigned with a quality label in English Wikipedia,[8] and there are 1.8 edits per second across Wikipedia (Wikipedia Statistics). This high volume of documents and high frequency of edits makes it infeasible for humans to assess documents in a timely fashion. Second, an automatic quality assessment system can apply the same grading scheme to all documents, which reduces inconsistency and subjectivity in assessing document quality. As mentioned previously, people can disagree on the quality of a document, even given the same grading scheme. Automatic quality assessment, in contrast, can apply the learned grading scheme to all documents, reducing potential subjectivity of humans.[9] Third, an automatic quality assessment system can enable immediate feedback, which benefits both document contributors and document readers. For instance, readers can trust the article with more confidence if a Wikipedia article is assigned to a high quality class; or, Wikipedia contributors can be notified to

[8] Statistics is based on the dump released on 2017.10.20 (Wikipedia Dump).

[9] Note that the consistency that an automatic quality assessment system brings about can come at a cost, such as biases learned from the training dataset. Any real-world deployment of models should carefully test for such biases, and mitigate them as much as possible.

further improve the low-quality article. Another example application is to help the initial screening of academic paper submissions, which saves time for conference organisers, journal editors, and reviewers, which authors can also benefit via getting immediate feedback on their papers. Lastly, an automatic quality assessment system can detect whether there are injected commercial advertisements in documents of high quality and remove any such text, guaranteeing the high quality of documents automatically.

A potential application of automatic document assessment in other domains is that it can be incorporated with information retrieval to provide quality-guaranteed documents to end users (Jeon et al., 2006, Suryanto et al., 2009). For example, there may be many answers of different quality to a specific question asked in Stack Overflow, it is time-consuming for end users to go through all the answers. Assuming such a quality assessment system, users only need to look at answers of good quality.

1.2 Aim and Research Questions

Our primary aim is to improve automatic quality assessment systems and measure uncertainty (how uncertain a model is of its predictions) of these systems to capture subjectivity of document quality assessment. To fulfil this aim, we approach document quality assessment from the following four aspects.

1. How can we capture visual features not present in the text, to enhance representations learned from text content? Observing that the visual rendering of a document can capture visual features, such as images and visual layout, we develop models based on the visual rendering of a document, and use them as the evidence in predicting the quality of a document. If visual features are effective in predicting the quality of a document, can we boost the performance of a quality assessment system using both textual content and visual representations?

2. How can we make use of hand-crafted features widely adopted in traditional machine learning approaches in the context of neural networks, to generate a more

accurate document quality assessment system? Recently, neural models such as long short-term memory networks (LSTMs: Hochreiter and Schmidhuber (1997)) and convolutional neural networks (CNNs: Kim (2014), Kalchbrenner et al. (2014)) have achieved state-of-the-art/competitive results over a range of text classification/regression tasks, with the advantage of learning document representations automatically. Thus recent research has focused on improving the capacity of neural models, spurning hand-crafted features. However, hand-crafted features have been shown to have their utility in predicting the quality of Wikipedia articles (Blumenstock, 2008, Warncke-Wang et al., 2015, Dang and Ignat, 2016c). Can neural networks fully capture such knowledge? If not, how can we make use of hand-crafted features?

3. How can we take the inherent subjectivity of quality assessment into consideration in evaluating the performance of quality assessment systems? An ideal quality assessment should predict quality correctly with high confidence for documents where people agree in their assessment of quality; and it should predict quality correctly with low confidence for documents where people disagree. Previous work has mainly focused on correctly predicting the quality of documents, without paying attention to the subjectivity of this task. We explore methods to capture and measure such subjectivity.

4. Can a quality assessment system detect whether there are intruder sentences in documents and identify the span of any such intruder sentences, given that they interrupt the coherence of documents, thereby lowering their quality? A document quality assessment system should be able to identify the presence of such intruder text. Failing to identify intruder sentences indicates that the quality assessment system does not model document coherence, an important quality indicator. Thus, we construct a sentence-level intrusion detection dataset and investigate behaviours of existing models on this task, ranging from averaging word embeddings to state-of-the-art pre-trained transformer models.

1.3 Contributions

The contributions of this thesis are as follows:

1. We conduct a comprehensive literature review on document quality assessment in different domains, analysing and discussing task modelling and feature engineering approaches under one umbrella. In the literature review, we focus on quality assessment in the following four domains: quality assessment of Wikipedia articles, essay scoring, academic paper rating, and content quality assessment in cQA. For each domain, we first give the task definition and describe how previous studies have modelled the task, followed by an extensive summary of hand-crafted features and end-to-end neural models explored in the literature. In addition to summarising work on quality assessment of documents from different domains, we also review previous studies related to measuring document coherence, which is an important quality indicator. Lastly, we discuss modelling and evaluation in different task settings, and analyse the general applicability and specificity of hand-crafted features explored in the literature.

2. Based on our observation that the visual rendering of a document can capture visual features that are not present in the text, such as images and visual layout, we propose to use neural models to capture such visual features. To capture visual features in the visual renderings of documents, we turn our attention to a widely used visual model in computer vision Inception V3 model (Szegedy et al., 2016), which achieves superior performance than models learned from textual data alone over the Wikipedia dataset, and competitive results with models learned from textual data alone over the academic paper reviewing dataset. We further propose a joint model to predict document quality by combining visual and textual features. We observe further improvements over both the Wikipedia and academic paper reviewing datasets, indicating complementary between visual and textual features, and the general applicability of our proposed method. We also investigate what kinds of features a neural model learns by modelling document quality assessment

and feature learning as multi-task learning, providing interpretability of both the textual and visual models. Lacking visual renderings in existing document quality assessment datasets, we further construct a large-scale Wikipedia dataset consisting of textual data, visual renderings, and quality class labels, and supplement an existing academic paper reviewing dataset with corresponding visual representations.

3. We investigate methods to incorporate hand-crafted features into neural networks to improve the performance of quality assessment systems. In our work, we investigate two methods to combine hand-crafted features with neural networks: (1) simply concatenating hand-crafted features with learned representations; and (2) utilising hand-crafted features to guide neural model learning by explicitly attending to feature indicators when learning the relationship between the input and target variables, rather than simply concatenating hand-crafted features. Experimental results show that concatenating hand-crafted features with learned representations can improve the performance of quality assessment systems, and using hand-crafted features to guide network learning by explicitly attending to feature indicators further improves over simply concatenating hand-crafted features with representations.

4. To imitate people's disagreement over the inherently subjective task of document quality assessment, we propose to measure the uncertainty in document quality predictions. As there is no existing large-scale dataset containing multiple labels from different people in terms of document quality, we firstly construct a Wikipedia dataset containing quality labels from different Wikipedia Projects,[10] where quality labels assigned to a document can be the same or different. We investigate two methods, which provide not only a prediction of the document quality but also the uncertainty over their predictions, enabling the measurement of their uncertainty. To imitate a real world scenario where decision-making processes based on model predictions can lead to different cost, such as the time and human effort involved in manual reviewing if a low-quality article is predicted to be a featured article versus

[10] A Wikipedia Project is a group of Wikipedia contributors who work together to improve Wikipedia articles that they are interested in.

no manual reviewing in the opposite case, we propose an asymmetric cost, which takes the prediction uncertainty into consideration and is used to measure the performance of two methods in such a scenario.

5. To understand whether existing models fully understand the nature of document quality, we focus on document coherence, an important dimension of document quality. In this work, we propose a novel task to detect whether there is an intruder sentence in a document, and identify the position of any such sentence. Existing datasets used in coherence detection in previous studies are not suitable for our task as they are either too small for training current data-hungry models or do not specify the span of incoherent text. With the goal of advancing research in document coherence understanding (and thus document quality assessment) and benchmark how existing pre-trained transformer models perform over this task, we construct a large-scale dataset that consists of documents from Wikipedia and CNN news articles, where the Wikipedia subset is accompanied with human annotations via Mechanical Turk. Performance of pre-trained transformer models are evaluated at the document and sentence levels, at both in-domain and cross-domain settings. To further examine the capacity of the models to capture document coherence, we construct a linguistic probe dataset. We further investigate the performance of pre-trained transformer models at recognising intruder sentences within incoherent documents, with and without the addition of respective linguistic probes.

1.4 Thesis Structure

Chapter 2 provides a comprehensive survey on document quality assessment in multiple domains: Wikipedia, academic paper reviewing, content quality assessment in cQA, and essay scoring. For each domain, we first delineate the task, and summarise different approaches to modelling the task and evaluation metrics used in this domain. Then, we present an extensive summary of hand-crafted features, followed by a summary of neural models explored in this domain to predict document quality. After reviewing each

domain, we discuss the inherent differences of document quality assessment in these domains. Additionally, we compare and discuss task modelling and evaluation in different task settings, before analysing the general applicability and specificity of hand-crafted features considered in different domains.

In Chapter 3, we propose to utilise visual models to capture visual features that are not present in the document text, such as images and visual layout, with the visual rendering of a document as the input. The visual model based on the visual renderings of documents achieves superior performance over a textual model based on the textual content in predicting the quality of Wikipedia articles and competitive results with the textual model in predicting the acceptance of academic papers. We further propose a joint model to predict document quality, combining visual and textual features learned by the visual and textual models, respectively. Further improvements are observed across different datasets, indicating complementarity between visual and textual features. To have a better understanding of what kinds of features a visual and textual model have learned, we model document quality assessment and feature learning in a multi-task learning setting. Analysis of the experimental results shows that visual models are better at capturing structural features and textual models are better at learning readability scores, confirming our findings from the joint model experimental results: there is complementary between visual and textual features.

In Chapter 4, we turn our attention to combining hand-crafted features with neural networks, aiming to introduce expert knowledge captured by hand-crafted features into neural network learned representations/learning. The first method simply concatenates hand-crafted features with neural model learned representations. The second method utilises hand-crafted features to guide neural model learning by explicitly attending to feature indicators when learning the relationship between the input and target variables. They both improve the performance in predicting the quality of Wikipedia articles, demonstrating the utility of hand-crafted features in the context of neural networks. Comparing these two methods demonstrates that the improvement introduced by the first method is outperformed by the second method across four different neural network architec-

tures. The effectiveness of the second method is additionally verified on a different task: signature count prediction of online petitions.

In Chapter 5, we propose to measure the uncertainty of document quality assessment systems in making their predictions. We pay attention to two methods: Gaussian processes (GPs) and random forests (RFs), both of which can provide predictive distributions, making it feasible for us to measure their uncertainty over the predictions. Lacking a large-scale dataset consisting of multiple quality labels from different people, we first construct such a dataset, which enables the imitation of people's disagreement over the inherent subjectivity of document quality assessment. Experimental results show that both methods can yield competitive results in predicting the quality of Wikipedia articles, with the advantage of providing an estimate of uncertainty in their predictions. We also evaluate these two methods in an asymmetric scenario, where overestimates are penalised more severely than underestimates. To measure the performance of these two methods under the asymmetric scenario, we propose an asymmetric cost, which takes the prediction uncertainty into consideration. Experimental results reveal that GPs are a better option if overestimates are penalised more severely.

In Chapter 6, we propose the new task of detecting whether there is an intruder sentence in a document, generated by replacing an original sentence with a similar sentence from a second document. To benchmark model performance over this task, we construct a large-scale dataset consisting of documents from English Wikipedia and CNN news articles. Experimental results show that pre-trained language models which incorporate larger document contexts in pretraining perform remarkably well in-domain, but experience a substantial drop cross-domain. In follow-up analysis based on human annotations, substantial divergences from human intuitions were observed, pointing to limitations in their ability to model document coherence. Further results over a linguistic probe dataset show that pre-trained models fail to identify some linguistic characteristics that affect document coherence, suggesting room to improve for them to truly capture document coherence, and motivating the construction of a dataset with intruder text at the intra-sentential level.

Finally, Chapter 7 summarises findings of this thesis and discusses possible future research directions.

Chapter 2

Literature Review

This chapter reviews the literature on document quality assessment across four domains: quality assessment of Wikipedia articles, academic paper rating, essay scoring, and content quality assessment in cQA, in addition to document coherence. While the focus of this thesis is on assessing the quality of Wikipedia articles and academic papers, we also survey work on essay scoring and content quality assessment in cQA, as the hand-crafted features and neural models developed for these two domains are applicable to our tasks of interest. We describe the task definition in Section 2.1, task modelling specifics in Section 2.2, evaluation metrics in Section 2.3, hand-crafted features in Section 2.4, and end-to-end models in Section 2.5. For these sections, we first outline them in a domain-specific way, followed by discussion of similarities and differences between these four domains. We finish this chapter by summarising in Section 2.6 work on measuring document coherence, an important quality indicator, and the topic of Chapter 6.

2.1 Task Definition

For quality assessment of Wikipedia articles, the task is to assign quality labels to articles, where gold-standard quality labels are assigned by Wikipedia contributors. For academic paper rating, the task of academic paper rating is to predict whether a paper is of suitable standard to have been accepted by a target venue, and sometimes to predict whether a

paper fits into the scope of target venues, as judged by paper reviewers or journal editors. For essay scoring, the task is to assign scores to essays written by (second) language learners, where the quality of the essay is determined by teachers or test examiners. For content quality assessment in cQA, the task is to assign quality labels to content, mostly answers, posted in a community platform, where the quality of content is determined by the asker or users in the platform.

In this section, we provide task definitions of quality assessment of Wikipedia articles in Section 2.1.1, academic paper rating in Section 2.1.2, essay scoring in Section 2.1.3, and content quality assessment in cQA in Section 2.1.4. Then we present discussion of the task definitions across these four domains in Section 2.1.5.

2.1.1 Quality Assessment of Wikipedia articles

In Wikipedia, contributors can either edit a Wikipedia article or assign a quality label to an article, or both. Figure 2.1 gives an example of a Wikipedia article, assigned a quality label of Featured Article by Wikipedia contributors. A Wikipedia article with a Featured Article label should satisfy: (1) it is well-written, comprehensive, well-researched, neutral, and stable; (2) it follows the Wikipedia style guidelines, with a lead, appropriate structure, and citations wherever necessary (in the red block); (3) it has appropriate media (in the blue block) to support its content; and (4) it if of appropriate length, providing enough information about the main topic and avoiding unnecessary details.[1]

Officially, there are six quality classes of Wikipedia articles, which are ordered from high to low: *Featured Article* (FA), *Good Article* (GA), *B-class Article* (B), *C-class Article* (C), *Start Article* (Start), and *Stub Article* (Stub). A detailed description of these quality labels is presented in Table 2.1.

The task of quality assessment of Wikipedia articles is to assign one of the aforementioned six quality classes to a Wikipedia article or to differentiate high quality articles

[1]Louis and Nenkova (2014) have proposed to predict the appropriate length for documents based on content types present in a snippet of constant length, where the predicted length are compared against the actual length to measure document verbosity. Elsewhere, Yang and Nenkova (2017) have proposed to detect whether a news document is content-dense, i.e. the lead section of the document provides important factual information.

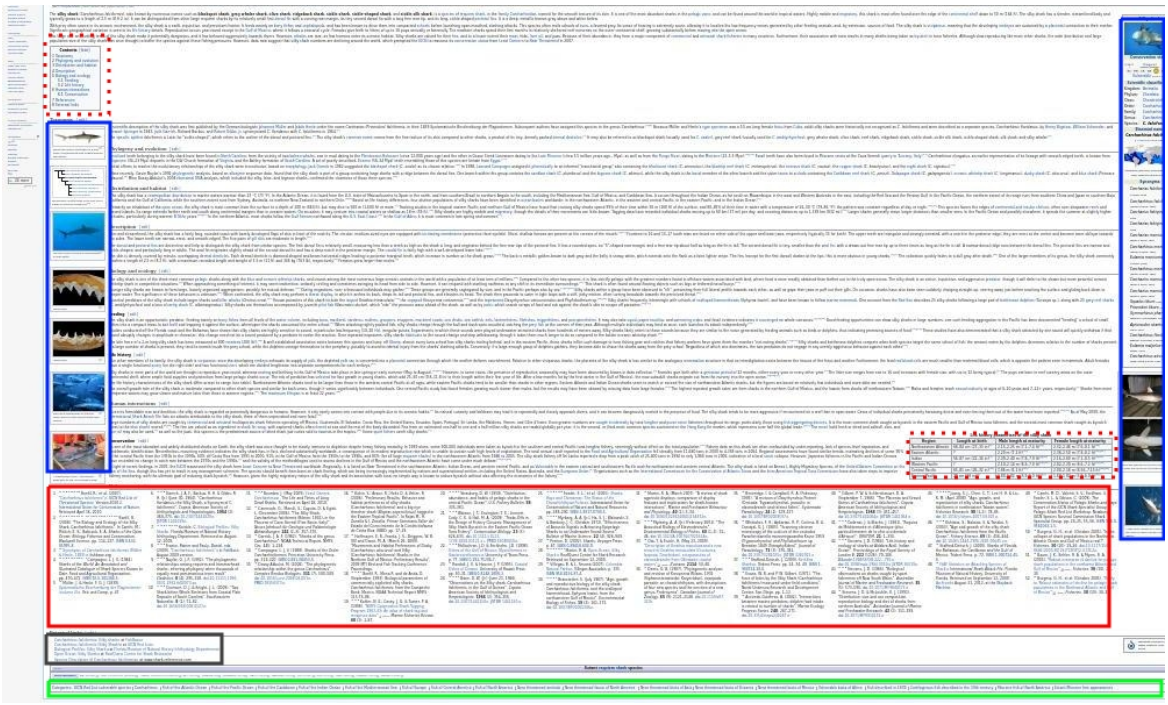


Figure 2.1 A screenshot of a Wikipedia article with the Featured Article quality label, not intended to be readable.

from low quality ones. Typically, this task is achieved by training a model to learn the relationship between the articles and their corresponding quality labels, either via extracting hand-crafted features from different sources, such as the article content and its revision history, or learning article representations via neural networks.

2.1.2 Academic Paper Rating

In academia, papers are generally written by multiple authors, and usually experience multiple rounds of editing coordinated among authors. After a submission is made, a handling reviewer (e.g., an associate editor or a meta-reviewer) assigns reviewers to the paper. In reviewing a paper, the reviewers' tendency to accept/reject a paper is usually reflected by the overall recommendation scores assigned by them, although scores are also given in terms of substance, clarity, appropriateness, impact, meaningful comparison, originality, and soundness. In-house discussions may be made among reviewers and

Class	Criteria	Article Example
FA	An FA article should pass an official peer review and be well-written, comprehensive, well-researched, stable, and with a neutral viewpoint. The article should have a lead section, appropriate structure, consistent citations, appropriate media, and proper length.	Elephant
GA	A GA article should pass an official peer review and be well-written, verifiable with no original research, stable, broad in its coverage, with a neutral viewpoint, and illustrated by images if possible .	YouTube
B	A B-class article is mostly completed and without major problems. The article should be properly referenced and reasonably well-written. The article should have a reasonable coverage over the topic, not contain obvious omissions or inaccuracies, and present content in an understandable way. The article should also have a defined structure and supporting materials where appropriate.	Sound
C	A C-class article is substantial, but still misses important content or contains many irrelevant materials. The article should have some references to reliable sources, but may still have significant problems or require substantial cleanup.	Comedy
Start	A Start article is developing, but is quite incomplete. A Start article might or might not cite adequate reliable sources.	JScript
Stub	A Stub article is a very basic description of the topic. All the lowest quality articles fall into this category.	Bee tree

Table 2.1 Descriptions and examples of Wikipedia quality classes.

scores can be adjusted after the discussion. Based on the review text and scores from reviewers, the handling reviewer makes a decision on whether to accept or reject the given submission, where review text and scores are also distributed to authors. For conferences such as *The Annual Meeting of the Association for Computational Linguistics*, authors can write a rebuttal to clarify unclear parts of the paper or answer questions asked by reviewers, to aid them in making the final decision. However, it is quite challenging for reviewers to review assigned papers in a timely manner, and for handling reviewers to manually determine whether a submission is within the scope of a given venue, given the rapid increase in the number of papers submitted to conferences and journals (Ghosal et al., 2019b). Thus, it is important to automatically assess whether a paper should be accepted/rejected and also whether a paper is within the scope of a target venue.

Formally, the task of academic paper rating is to predict whether a paper should be accepted/rejected, score a paper with regard to different aspects, or determine whether a submission is within the scope of a target venue.[2] Academic paper rating is achieved by learning relationships between the quality labels/scores and a variety of features, either hand-crafted or learned automatically. These features are extracted from multiple sources, such as the paper content, bibliographic information in the reference section, review text, and metadata of the authors.

2.1.3 Essay Scoring

In essay writing, (second) language learners are asked to write an essay in response to a given prompt (essay topic) or reading material, where the prompt is used to guide learners to write an essay responding to the topic stated in the prompt. Figure 2.2 gives an example of a TOEFL essay with a score of five (the highest score), in response to a given prompt. In this thesis, we consider a high-scored essay to have the following characteristics: (1) it focuses on the topic described in the prompt; (2) it is well-organised and developed; (3) it supports its points by clear reasoning; and (4) it demonstrates the author's ability to use a variety of sentence structures and words appropriately.

Multiple aspects of an essay are measured in scoring essays, such as persuasiveness and topic relevance, organisation and coherence, sentence complexity and word usage, and grammar and mechanics, as detailed in Section 2.4.3. However, it is worth mentioning that grading rubrics for essays in response to different prompts may emphasise different aspects. For example, the first task of set two in the automated student assessment prize (Phandi et al., 2015) assesses essays in terms of writing applications, such as ideas and content, organisation and coherence, and word usage and language skills, while the second task of set two assesses essays in terms of language conventions, such as capitalisation, punctuation, grade-level appropriate spelling, paragraphing, and sentence structure. Technical correctness is not evaluated in scoring essays, although Pearson

[2] There are various reasons for rejecting a paper, including methodological issues and relevance to a given venue. In this thesis, we do not directly model methodological and faithfulness issues in predicting the acceptance of papers.

Prompt
Do you agree or disagree with the following statement?
Always telling the truth is the most important consideration in any relationship.
Use specific reasons and examples to support your answer.
Response
There are certain considerations or factors that everyone takes into account in a relationship. People may look for honesty, altruism, understanding, loyalty, being thoughtful etc! Everyone would more or less wish that the person s/he is dealing with, has some of these virtues above. Putting them in an order according to their importance, however can be very subjective and relative.

When someone asks him/herself the question "What do I consider to be the most important thing in my relationship?" the answer depends on a lot of factors such as how his/her earlier relationships were.

After stating that everyone's opinion can be different about this, for me honesty, in other words, always telling the truth is the most important consideration in a relationship. Opposite of this is inarguably lying and if someone needs to lie, either s/he is hiding something or is afraid of telling me something.

In any relationship of mine, I would wish that first of all, the person I'm dealing with is honest. Even though s/he thinks that s/he did something wrong that I wouldn't like, s/he'd better tell me the truth and not lie about it. Later on if I find out about a lie or hear the truth from someone else, that'd be much more unpleasant. In that case how can I ever believe or trust that person again? How can I ever believe that this person has enough confidence in me to forgive him/her and carry on with the relationship from there. So if I cannot trust a person anymore, if the person doesn't think I can handle the truth, there's no point to continuing that relationship.

Although I would like to see altruistic, understanding, thoughtful and loyal behavior from people, an instance of the opposite of these behaviors would not upset me as much as dishonesty would. Among all the possible behaviors, dishonesty is the only one for me that terminates how I feel about a person's reliability. Therefore honesty would be my first concern and the most important consideration in a relationship.

Figure 2.2 An example TOEFL essay with the highest score, in response to the given prompt, which is from the book entitled "The Official Guide to the TOEFL Test".

Knowledge Technologies, producers of Intelligent Essay Assessor (Schreiner et al., 1997, Foltz et al., 1999, Landauer et al., 2003), claim that their system can assess "expository essays on factual topics", such as descriptions of the function of the heart (Murray, 1998). The main reason is that test takers are usually asked to state and defend their opinion with

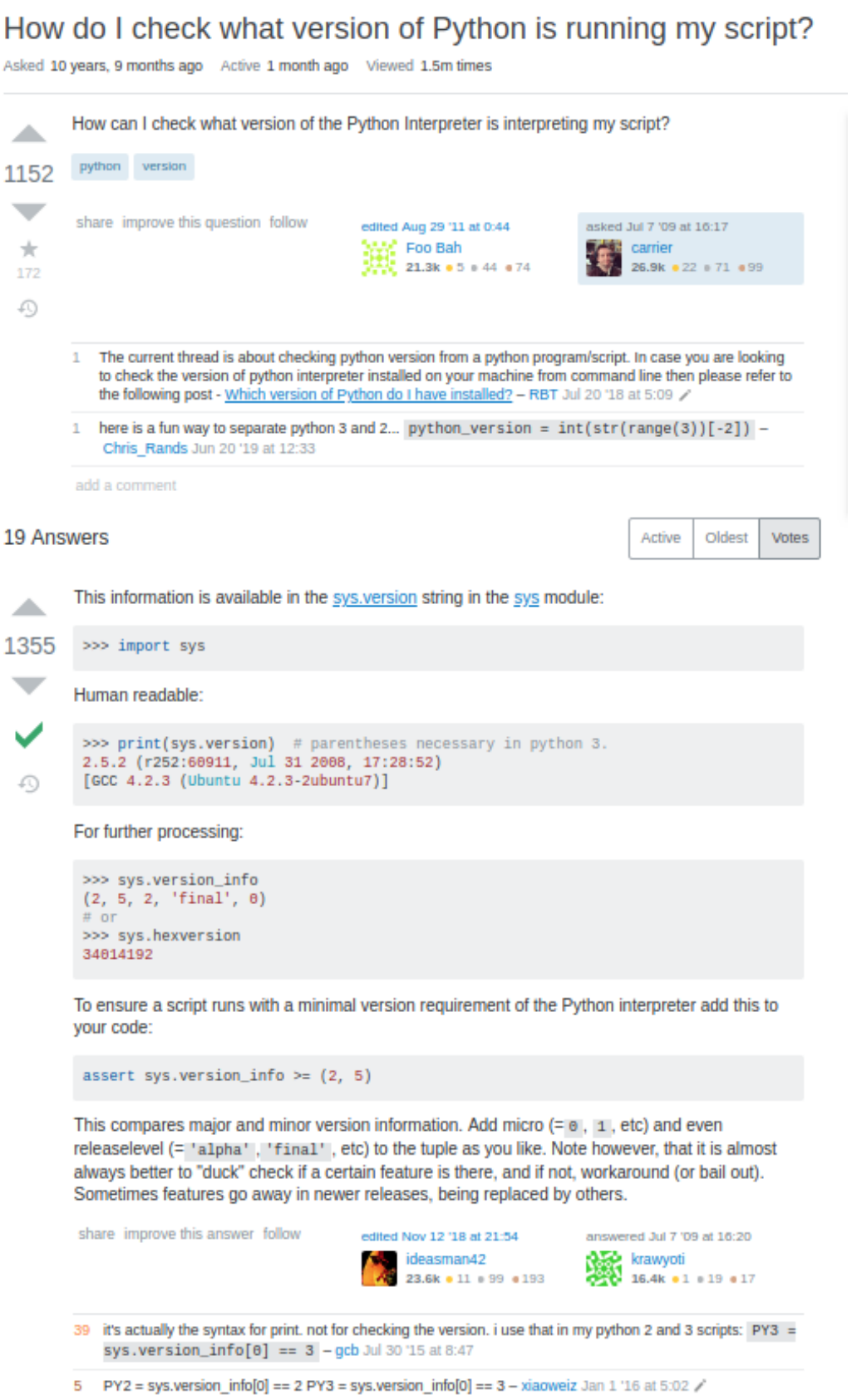
supporting materials, where the opinion is usually subjective and supporting evidence may come from personal experience, making it difficult to validate them in practice.

The task of essay scoring is to assign a score to an ungraded essay by learning the relationship between essays and their scores. Traditional methods represent essays by various types of features, such as persuasiveness and topic relevance, word usage and sentence complexity, and grammar and mechanics, while recent neural models learn essay representations automatically.

2.1.4 Content Quality Assessment in cQA

In cQA platforms, users can either ask questions to seek help from other users or provide answers to questions asked by other users. Figure 2.3 gives an example of a thread in Stack Overflow.[3] In addition to providing a question title, an asker also details the question in the body, accompanied with tags to make it easier for topic-related experts to locate the question. Tags of the question in the given example are *python* and *version*. In addition to answering the question, other registered users can upvote, downvote, and comment on the question and answers. For example, the number 1,152 in the left part of the question is the difference between the number of upvotes and downvotes that the question has received. Users are also allowed to vote on comments made to the question and answers. Besides upvotes and downvotes, registered users can also favourite a question, which is marked by a star. The ability to upvote, downvote, and favourite is associated with different reputation levels within a given community. In Stack Overflow, a user can upvote an answer if their reputation score is at least 15, and downvote an answer if their reputation score is at least 125. If the asker is satisfied with the provided answer, they can select the answer as the best answer, indicated by a green tick. Once the asker has selected the best answer or after a fixed period of time, a question is closed and no more answers are accepted.

[3] `https://stackoverflow.com/questions/1093322/how-do-i-check-what-version-of-python-is-running-my-script` accessed on 23rd of April 2020.



How do I check what version of Python is running my script?

Asked 10 years, 9 months ago Active 1 month ago Viewed 1.5m times

1152

How can I check what version of the Python Interpreter is interpreting my script?

python version

share improve this question follow

edited Aug 29 '11 at 0:44 Foo Bah 21.3k ● 5 ● 44 ● 74

asked Jul 7 '09 at 16:17 carrier 26.9k ● 22 ● 71 ● 99

172

1 The current thread is about checking python version from a python program/script. In case you are looking to check the version of python interpreter installed on your machine from command line then please refer to the following post - Which version of Python do I have installed? – RBT Jul 20 '18 at 5:09

1 here is a fun way to separate python 3 and 2... `python_version = int(str(range(3))[-2])` – Chris_Rands Jun 20 '19 at 12:33

add a comment

19 Answers

Active Oldest Votes

1355

This information is available in the sys.version string in the sys module:

```
>>> import sys
```

Human readable:

```
>>> print(sys.version)  # parentheses necessary in python 3.
2.5.2 (r252:60911, Jul 31 2008, 17:28:52)
[GCC 4.2.3 (Ubuntu 4.2.3-2ubuntu7)]
```

For further processing:

```
>>> sys.version_info
(2, 5, 2, 'final', 0)
# or
>>> sys.hexversion
34014192
```

To ensure a script runs with a minimal version requirement of the Python interpreter add this to your code:

```
assert sys.version_info >= (2, 5)
```

This compares major and minor version information. Add micro (=`0`, `1`, etc) and even releaselevel (=`'alpha'`, `'final'`, etc) to the tuple as you like. Note however, that it is almost always better to "duck" check if a certain feature is there, and if not, workaround (or bail out). Sometimes features go away in newer releases, being replaced by others.

share improve this answer follow

edited Nov 12 '18 at 21:54 ideasman42 23.6k ● 11 ● 99 ● 193

answered Jul 7 '09 at 16:20 krawyoti 16.4k ● 1 ● 19 ● 17

39 it's actually the syntax for print. not for checking the version. i use that in my python 2 and 3 scripts: `PY3 = sys.version_info[0] == 3` – gcb Jul 30 '15 at 8:47

5 PY2 = sys.version_info[0] == 2 PY3 = sys.version_info[0] == 3 – xiaoweiz Jan 1 '16 at 5:02

Figure 2.3 An example of a cQA thread on Stack Overflow, where only one of 19 answers is shown in this figure.

Although there are a large number of studies assessing the quality of content in cQA, (see Sections 2.4.4 and Section 2.5.4 for a detailed description), there is no universal definition of high-quality content in cQA. In this thesis, we consider a high-quality answer in cQA satisfying the following requirements: (1) the answer provides useful information to answer the question of interest; (2) the answer is easy to comprehend; (3) the answer contains references; (4) the answer uses appropriate words; and (5) the answer conforms to common writing practices and community policies.

The task of content quality assessment in cQA is to select the best answer, or to find high-quality answers, or to decide the rank between answers, achieved by learning the relationship between cQA content and quality labels (ranks). In general, the task of content quality assessment in cQA is to reproduce the ratings collected from a website, aiming to predict ratings for newly posted answers. Content in cQA can be represented by a variety of features capturing different aspects of the content quality or learned automatically via neural models. These features can be extracted from different sources, such as the content itself, the relationship between the question and answer, and usage information.

2.1.5 Discussion

There are similarities and differences among the document types targeted by these four tasks. In the following, we draw comparisons based on the following aspects: (1) article authors, (2) article examiners, (3) utility of an article, (3) characteristics of the document content, (4) input to a quality assessment system, and (5) output of a quality assessment system.

Documents examined in these tasks can be written by a single author or multiple authors. Specifically, Wikipedia articles are written and improved by volunteers who work collaboratively. Similarly, academic papers mostly are written by multiple authors, who work together in an integrated way to make contributions, and multiple iterations of editing among authors are made to improve the paper quality. Essays are written by (second) language learners, while questions and answers in cQA are written by askers and answers on a given platform.

The quality of documents examined by these tasks is determined by different groups. The quality of Wikipedia articles, except for FA and GA, is mainly determined by members of Wikipedia Projects, who are groups of Wikipedia contributors working together to improve Wikipedia articles that they are interested in. Being the highest and second highest quality of Wikipedia articles, FA and GA articles are assessed by independent editors: a panel for FA, and an editor for GA.[4] The acceptances of academic papers are determined collectively by reviewers recruited by action editors, area chairs, and program chairs. Essay scores are assigned by teachers or test examiners, according to a grading schema, which focuses on assessing a student's ability to write coherent and topic relevant essays by using appropriate words and sentence structures, free of grammatical and mechanistic errors. For content quality assessment in cQA, the quality of answers is dependent on askers (by selecting the best answers) or other community members (by voting).

The target audiences of these documents are different. For example, Wikipedia articles target users who are looking for general information about a topic or entity. Academic papers deliver new knowledge and methods to researchers from the same field. Essays written by (second) language learners aim to test the writer's level of mastery of a language. Answers in cQA target users who are looking for solutions to questions of interest.

There are differences among documents from different domains in terms of document length and language formality. In general, academic papers tend to be the longest, followed by Wikipedia articles, which in turn are longer than essays, with cQA documents being the shortest. Language formality also differs across domains, where the degree of language formality follows the same trend across the four document types as document length, with academic papers adopting the most formal language.

Information fed into a quality assessment system varies. For quality assessment of Wikipedia articles, information from the article itself, the revision history, and metadata of Wikipedia contributors, is used to assess the quality of a given Wikipedia article. For academic paper rating, only the papers themselves are fed into a quality assessment

[4] `https://en.wikipedia.org/wiki/Wikipedia:Content_assessment`

system. In essay scoring, the essays themselves and the associated prompt are the input to a quality assessment system. In assessing the quality of content in cQA, information from multiple sources, including question-answering pairs, metadata of questions, answers, and users, and usage information, is fed to a quality assessment system.

Outputs from a quality assessment system also vary across different domains. For quality assessment of Wikipedia articles, the predicted quality labels are from the six following predefined quality labels: FA, GA, B, C, Start, and Stub, ordered from high to low in terms of quality. For academic paper rating, outputs of an automatic paper rating system are acceptance categories (usually accept/reject), although outputs sometimes are aspect scores, measuring different aspects of papers. For essay scoring, outputs are a set of scores, within a predefined range. For content quality assessment in cQA, outputs of a quality assessment system are whether answers are the best answers or of good quality, or ranks of answers.

A general approach can be applied to assessing documents from each domain that we have discussed, ignoring the specifics of individual domains. The criteria for document quality in different domains have different weights over the six dimensions discussed in Chapter 1. For example, structure organisation and references can be more important in assessing the quality of Wikipedia articles, while grammaticality and readability can be more important in assessing student essays. A general approach, not tailored for a specific task, would not explicitly pay attention to specific aspects important to the task of interest. Instead, it should learn the importance of different dimensions automatically.

Louis (2013) proposed to evaluate documents from the aspects of content, organisation, and reader interest, where the documents can be from academic writing, science journalism, or automatically generated summaries. Content is measured by specificity and verbosity, where specificity measures the amount of details in the documents and verbosity measures which details are important to include. Document organisation is measured by regularities of the intentional structure in the documents, which are captured by syntax regularities between adjacent sentences. Reader interest is used to identify interesting articles, as captured by hand-engineered features such as the use of visual language,

and the use of sentiment and emotions. The focus of Louis (2013) is on examining the impact of content, organisation, and reader interest across documents from different genres.

2.2 Task Modelling

To automatically assess the quality of documents, supervised machine learning techniques have generally been adopted, modelling the task as one of classification, regression, ordinal regression, or ranking. Below, we present a general cross-task description of classification, regression, ordinal regression, and ranking approaches from Section 2.2.1 to Section 2.2.4, respectively, followed by task-specific modelling details in Section 2.2.5.

2.2.1 Classification

One way of modelling document quality assessment is via classification, whereby document labels are viewed as unstructured discrete categories. While conceptually simple, document quality classifiers are unable to explicitly model the ordinal relationship between gold labels (e.g., the fact that an FA document in Wikipedia is more similar to a GA document than a Stub document).

2.2.2 Regression

When document quality assessment is treated as a regression problem, gold labels of documents are first transformed into integers or real values, thus making comparison of labels numerically possible and preserving the ordinal relationship of labels. During inference, the predicted real value is often first rounded to an integer value, which is then converted back to the predefined label. The difficulty in modelling the task as a regression problem is to quantify the gaps among labels so as to determine the "correct" integer values corresponding to gold labels. This is because the quality difference between adjacent quality labels is not identical. For example, the quality difference between an

FA Wikipedia article and a GA article is different from that between a Start article and a Stub article.

2.2.3 Ordinal Regression

Ordinal regression can be considered as an intermediate approach between classification and regression, where the relative ordering of labels is captured, but no assumptions are made about the relative similarity of different labels. Take quality assessment of Wikipedia articles as an example, the six quality labels are discrete, and there is also an ordinal relationship between them: FA > GA > B > C > Start > Stub.

2.2.4 Ranking

When a task is modelled as a ranking problem, the gold labels are the rank of documents, either via a pair-wise comparison or comparison within a set of documents. The ranking model is able to predict the rank for a set of documents but not their actual classes.

2.2.5 Domain-specific Modelling

Quality Assessment of Wikipedia Articles: Identifying the quality label of Wikipedia articles has been modelled as classification, regression, and ranking problems in previous work. The predominant approach has been to use classification approaches (Stvilia et al., 2005, Lipka and Stein, 2010, Warncke-Wang et al., 2013, Dang and Ignat, 2016c,a, 2017), treating the quality labels of Wikipedia articles as discrete categories. In studies modelling the task as a regression problem (Dalip et al., 2009, 2011, 2017, Zhang et al., 2018), the quality labels are converted to integers ranging from 1 to 6, where 1 corresponds to Stub and 6 corresponds to FA. Elsewhere, Hu et al. (2007), de La Robertie et al. (2015), and Li et al. (2015) have modelled the task as a ranking problem, where quality labels are ranked from FA (high) to Stub (low).

Academic Paper Rating: Predicting whether a paper is accepted (Kang et al., 2018, Ghosal et al., 2018a, Leng et al., 2019) and whether a submission is within the scope of a

target venue (Ghosal et al., 2018b, 2019a) have been modelled as classification problems. On the other hand, it is also possible to treat it as a regression problem (Kang et al., 2018, Qiao et al., 2018, Ghosal et al., 2019d,c) if the prediction is at a fine-grained level, such as recommendation scores.

Essay Scoring: Essay scoring has been modelled as classification, regression, ordinal regression, and preference ranking problems in the literature. Classification approaches have been most popular (Rudner and Liang, 2002, Burstein et al., 2003, McNamara et al., 2015, Wang et al., 2018b), including the Intelligent Essay Assessor (Schreiner et al., 1997, Foltz et al., 1999, Landauer et al., 2003), which adopts a 1-nearest neighbour approach, assigning the same score as the graded essay which has the highest similarity with the essay of interest. When treated as a regression problem (Page, 1994, Attali and Burstein, 2004, Persing et al., 2010, Phandi et al., 2015, Wachsmuth et al., 2016, Cozma et al., 2018), the essay scores are used as is. Elsewhere, Woods et al. (2017) formulate the task as an ordinal regression problem. There has also been work that has modelled the task as a preference ranking problem (Briscoe et al., 2010, Yannakoudakis et al., 2011, Yannakoudakis, 2013).

Content Quality Assessment in cQA: Content quality assessment in cQA has been modelled as classification, regression, and ranking problems. Once again, classification approaches have been common (Agichtein et al., 2008, Shah and Pomerantz, 2010, Burel et al., 2012, Le et al., 2016, Mihaylov and Nakov, 2016). There has also been work based on a regression formulation (Tran et al., 2015) or ranking (Huang et al., 2007, Dalip et al., 2013, Guzmán et al., 2016a,b, Molino et al., 2016, Dalip et al., 2017).

2.3 Evaluation Metrics

To assess model performance, a variety of evaluation metrics have been adopted to compare predicted quality labels against human-assigned scores. Below, we summarise discrete evaluation metrics (Section 2.3.1), rank-based evaluation metrics (Section 2.3.2), and continuous evaluation metrics (Section 2.3.3), followed by discussion of the specific evaluation metrics that have been commonly adopted in each domain, in Section 2.3.4.

2.3.1 Discrete Evaluation Metrics

Discrete evaluation metrics compare the predicted discrete labels against the actual discrete labels, and include accuracy, precision, recall, F_1, Kappa (Cohen, 1960), and quadratic weighted Kappa (QWK) (Cohen, 1968), which are described below.

Accuracy: Accuracy is defined as the proportion of correct predictions over the number of predictions made, where predictions for all classes are considered. However, accuracy penalises all wrong predictions equally, not taking the degree of wrong predictions into consideration. It is also sometimes referred to as exact agreement.

Precision: Precision for a class is defined as the number of items correctly labelled as belonging to the class over the overall number of items labelled as belonging to the class.

Recall: Recall for a class is defined as the number of items correctly labelled as belong to the class divided by the overall number of items that actually belonging to the class.

F_1: F_1 is the harmonic mean between precision and recall, which is computed as $\frac{2\times(precision\times recall)}{precision+recall}$.

Kappa: Kappa is a measure of agreement between human-assigned scores and predictions, or between two human-assigned scores. It provides a statistical method to correct for chance agreement.[5] Kappa value is computed as

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.1}$$

where $P(A)$ is the proportion of observed agreement and $P(E)$ is the proportion of expected agreement by chance. $P(E)$ is computed as

$$P(E) = \frac{1}{N^2}\sum_k n_{k1}\, n_{k2} \tag{2.2}$$

[5]Among the four domains discussed in this thesis, Kappa is usually used to evaluate model performance in essay scoring. A brief description of Kappa is beneficial for readers to understand different evaluation metrics adopted in different domains, noting that we do not perform experiments on essay scoring in this thesis, so never use this metric.

Here, N is the number of instances, k is the number of categories, and n_{ki} is the number of times rater i predicted category k. The disadvantage of this metric is that it treats different degrees of disagreement equally.

Quadratic Weighted Kappa: QWK not only corrects for chance agreement but also takes the extent of disagreement into account. QWK is computed as

$$\kappa_{qwk} = 1 - \frac{\sum_{i,j} \mathbf{w}_{i,j} \mathbf{O}_{i,j}}{\sum_{i,j} \mathbf{w}_{i,j} \mathbf{E}_{i,j}} \tag{2.3}$$

Here, $\mathbf{O}$, $\mathbf{E}$, and $\mathbf{w}$ are matrices of observed scores, predicted scores, and weights, respectively. $\mathbf{O}_{i,j}$ represents the number of documents that receive a label i from the first rater, and a label j from the second rater. $\mathbf{E}$ represents the outer product of the two label vectors returned by two raters, which is normalised to have the same sum as $\mathbf{O}$. $\mathbf{w}_{i,j}$ is computed as $(i-j)^2/(N-1)^2$, where N is the number of possible labels.

2.3.2 Rank-based Evaluation Metrics

Rank-based evaluation metrics compare two sets of rankings, and include normalised discounted cumulative gain (NDCG) (Järvelin and Kekäläinen, 2002), Spearman's rank correlation (Spearman, 1961), mean reciprocal rank (MRR), and mean average precision (MAP), which are described below.

Normalised Discounted Cumulative Gain: NDCG of n articles is computed as

$$NDCG = \frac{1}{N}(r_1 + \sum_{i=2}^{n} \frac{r_i}{\log_2(i)}) \tag{2.4}$$

where r_i is the corresponding transformed value of the ith ranked article and N is the discounted cumulative gain score for an ideal prediction where all documents are assigned quality labels correctly. To compute NDCG, each quality label is first transformed into an integer score. NDCG assigns a higher weight to articles which are predicted as higher quality classes than as lower quality classes. In other words, NDCG penalises wrong predictions according to their degree, and the penalty is asymmetric: a more severe

penalty is imposed for predicting high-quality documents to be of low quality than the other way around. NDCG scores range from 0 to 1, where a higher value indicates a better quality ranking.

Spearman's Rank Correlation: Spearman's rank correlation is a non-parametric robust measure of rank correlation, which assumes the ranking of two variables are statistically dependent. Spearman's rank correlation is only sensitive to the ordinal arrangement of values, thus making it robust to outliers. Spearman's rank correlation is computed as

$$\rho_s = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \tag{2.5}$$

where d_i is the rank difference between two observations, and n is the number of observations.

Mean Reciprocal Rank: MRR of an individual query is the reciprocal of the rank at which the first relevant answer is returned, or 0 if none of the top N results is relevant. For a set of queries, MRR is the mean of reciprocal ranks for each individual query, which is computed as

$$\text{MRR} = \frac{1}{|Q_r|} \sum_{q \in Q_r} \frac{1}{r_q} \tag{2.6}$$

where Q_r is a set of queries, $|Q_r|$ is the number of queries, and r_q is the rank of the first relevant document for the query q. Here, in the context of quality assessment in cQA, a question is a query, and the corresponding set of answers is considered to be the retrieved documents. High-quality documents are considered to be relevant documents. If all the documents are high-quality, then MRR is 1. If none of the documents are high-quality, then MRR is 0. MRR only takes the first relevant answer into account. If there is only one relevant answer, MRR can fully capture model performance. However, it is unable to fully measure model performance if there are multiple relevant documents.

Mean Average Precision: Before explaining MAP, we first describe the definition of precision@k, and average precision for an individual query. For a given query, precision@k measures the fraction of documents ranked in the top k results that are relevant. The average precision for a query is defined as the mean of precisions calculated after each

relevant document is retrieved. MAP is defined as the mean of average precisions for a set of queries, which is calculated as

$$\text{MAP} = \frac{1}{|Q_r|} \sum_{q \in Q_r} \frac{\sum_{r=1}^{N} (P(r) \times rel(r))}{|R_q|} \tag{2.7}$$

where Q_r is a set of queries; R_q is a set of relevant documents for query q; $|Q_r|$ is the number of queries; $|R_q|$ is the number of relevant documents for query q; N is the number of returned documents; r is the rank position; $P(r)$ is the precision value at a given cut-off rank r; and the function *rel* indicates whether the document is relevant. Unlike MRR that only takes the first relevant document into consideration, MAP takes all relevant documents into account.

2.3.3 Continuous Evaluation Metrics

Continuous evaluation metrics measure the predicted integer/real values against actual integer/real values, and include mean square error, adjacent agreement, and Pearson's correlation coefficient (Benesty et al., 2009). Below, we briefly describe each of these metrics.

Mean Square Error: Mean square error is defined as the average difference between the predicted values and actual values, and penalises different degrees of wrong predictions differently. The disadvantage of mean square error is that the penalty of wrong predictions with the same degree is symmetric. Take quality assessment of Wikipedia articles as an example, the penalty of predicting an FA article to be Stub is the same as that of predicting a Stub article to be FA. In the real world, humans are more interested in high-quality documents, and correct prediction of their quality labels is more important, which mean square error is unable to capture.

Adjacent Agreement: Adjacent agreement is a simple way of measuring the agreement between human-assigned scores and predicted scores. It is computed as the percentage of predictions within a predefined score difference window of one (e.g., the difference

between a human-assigned score and predicted score is one), and is a relaxed version of exact agreement.

Pearson's Correlation Coefficient: Pearson's correlation coefficient is a parametric metric, which measures the linear correlation between two sets of values. It ranges from -1 to 1, where 1 denotes perfectly positive linear correlation; 0 denotes no linear correlation; and -1 denotes perfectly negative linear correlation. Pearson's correlation is computed as

$$\gamma = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \tag{2.8}$$

where x_i and y_i are sample data pairs from datasets x and y, respectively; and $\bar{x}$ and $\bar{y}$ are the mean value of sample datasets x and y, respectively.

2.3.4 Evaluation Metrics Adopted For Each Task

Quality Assessment of Wikipedia Articles: When the task is modelled as a classification problem, accuracy is the most widely adopted metric. However, accuracy penalises all wrong predictions equally, and does not reflect the different costs of wrong predictions. For example, a model should be penalised more if its prediction is far away from the true label (e.g., predicting an FA as Stub compared with predicting an FA as GA). When the task is modelled as a regression problem, mean square error is generally used to measure model performance. When the task is modelled as a ranking problem, NDCG and Spearman's rank correlation are used, where each quality label is first transformed into an integer value: e.g., 5 for FA, 4 for GA, 3 for B, 2 for C, 1 for Start, and 0 for Stub..

Academic Paper Rating: When academic paper rating is modelled as a classification problem, evaluation metrics such as accuracy, precision, recall, and F_1, are generally adopted. Here, F_1 is more suitable to measure the performance of models over imbalanced datasets, such as a large number of rejected papers vs. a relatively small number of accepted papers. Micro-averaging is used to compute the precision, recall, and F_1 among classes if the dataset is balanced; otherwise, macro-averaging is adopted. When academic

paper rating is modelled as a regression problem, NDCG, and root mean square error are commonly adopted.

Essay Scoring: When the task of essays scoring is modelled as classification, regression, and ranking problems, evaluation is generally via correlation or agreement. Common evaluation metrics used in essay scoring include: exact agreement, adjacent agreement, mean square error, Kappa, QWK, Pearson correlation, and Spearman's rank correlation, among which QWK is the most commonly used evaluation metric. Exact agreement, adjacent agreement, and mean square error do not take chance agreement into consideration. Although Kappa considers chance agreement, it does not take different degrees of disagreement into account, where QWK comes to the rescue. QWK not only takes chance agreement into account but also assigns different weights to different degrees of disagreement. Pearson correlation is also used in the literature.

Content Quality Assessment in cQA: When content quality assessment in cQA is modelled as classification and regression problems, accuracy, precision, recall, and F_1 score with regards to the high-quality class are widely used. When the task is to differentiate high-quality answers from bad ones, it is often modelled as a ranking problem, where MRR, MAP, and NDCG are used.

2.4 Hand-crafted Features

Various types of hand-crafted features have been engineered for different sources, capturing different aspects of document quality. For quality assessment of Wikipedia articles, features can be extracted from the article content itself, the revision history, and metadata of Wikipedia contributors, as described in Section 2.4.1. For academic paper rating, features are mostly extracted from the paper, sometimes from the metadata of authors, as described in Section 2.4.2. For essay scoring, features are extracted from the essay and the corresponding prompt, as summarised in Section 2.4.3. For content quality assessment in cQA, features can be extracted from the question or answer itself, users' metadata, and

usage of the content. Lastly, an overall discussion of hand-crafted features is presented in Section 2.4.5.

2.4.1 Quality Assessment of Wikipedia articles

Wikipedia article features can be extracted from different sources, such as the article content itself, the revision history, and metadata of Wikipedia contributors. Features extracted from the article itself measure whether the article is properly organised, is supported by evidence, uses appropriate words, and is easy to understand. Features extracted from the revision history of a Wikipedia article measure how stable an article is, including the article age, the number of revisions made to the article, the number of revisions per day, and the number of revisions made by registered editors. With the assumption that the reputation of editors and the quality of articles edited by them are correlated, studies have explored the mutual dependency between article quality and editor reputation.

Below, we detail these different types of features.

2.4.1.1 Article Internal Features

Article length is the most commonly explored metric (Dondio and Barrett, 2007, Blumenstock, 2008, Dalip et al., 2009, 2011, Warncke-Wang et al., 2013, 2015, Dang and Ignat, 2016a, Dalip et al., 2017), measured at the byte, character, word, or phrase level. A high-quality article should be neither too short (an overly short article may indicate incomplete topic coverage) nor too long (an overly long article may indicate overly verbose content). Article length is also used to normalise other features correlated with it (Dalip et al., 2009, 2011, Warncke-Wang et al., 2013, Dalip et al., 2014, Warncke-Wang et al., 2015, Dang and Ignat, 2016a), such as the number of references, links, images, and headings.

The structure of an article reflects how well the article is organised. A well organised article is generally easier to read and more likely to be of high quality than one which combines everything into one paragraph. Warncke-Wang et al. (2013) and Dang and Ignat (2016a) explore the number of level 2 and level 3+ headings. Dalip et al. (2009, 2011,

2014, 2017) explore structure features in a detailed way: such as the summary section length, average paragraph length, and statistics of sections and subsections (the number of sections, average section length, average subsection per section, etc.).

Other structural features have included images, Wikipedia templates, the categories an article belongs to, article versions in other languages, and the infobox. Images in an article can make the content clearer and visually pleasant to read. The (article-length normalised) number of images has been widely explored (Dondio and Barrett, 2007, Dalip et al., 2009, 2011, Warncke-Wang et al., 2013, Dalip et al., 2014, Warncke-Wang et al., 2015, Dang and Ignat, 2016a, Dalip et al., 2017). The use of Wikipedia templates reflects whether the content conforms to Wikipedia rules. Warncke-Wang et al. (2015) and Dang and Ignat (2016a) examined the number of citation and non-citation templates. Similarly, the more categories an article is linked to, the more likely the article is edited and reviewed by contributors. Warncke-Wang et al. (2015) and Dang and Ignat (2016a) explored the number of categories in the article. With the same intuition, the number of article versions in other languages has also been explored (Dalip et al., 2009, 2011, 2014, 2017). Elsewhere, Warncke-Wang et al. (2013), Warncke-Wang et al. (2015), and Dang and Ignat (2016a) have explored whether an article contains an infobox.

The presence or number of references/links can provide evidence for an article, thereby enhancing the trustworthiness of an article. Dondio and Barrett (2007), Dalip et al. (2009, 2011, 2014), Warncke-Wang et al. (2015), Dang and Ignat (2016a), and Dalip et al. (2017) explore such features, either in absolute terms or article-length normalised terms, or both. There have also been studies (Dalip et al., 2009, 2011, 2014) that have differentiated internal and external links, where internal links link to other Wikipedia pages and external links link to pages outside Wikipedia. A Wikipedia entry that is mentioned in other Wikipedia articles without explicitly referencing the entry may indicate that this entry is less trustworthy. McGuinness et al. (2006) study the ratio between the number of citations to the article entry of interest and the number of mentions (cited and non-cited) of the entry by other articles to measure the trustworthiness of the article. Other studies (Dalip et al., 2009, 2011, 2014, 2017) have focused on features based on the network of articles,

where articles are nodes and citations between articles are edges. Features based on the article network include *PageRank* value of an article (Page et al., 1998), *in-degree* (the number of links to this article), *out-degree* (the number of links to other articles), assortativity *in-in*, *in-out*, *out-in*, and *out-out*, etc., where the assortativity values are computed on the degree of the node and the average degree of its neighbours, either in-degree or out-degree.

Word usage and writing style have also been studied. Lipka and Stein (2010) exploit character and POS trigrams, which are assumed to reveal author preferences for word usage (such as the utilisation of stop words, adverbs, and punctuations) and author preferences for sentence constructions, respectively. With the same intuition, Dalip et al. (2009, 2011, 2014, 2017) explore POS features in more detail, such as the number of auxiliary verbs, *to be* verbs, conjunctions, prepositions, and pronouns. They also explore the number of passive and question sentences. The subjective opinions of Wikipedia contributors can potentially lower the quality of Wikipedia articles. Thus, Lewoniewski et al. (2017) study morphological features of Russian Wikipedia articles, such as personal and possessive pronouns of the first and second person to measure the subjective opinions of Wikipedia contributors. Xu and Luo (2011) explore lexical features, such as the number of words, unique words, nouns, unique nouns, verbs, unique verbs, and copulas (e.g., *seem*, and *appear*). They also study interactions between these lexical features, such as the ratio of unique words to words (i.e., types to tokens) to measure the variety of word usage. Their intuition is that high-quality articles should include more concepts, objects, facts, and longer sentences than low-quality ones of the same length. Warncke-Wang et al. (2015) and Dang and Ignat (2016a) turn their attention to information noise score (Stvilia et al., 2005), based on the ratio of the number of tokens after stemming and stopping to the number of tokens before processing, to measure the informativeness of Wikipedia articles.

To measure the comprehension difficulty of Wikipedia articles, various readability scores have been explored (Stvilia et al., 2005, 2007, Dalip et al., 2009, 2011, 2014, Dang and Ignat, 2016a, Dalip et al., 2017), including Flesch reading ease score (Kincaid et al.,

1975), Flesch-Kincaid grade level (Kincaid et al., 1975), Smog index (McLaughlin, 1969), Coleman-Liau index (Coleman and Liau, 1975), automated readability index (Kincaid et al., 1975), Gunning-Fog index (Gunning, 1969), Dale-Chall score (Dale and Chall, 1948), Linsear write formula (Chen, 2012), and difficult words (Chall and Dale, 1995). Readability scores, including Flesch reading score, Flesch-Kincaid grade level, Smog index, and Linsear write formula, use the average syllables per word or the number of polysyllables with different weight values to measure how difficult a text in English is to understand. Both Coleman-Liau index and automated readability index use the average word length with different weights to measure the readability of texts. Difficult words, Dale-Chall score, and Gunning-Fog index use the number of difficult words or percentage of difficult words to measure the comprehension difficulty of a text. Here, a word is considered to be difficult if it is not in a list of 3,000 common English words that fourth-grade American students can reliably understand. Läsbarhetsindex score is computed based on the ratio of complex words (words with more than 6 characters) and average sentence length. Similarly, to measure the complexity of Russian Wikipedia articles, Lewoniewski et al. (2017) explore the number of unique words, and the number of words with more than 2, 3, and 4 syllables.

2.4.1.2 Metadata Features

An article that receives a large number of edits is more likely to improve over time. Dalip et al. (2009), Wöhner and Peters (2009), and Dalip et al. (2011, 2014, 2017) examine the number of revisions an article experiences, and the article age (in days). They also explore the interaction between revisions and article age, such as revisions per day, average time per revision, percentage of revisions made in the last three months, and the number of lines modified, comparing the current version of an article to a version from three months prior. Here, revisions per day denotes how frequently an article is reviewed and updated, while the other measures indicate how stable an article is. The number of posts in the talk page has been explored by previous studies (Dondio and Barrett, 2007, Dalip et al., 2009, 2011, 2014, 2017), as it is a proxy for the improvement process of an article, and can indicate the quality status of the article to some extent.

There have been studies exploring editor-oriented features. For example, Dondio and Barrett (2007), and Warncke-Wang et al. (2013) explore the percentage of registered editors or the number of unique editors over the total number of edits to an article. Dalip et al. (2009, 2011, 2014, 2017) also explore the interaction between revisions and editors, such as the number of revisions made by anonymous users, registered users, and occasional users (users who edit the article less than four times). Warncke-Wang et al. (2013) explore the tenure of the article, as the sum of tenures of editors who edit this article. The tenure of an editor is based on how long the editor has been registered and the total number of edits made by the editor in Wikipedia. There has been work exploring team characteristics to predict article quality, where the team of an article consists of all contributors who have edited the article. For example, Betancourt et al. (2016) explore team characteristics, such as statistics (e.g., min, max, and mean) of team efficiency and tenure. Here, the efficiency of a user is the ratio of the number of FA/GA articles that the user works on, to the total number of articles that they work on.

2.4.1.3 Article-Editor Mutual Dependency Features

An editor's reputation may correlate with the quality of articles edited by the editor. Studies explore such correlations by looking through the article edit history. For example, Stein and Hess (2007) explore the reputation of editors to measure the quality of articles. The reputation of an editor is computed in two different ways: page-based author reputation, and edit-based author reputation. Page-based author reputation is computed as the number of FA articles an editor has edited over the total number of articles they have edited. Edit-based author reputation is computed as the number of edits to FA articles contributed by an editor, over the total number of edits on all articles made by the editor. The quality of an article is determined by the average page-based or edit-based reputation of the editors who edited the article. Li et al. (2015) explore the editor-article network, where articles and editors are nodes and there is an undirected edge between the editor and article nodes if the editor edits the article. The edges are weighted by the number of edits or the number of words contributed by the editor to the article. Node values of an

article, computed by PageRank, are used to rank articles according to their quality. Zeng et al. (2006) utilise the revision history of an article to compute the quality of a particular article version. They use a dynamic Bayesian network to model the trustworthiness of an article version, which depends on the reputation of its editor, number of words the editor has changed, and the quality score of the previous version of the same article.

There have been studies (Javanmardi and Lopes, 2010, de La Robertie et al., 2015) exploring peer review based approaches, where text quality is determined by not only the reputation of the editors who author it but also the reputation of editors who review it. For example, Suzuki and Yoshikawa (2012, 2013) adopt a survival ratio based approach. Text contributed by an editor not deleted by subsequent contributors is considered to be survived text, and the number of revisions the text has survived is called its survival ratio. An editor's reputation is determined by the survival ratio of text contributed by the editor, which in turn can be used to assess the quality of an article. The same authors also use h-index to measure an editor's reputation (Suzuki, 2015). The h-index of an editor e is $h(e)$ if the editor gets at least $h(e)$ positive ratings in $h(e)$ articles and less than $h(e)$ positive ratings in the other $N - h(e)$ articles, where N is the number of articles the editor e edits. If the proportion of text contributed by the editor and kept by subsequent editors exceeds a predefined threshold, then we say that the editor gets a positive rating from subsequent editors. The quality of an article is then determined by the text contributed by the editors and the editors' h-index values. With the observation that content surviving through subsequent edits indicates approval from subsequent editors, Lim et al. (2006), and Hu et al. (2007) propose a peer review based model, where the quality of an article depends on the authorities of contributors who authored or reviewed it, and the authority of contributors depends on the quality of articles authored or reviewed by contributors.

2.4.2 Academic Paper Rating

There are also relevant tasks where hand-crafted features are widely explored, such as predicting the impact of scientific concepts (McKeown et al., 2016), predicting the number of citations (Yan et al., 2011), and predicting a community's responses to a paper (Yogatama

et al., 2011). These tasks have explored different aspects of academic papers. As our focus in this thesis is to predict the acceptance of academic papers, we leave these tasks to future work. Below, we describe features used in predicting the acceptance of a paper, followed by features in measuring whether a submission is within the scope of target venues.

Hand-crafted features are extracted from multiple paper sections to predict the acceptance of a paper. For example, Kang et al. (2018) extract features from the paper title, abstract, and references, including the length of the title, whether specific words (e.g., *outperform*, *state-of-the-art*, and *novel*) appear in the abstract, and an embedded representation of the abstract. These features are used as input to different downstream classifiers, such as logistic regression, decision tree, and random forest, to predict the acceptance of a paper. Louis and Nenkova (2012) use syntactic patterns to differentiate conference papers from workshop papers, where syntactic features are extracted from abstracts, introductions, and related work. Elsewhere, Vincent-Lamarre and Larivière (2019) examine the differences between accepted and rejected papers from lexical, semantic, and psycholinguistic views. Specifically, they compare the title, abstract, and introduction of accepted and rejected papers from lexical, semantic, and psycholinguistic views. Findings show that readability measured via Flesch reading ease score and New Dale-Chall readability correlate negatively with the acceptance of papers and scientific jargons identified via scientific-specific and scientific-general jargon lists correlate positively with the acceptance of papers. Analysing words used in papers, they find that words used in accepted papers are less frequent and more abstract than those in rejected papers. They also find that accepted papers tend to have more references in common by bibliographic coupling (finding the number of shared references), and accepted papers are more similar than rejected papers measured by the semantic (cosine) similarity computed over the *tf-idf* weighted title, abstract, and introduction of papers.

To predict whether a submission is within the scope of target venues, Ghosal et al. (2018a,b) hand-engineer features capturing the scope and quality of a paper, where scope features measure whether a submission fits into the scope of a given venue, and quality features measure whether a paper is of high quality. Scope features are obtained by

comparing the bibliographic information in the given submission with that in historical papers accepted by the target venue, such as the number of bibliographic titles and venues shared by both the paper of interest, and historical papers accepted by the venue. Quality features are obtained by examining authors, affiliation, references, non-text content in the given paper, such as primary author's *h*-index, the number of papers from the authors' institutions, and the number of figures and tables. Similarly, Ghosal et al. (2019a) have explored three types of features: lexical features represented by applying *tf-idf* to the paper content, semantic features represented by concatenating word embeddings of the top 30 most frequent words and keywords from the paper, and bibliography features represented by applying *tf-idf* to titles and venues in the reference section.

2.4.3 Essay Scoring

Various types of features have been proposed to capture different aspects of an essay, such as persuasiveness and topic relevance, organisation and coherence, sentence complexity and word usage, and grammar and mechanics features. Persuasiveness and topic relevance indicate how well an author argues in favour of their stance on a topic, which is reflected through ideas and content. A well-developed persuasive essay clearly takes a position in a consistent way without going off topic, and provides corresponding evidence to support its stance. Organisation and coherence indicate how well ideas in an essay are organised. A well-organised and coherent essay presents ideas in a logical way with smooth transitions between ideas, paragraphs, sentences, and phrases. Sentence complexity and word usage indicate how well an author has mastered the language. A well-written essay uses a variety of sentence structures and precise words to clearly present its content, such as complex sentences, varied word choice, and specialised vocabulary. Grammar and mechanics indicate how well an author follows the rules and conventions of a language. A well-written essay has few if any issues in terms of grammar (e.g., noun-verb agreement) and mechanics (e.g., capitalisation, punctuation, and spelling conventions).

Next, we detail these types of features.

2.4.3.1 Persuasiveness and Topic Relevance Features

A persuasive essay should state its opinion in a consistent way without going off topic. Farra et al. (2015) exploit opinion, topic, and opinion-target related features to score essays from the point of view of persuasiveness. Opinion features, such as the number of opinion words in an essay, are used to represent opinions in an essay, where opinion words include sentiment expressions (e.g., *learned the most*) and arguing expressions (e.g., *for these reasons, I claim with confidence*). Topic features, such as the number of topic words in the essay, are used to represent topics in the essay, where topic words are from a corpus of high-scoring essays in response to the same prompt. Opinion-target pair features, such as the number of times topic words appear as a target in essay opinion-target pairs, and a binary feature indicating the presence of opinion flips towards any target, are used to represent opinion relevancy and opinion consistency. Here, targets refers to all nouns, verbs, and adjectives in the same sentence or phrase with the opinion word. McNamara et al. (2015) and Crossley et al. (2015) exploit lexical features to capture rhetorical strategies. Lexical features, such as the number of words from different lexical categories, are used to measure rhetorical style. Lexical categories include: (1) amplifiers (e.g., *completely* and *extremely*); (2) hedges (e.g., *almost* and *maybe*); (3) assent or agreement terms (e.g., *agree*, *absolutely*, and *yes*); (4) exemplification (e.g., *for instance* and *for example*); (5) prediction modals (e.g., *will* and *shall*); (6) downtoners (e.g., *barely* and *somewhat*); (7) negations (e.g., *not, neither*, and *nor*); and (8) the use of the terms *seem* and *appear*.

Lexical overlap has also been exploited to measure the prompt relevance of an essay. Louis and Higgins (2010) measure topic relevance of an essay by expanding the prompt using inflected forms, synonyms, distributional similar words, and word association norms of prompt words. Then the *tf-idf* weighted cosine score between the essay and the expanded prompt is computed to measure similarity. There has been research that has made use of topic features (Persing and Ng, 2013, 2014, Phandi et al., 2015, Farra et al., 2015, McNamara et al., 2015, Crossley et al., 2015), such as the number of words appearing in both the essay and prompt, and the number of synonyms of prompt words in an essay,

to measure topic relevance. Conversely, Higgins et al. (2006) use the proportion of words that only appear in the essay to score essays.

Semantic overlap between an essay and its corresponding prompt or between essays has also been explored to measure prompt relevance. Higgins et al. (2004) use semantic similarity features derived from random indexing (Sahlgren, 2005), such as the random indexing similarity between a discourse element and the prompt, to measure to what extent a discourse element and the prompt contain words from the same semantic domain. If they share many semantic similar words, it indicates that the discourse element is related to the topic; otherwise, there is a chance that this discourse element is off-topic. Similarly, Persing and Ng (2013) and Persing and Ng (2014) use the random indexing similarity between an essay and groups of prompt keywords to measure the completeness and relevance of an essay. McNamara et al. (2015), Crossley et al. (2015), and Ran et al. (2018) use latent semantic analysis (LSA) (Landauer et al., 1998) between an essay and its corresponding prompt to measure whether the essay is relevant to the prompt. Elsewhere, Persing and Ng (2014) use manually annotated latent Dirichlet allocation (LDA) (Blei et al., 2003) topics to capture topic relevance. The Intelligent Essay Assessor (Schreiner et al., 1997, Foltz et al., 1999, Landauer et al., 2003) uses a LSA to measure the semantic similarity between the ungraded essay and scored essays in response to the same prompt. Burstein and Higgins (2005) and Higgins et al. (2006) exploit semantic similarity between an essay and a variety of essay prompts to detect off-topic essays. If the similarity score between the essay and its corresponding target prompt is ranked among the top few highest similarity scores, the essay is considered to be on topic; otherwise, it is considered to be off-topic.

2.4.3.2 Organisation and Coherence Features

There has been work measuring coherence at the sentence or discourse level. McNamara et al. (2015) and Crossley et al. (2015) utilise lexical and semantic co-referentiality features, such as the degree of word overlap (noun, argument, stem, and content) and semantic similarity between sentences, to measure the coherence between sentences. McNamara et al. (2015) and Crossley et al. (2015) have also explored spatial and temporal cohesion as

they can contribute to the construction of a well-structured situational model (Kintsch and Van Dijk, 1978). Spatial cohesion can be measured through location (e.g., *here*) and motion (e.g., *into*) words via WordNet (Miller, 1995), and temporal cohesion can be measured through temporal adverbials (e.g., *afterwards*), aspect repetition (e.g., *progressive*), tense repetition (e.g., *past*), and the combination of aspect and tense repetition. Higgins et al. (2004) exploit semantic similarity features derived from random indexing to measure coherence between different discourse elements. Somasundaran et al. (2014) explore interactions between lexical chains and explicit discourse cues, such as the number of chains that start after, end before, and continue over a discourse cue, to capture discourse transitions. With the assumption that *Rough-Shift transitions* from Centering Theory often occur if topics in an essay are short-lived and unconnected to prior discourse units, Miltsakaki and Kukich (2004) exploit the number and percentage of *Rough-Shift transitions* to measure the local coherence of essays.

Studies have also measured coherence at the paragraph and essay levels. For example, McNamara et al. (2015) and Crossley et al. (2015) utilise paragraph cohesion features, such as lexical and semantic overlap between paragraphs, to capture whether different paragraphs in the same essay focus on the same topic. They also explore lexical diversity features, such as the ratio of unique words in an essay, to capture coherence, under the assumption that a high ratio of unique words indicates that the essay is more likely to be incoherent. Somasundaran et al. (2014) exploit properties of lexical chains (Morris and Hirst, 1991a) to measure coherence. For example, the presence of large chains indicates that the topic is discussed in detail, while the presence of many small chains or no chains indicates the text jumps from one topic to another, or has no central ideas. Lexical chains consist of sequences of related words, such as repeat words, and synonyms.

There has been work focusing on identifying discourse elements, such as thesis, conclusion, and premise, which affect the quality of essays. For example, Persing et al. (2010) use specific words and paragraph position to identify discourse elements. Specifically, if the words *agree*, *think*, or *opinion* appear in a sentence, there is a high probability that the sentence is a thesis, which in turn is used to decide the function of a paragraph. Position

can also be used to identify discourse elements: the last paragraph in an essay is more likely to be a conclusion. Wachsmuth et al. (2016) utilise prompt similarity, n-gram, POS n-gram and sentence position features to assign appropriate argumentative discourse unit labels (e.g., thesis, conclusion, premise, and none) to sentences, which are later used to score essays. Here, the similarity between a sentence and the prompt can be measured based on Euclidean, Manhattan, or Jaccard distance. Sentence position is the relative position of a sentence in a paragraph, such as the first or last sentence in a paragraph. Ong et al. (2014) exploit the appearance of different types of connective senses (e.g., expansion, contingency, comparison, and temporal) to identify different diagram ontology elements (e.g., hypothesis and claim). Ghosh et al. (2016) use lexical features (e.g., word uni-grams, bi-grams, or tri-grams, and verbs), syntactic features (e.g., production rules from parse trees and the number of clauses in the argument), to identify argument components (e.g., major claim, claim, premise, and none).

There has also been work on identifying discourse structures (Wachsmuth et al., 2016). If there are discourse elements missing or the discourse elements are not organised in a logical way, the score of an essay is low. Burstein et al. (2003) explore status and rhetorical relations features, such as nucleus, and satellite of a sentence's parent node and a sentence's rhetorical relations (e.g., elaboration, background, and cause), to capture discourse structure in essays. Here, the nucleus contains basic information, which is often comprehensible without satellites, and the satellite provides additional information about a nucleus, which is usually incomprehensible without the nucleus. Under the assumption that cue words, terms, and syntactic structures can function as discourse markers, Burstein et al. (2003) exploit cue words (e.g., *first* and *because*) and syntactic structure (e.g., subordinate and complement clauses) to identify discourse structure. Position features, such as the relative position of a paragraph, and sentence punctuation, such as full stops, question marks, and exclamation marks, can be exploited to identify discourse structure.

Relations between discourse units have been the target of past research (Longo, 1994, McNamara et al., 2015, Crossley et al., 2015, Ghosh et al., 2016). McNamara et al. (2015)

and Crossley et al. (2015) explore connective features to capture relationships between clauses, sentences, and ideas in an essay, including the number of connectives per 1000 words, the difference between positive connectives (e.g., *also* and *moreover*) and negative connectives (e.g., *however* and *but*), and the number of connectives revealing different relations between discourse units (e.g., causal, contrastive, and temporal). Ghosh et al. (2016) and Amorim et al. (2018) use discourse marker indicators, such as comparison, contingency, and expansion, to identify argument relations (support and non-support).

2.4.3.3 Sentence Complexity and Word Usage Features

There has been work focusing on measuring sentence structure and sentence complexity. To measure sentence structure, Attali and Burstein (2004) use the number of passive and long sentences, while PEG (Page, 1968, 1994) uses the number of prepositions and relative pronouns. IntelliMetric (Rudner et al., 2006) uses syntactic variety and readability to measure sentence variety and complexity. Yannakoudakis et al. (2011) and Yannakoudakis (2013) make use of distance features between a head and dependent in a grammatical relation, such as the mean value of grammatical relation distance per sentence, to represent syntactic complexity of sentences. To measure syntactic complexity, McNamara et al. (2015) and Crossley et al. (2015) exploit syntactic and syntactic similarity features. Syntactic features include the average number of words before the main verb and the average number of higher-level constituents (sentences and embedded sentence constituents) per word. Syntactic similarity is computed by calculating the uniformity and consistency of the syntactic constructions at the clause, phrase, and word levels.

Word variety, lexical knowledge, and lexical complexity have also been studied. Somasundaran et al. (2014) use features based on lexical chains (Morris and Hirst, 1991b) to capture word usage variety, such as the number and percentage of chains, where a lexical chain is a sequence of related words spanning a short or long text. To measure word usage and lexical complexity, Attali and Burstein (2004) explore features such as the number of verb formation errors, average word frequency, and average word length in characters, where verb formation errors measure the extent of words being correctly

used and average word frequency and average word length in characters measure lexical complexity of an essay. McNamara et al. (2015) and Crossley et al. (2015) further explore word information and semantic features. Word information features are used to measure lexical sophistication, including measures of concreteness, imageability, meaningfulness, hypernymy, and polysemy. Semantics features include verbal semantic and psychological semantic features. Verbal semantics is measured through private verbs (e.g., *doubt*), public verbs (e.g., *say*), and *be* verbs. Psychological semantic features are derived from LIWC (Pennebaker et al., 2007), such as the number of words from the category of cognitive processes (e.g., *think*), perceptual processes (e.g., *hear*), and social processes (e.g., *talk*). Elsewhere, Alikaniotis et al. (2016) propose a word embedding model, which is able to capture both local contextual and word usage information, such as the extent of specific words contributing to essay scores.

Word association and word similarity between essays have also been explored. Klebanov and Flor (2013) use point-wise mutual information to quantify the degree of co-occurrence between two words. A higher proportion of word pairs with high point-wise mutual information values indicates a better-developed topic, and a higher proportion of word pairs with negative point-wise mutual information values indicates creative use of language. With the assumption that high-score or low-score essays resemble other high-score or low-score ones in their word choice, Attali and Burstein (2004) measure the vocabulary similarity between an essay of interest and essays from different score categories to capture how an essay resembles other essays in terms of word choice.

2.4.3.4 Grammar and Mechanics Features

Both n-gram and POS n-gram features have been explored to identify grammatical errors. Briscoe et al. (2010), Yannakoudakis et al. (2011), and Yannakoudakis (2013) exploit uni-gram, bi-gram, and tri-gram lexical features, such as the number of tri-grams that do not appear in a n-gram model, and phrase structure features, such as the appearance of rare constructions, to identify grammatical errors in an essay. To better estimate the error rate, they extend the tri-gram language model with high frequency tri-grams from First

Certificate in English. Higgins et al. (2004) and Attali and Burstein (2004) also explore grammatical features, such as the number of agreement errors and preposition word usage. Similarly, Lonsdale and Strong-Krause (2003) exploit a modified link grammar parser (Sleator and Temperley, 1995) to capture grammatical errors in essays. Persing and Ng (2013) exploit POS n-gram features, such as the number of discriminative POS tri-grams in an essay. A POS tri-gram is considered to be discriminative if it has a high rank based on information gain in identifying high-quality essays from low-quality essays. Similarly, Phandi et al. (2015) explore the number of bad POS n-grams to capture grammatical errors, where a POS n-gram is considered to be a bad POS n-gram if it is not in a POS n-gram list retrieved from a grammatically correct text corpus.

Previous studies have also measured mechanics errors (Higgins et al., 2004, Attali and Burstein, 2004). For example, Persing and Ng (2013) use the number of spelling errors in the most misspelled sentence of an essay to measure the effect of misspellings on essay clarity. A word is considered to be a spelling error if this word does not appear in a predefined dictionary. McNamara et al. (2015), Crossley et al. (2015), and Phandi et al. (2015) explore frequency of punctuation, such as commas, periods, semicolons, and ending punctuation symbols (e.g., ".", "?", and "!"). Rudner et al. (2006) use mechanics features to examine whether an essay follows the conventions of standard American English, such as capitalisation and punctuation.

2.4.4 Content Quality Assessment in cQA

To assess the quality of content in cQA, features capturing different perspectives are engineered from different sources, such as the answer itself, the relationship between questions and answers, metadata of questions and answers, asker and answerers characteristics and their behavioural history in the community, relations of the answer of interest with other answers from the same thread, and usage information. Features extracted from the content itself measure whether the content: (1) conforms to common writing practices and policies of community platforms; (2) provides enough information to the asker; and (3) is easy to understand. To measure the relevance between the question

and answer, similarity between them is measured. Metadata of questions and answers is also beneficial in measuring the quality of an answer, such as the posting time difference between the question and answer. Users' characteristics and behavioural history in the community also reflect the content quality provided by the user to some extent, as a user with high reputation in a community and who has provided historically good answers is more likely to provide good answers in the future. The perceived quality of an answer is also dependent on all other competing answers within the same thread. If the quality of an answer is not high, but the quality of other competing answers within the same thread is lower than the answer of interest, the answer is more likely to be selected as the best answer. Usage features, such as the number of times the answer is printed, indicate the utility of an answer, which in turn reflects the quality since low-quality answers are less likely to be useful.

In what follows, we first describe features engineered from the question and answer and their metadata in Section 2.4.4.1, followed by features extracted from user metadata in Section 2.4.4.2. Next, we summarise features capturing content quality at the thread level in Section 2.4.4.3, before outlining usage features in Section 2.4.4.4.

2.4.4.1 Question and Answer Features

Question and answer features include five types of features: question and answer textual features, question and answer relationship features, question-related metadata features, answer-related metadata features, and question and answer metadata relationship features. The first two types of features are derived from the question and answer themselves, and the relationship between the question and answer, while the other features are engineered from question metadata, answer metadata, and question and answer metadata relationship, respectively. Below, we detail these features.

Question and answer textual features include text length, visual quality of content (mechanics and grammaticality), presence of special tokens, references, and links, word usage and lexical complexity, and readability.

Text length is the most commonly used feature in assessing the quality of content (Jeon et al., 2006, Adamic et al., 2008, Suryanto et al., 2009, Shah and Pomerantz, 2010, Burel et al., 2012, Le et al., 2016). Overly short answers are unlikely to provide sufficient details to cover the topic, while overly long answers may be too verbose. Text length can be measured at the character, word, and sentence levels (Shah, 2015, Calefato et al., 2016). Text length can also be used to normalise other features that correlate with it. For example, the number of misspellings can be normalised by text length, as a longer document has a higher chance of containing more misspellings than a shorter version.

Low-quality content often has low conformance to common writing practices, thus resulting in low visual quality. Capitalisation, spelling mistakes or typos, out-of-vocabulary (OOV) words, punctuation, and presence of emoticons have been examined in quality assessment (Weimer et al., 2007, Weimer and Gurevych, 2007, Weimer et al., 2007, Agichtein et al., 2008, Wanas et al., 2008, Liu and Agichtein, 2008, Liu et al., 2008, Wang et al., 2009, Toba et al., 2014, Maleewong, 2016, Filice et al., 2016). Here, OOV words can be detected by examining whether a word is in a list of top k frequent words identified from the community or in an independent dictionary or lexical resource, such as WordNet (Baccianella et al., 2010). Consecutive capital letters, a sequence of three or more repeat characters, and words with more than 15 characters have also been studied (Weimer and Gurevych, 2007, Weimer et al., 2007, Wanas et al., 2008, Filice et al., 2016). Similarly, excessive or no punctuation has also been explored (Weimer and Gurevych, 2007, Agichtein et al., 2008, Liu et al., 2008, Wanas et al., 2008, Maleewong, 2016). Elsewhere, under the assumption that answers containing questions are less likely to be of high quality, Wanas et al. (2008) and Mihaylova et al. (2016) explore the number of questions in the content, where questions in the answer can be identified based on question marks, *wh* words, and POS tags of words surrounding a *wh* word.

An answer with a clear paragraph structure is easier to read and more likely to be of high quality than one which combines everything into one paragraph. Tian et al. (2013) explore the maximum paragraph length and average paragraph length. Similarly, Dalip

et al. (2013) explore the number of paragraphs, sections, and subsections, and statistics (e.g., min, max, and average) of section lengths.

Whether an answer contains references, web links, and technical information has been widely studied (Wanas et al., 2008, Shah and Pomerantz, 2010, Arai and Handayani, 2013, Dalip et al., 2013, Roochi et al., 2013, Ponzanelli et al., 2014, Hou et al., 2015, Calefato et al., 2016, Mihaylova et al., 2016). Email addresses can also indicate the trustworthiness of the content, thus the presence and number of email addresses in an answer have also been examined (Ponzanelli et al., 2014, Nicosia et al., 2015, Barrón-Cedeño et al., 2015, Joty et al., 2015, 2016, Suggu et al., 2016a,b, Mihaylova et al., 2016). Weimer and Gurevych (2007), Weimer et al. (2007), Tian et al. (2013), Dalip et al. (2013), and Ponzanelli et al. (2014) examine the presence and volume of code and file-system paths in the technical content, as such information can potentially increase the trustworthiness of the content.

Word usage in the content is measured by examining the existence of specific types of words, mostly via POS. An answer using words from high/low-quality answers is more likely to be of high/low quality. Tran et al. (2015) explore the appearance of specific words that are typical for each answer quality class, where these specific words are highly frequent words extracted from answers belonging to each quality class. As conversational content may not convey much useful information (Heylighen and Dewaele, 2002), Agichtein et al. (2008) use the formality score is based on the number of *more* context-dependent and *less* context-dependent words, where the degree of context-dependence of words is relative to other words. Here, *more* context-dependent words include: pronouns, verbs, adverbs, and interjections; and *less* context-dependent words include: nouns, adjectives, prepositions, and articles. Similarly, Roochi et al. (2013), Hou et al. (2015), and Mihaylova et al. (2016) utilise POS features, such as the number of nouns, verbs, pronouns, and *wh* words. To assess the quality of content in Arabic, Hou et al. (2015) utilise various n-gram features, such as frequent uni-grams and bi-grams, from both the original Arabic and the translated English version. Named entity features, such as the number of person, location, and organisation named entities, have also been examined by Roochi et al. (2013), Hou et al. (2015), and Mihaylova et al. (2016).

Sentiment reflected in word usage has also been explored. An answer containing swear words is likely to be of low quality. Weimer and Gurevych (2007), Weimer et al. (2007), and Mihaylova et al. (2016) explore the percentage of swear words in the answer, where these swear words are from a predefined list identified from WordNet and Wikipedia. Positive sentiment expressed in the answer potentially suggests a good answer. Roochi et al. (2013), Toba et al. (2014), and Balchev et al. (2016) utilise the sum of sentiment scores for words in the answer/comment, where the sentiment score of a word can be obtained through a sentiment lexicon. To measure the subjectivity of content, Mihaylova et al. (2016) explore whether an answer contains a first or third person singular/plural, while Roochi et al. (2013) explore the subjectivity tone and psychological characteristics. Here, subjectivity tone is based on a subjectivity lexicon (Wilson et al., 2005); and psychological characteristics (e.g., swear, sadness, angle, and religion) of a comment is computed using LIWC (Tausczik and Pennebaker, 2010).

Word usage can also be measured by comparing words in the content with those from the community or an independent knowledge source, which is usually measured via KL-divergence between distributions of words (Ponzanelli et al., 2014). To measure how close the word usage of an answer to words used in the community, Gkotsis et al. (2014) and Calefato et al. (2016) use the number of times a word appears in the answer to weight the probability of the word appears in the community corpus. Under the assumption that an answer using words different from the community vocabulary or an independent knowledge source is less likely to be of high quality, KL-divergence is used to measure how an answer's language model diverges from a reference distribution. For example, Agichtein et al. (2008), Liu et al. (2008), Liu and Agichtein (2008), and Shah (2015) explore the KL-divergence between word distributions of an answer and Wikipedia, TREC, and Yahoo! Answers.

To measure comprehension difficulty of the question and answer, readability scores have been studied. Automated readability index (Kincaid et al., 1975), Flesch reading ease score (Kincaid et al., 1975), Flesch-Kincaid grade level (Kincaid et al., 1975), Smog index (McLaughlin, 1969), and Gunning-Fog index (Gunning, 1969) are widely used to measure

the readability of questions and answers (Agichtein et al., 2008, Burel et al., 2012, Chua and Banerjee, 2015, Shah, 2015, Maleewong, 2016, Calefato et al., 2016, Le et al., 2016). To measure the informativeness of content, Dalip et al. (2013) use the information to noise score (Zhu and Gauch, 2000) to measure the informativeness of text, where information to noise is computed as the proportion of stemmed non-stopwords in the text. Similarly, Shah (2015) use the ratio of unique words over the total number of words in the answer.

Question and answer relationship features are extracted from both the question and answer. The aforementioned textual features are extracted from the question or answer alone. In cQA, however, the quality of an answer also depends on its relevance to the given question. An answer providing irrelevant information to the question is of low quality and unlikely to be selected as the best answer. To measure the relevance of an answer to a question, question and answer relationship features are explored by explicit and implicit lexical similarity, syntactic similarity, and semantic similarity. Additionally, web search results, knowledge bases built on historical questions and answers from the same community, and independent knowledge resources are beneficial to measuring the relationship between the question and answer.

Explicit lexical similarity between the question and answer is measured using word and n-gram overlap, cosine similarity in (weighted) vector space, and KL-divergence between the question and answer. Here, we denote this explicit lexical similarity in contrast with alignment-based similarity, as described later. Answer length over question length has been widely explored (Agichtein et al., 2008, Wang et al., 2009, Mihaylova et al., 2016, Mihaylov and Nakov, 2016, Guzmán et al., 2016a,b). There is a high probability that the answer is not related to the question if there are no or few shared words between the question and answer, and as such, the number of shared words (string or n-gram) between the question and the answer has been widely explored (Huang et al., 2007, Wanas et al., 2008, Liu et al., 2008, Liu and Agichtein, 2008, Agichtein et al., 2008, Dalip et al., 2013, Belinkov et al., 2015, Hou et al., 2015, Mohamed et al., 2015, Wu and Lan, 2016, Molino et al., 2016). Similarly, Dalip et al. (2013) and Molino et al. (2016) explore the number of adjectives, nouns, and verbs only present in the answer. Tran et al. (2015) and Le

et al. (2016) exploit the cosine similarity between bag-of-words vectors of the question and answer. Cosine similarity based on *tf-idf* weighed word vector model has also been explored (Belinkov et al., 2015, Hou et al., 2015, Nicosia et al., 2015, Barrón-Cedeño et al., 2015). In contrast, Agichtein et al. (2008) use KL-divergence between the language models of the question and answer to measure the similarity between the question and answer.

There are studies measuring the similarity between the question and answer using alignment-based and translation-based features, referred to as implicit lexical similarity. For example, Tran et al. (2015), Vo et al. (2015), and Wu and Lan (2016) use the sum of alignment scores (from the question to the answer) normalised by the question length. The alignment scores include translation probability and word embedding based scores. The translation probability of one word in the question is achieved by aligning the word in the question with the word in the answer with the highest translation probability; and word embedding based scores are computed in a similar way. Guzmán et al. (2016a) and Guzmán et al. (2016b) explore machine translation based features, including BLEU (Papineni et al., 2002), NIST (Doddington, 2002), TER v0.7.25 (Snover et al., 2006), METEROR v1.4 (Lavie and Denkowski, 2009), unigram precision, and unigram recall, where the question is treated as the reference and the answer is treated as text translation.

Syntactic similarity between the question and answer has been explored through partial tree kernels (Moschitti, 2006). For example, Nicosia et al. (2015), Barrón-Cedeño et al. (2015), Joty et al. (2015), and Joty et al. (2016) apply partial tree kernels over shallow syntactic parses of the question and answer to measure syntactic similarity between the question and answer. Dalip et al. (2013) explore syntactic dependencies and generalised dependencies to measure the similarity between the question and answer. Vo et al. (2015), Filice et al. (2016), and Filice et al. (2017) utilise syntactic generalisation (Galitsky, 2013) to find a set of maximal common subtrees for the pair of question and answer parse trees.

Semantic similarity between the question and answer is computed based on distributed representations or topic representations of the question and answer (Franco-Salvador et al., 2016, Mohtarami et al., 2016, Charlet and Damnati, 2017). Cosine similarity computed over average word embeddings of the question and answer has been widely

explored (Belinkov et al., 2015, Zamanov et al., 2015, Filice et al., 2016, Mihaylov and Nakov, 2016, Filice et al., 2017). Nicosia et al. (2015) and Vo et al. (2015) leverage cosine similarity between the sum of question word vectors and answer word vectors, where word vectors can be LSA trained on the dataset, pre-trained GloVe word embeddings (Pennington et al., 2014), or pre-trained COMPOSES word embeddings (Baroni et al., 2014). Similarly, Tran et al. (2015), Vo et al. (2015), Wu and Lan (2016), and Mihaylov and Nakov (2016) explore cosine similarity between topic vectors of the question and answer, where topic vectors are computed through topic modelling.

Similarity between the question and answer can also be measured by exploiting web search results, knowledge bases built on historical questions and answers in the community, and independent knowledge resources, such as Wikipedia. To provide a broader context for the question and improve similarity measurement between the question and answer, Hieber and Riezler (2011) leverage web search results to enrich the representation of the original question. Specifically, the question is regarded as a query, and snippets of the top search results are used to represent the original question. The similarity between the question and answer is measured by computing the similarity between the answer and snippets. Archives of historical questions and answers can help to bridge the lexical gap between the question and answer. Wang et al. (2009) build a knowledge base from a collection of historical questions and answers to bridge the lexical gap between the question and answer by using a set of relevant question-answer pairs retrieved from this knowledge base.

Question metadata features are extracted from question metadata, which are also beneficial in assessing the quality of answers. The number of answers, comments, and stars received by the question reflect the popularity and quality of the question, which in turn indirectly reflect the quality of the answer. These features are the most widely explored features (Jeon et al., 2006, Liu et al., 2008, Adamic et al., 2008, Wang et al., 2009, Bian et al., 2009, Suryanto et al., 2009, Shah and Pomerantz, 2010, Burel et al., 2012, Yao et al., 2013, Hou et al., 2015, Yao et al., 2015). Elsewhere, Dalip et al. (2013) explore edits related features, such as the number of suggested edits to the question and the number

of suggested edits approved or rejected by the asker. Based on the observation that the quality distribution among different categories varies, the category a question belongs to has been explored in multiple ways (Blooma et al., 2008, Agichtein et al., 2008, Tran et al., 2015, Nicosia et al., 2015, Barrón-Cedeño et al., 2015, Hou et al., 2015, Suggu et al., 2016a,b, Barrón-Cedeño et al., 2016). Ponzanelli et al. (2014) explore the number of tags assigned to the question, while Liu et al. (2008) and Bian et al. (2008) measure the question posting time.

Answer metadata features are extracted from answer metadata, which are usually based on votes and comments that an answer has received. Ratings received by the answer can indirectly reflect its quality. Jeon et al. (2006), Bian et al. (2008), Suryanto et al. (2009), Wang et al. (2009), Burel et al. (2012), Arai and Handayani (2013), Tian and Li (2016), and Calefato et al. (2016) have explored the number of votes an answer or comment has received. There has been work distinguishing positive votes from negative votes. For example, Suryanto et al. (2009), Wang et al. (2009), Blooma et al. (2012), and Arai and Handayani (2013) explore the number of positive and negative votes an answer has received. The number of comments to the answer has also been explored (Burel et al., 2012, Yao et al., 2013, 2015, Tian and Li, 2016). Elsewhere, Dalip et al. (2013) explore edit and comment related features in a detailed way, such as the number of suggested edits or comments to the answer, and the number of users who comment, edit, and suggest edits to the answer. As the number of votes and comments can be accumulated over time, Bian et al. (2008) explore how long since the answer was posted. Under the assumption that users using mobile devices to submit answers or typing answers quickly are more likely to make mistakes, Le et al. (2016) use the device used and typing speed to quantify mistakes in the answer.

Question and answer metadata relationship features are derived from the question and answer metadata, capturing the relationship between the question and answer in an indirect way. A delayed answer is less likely to be selected as the best answer or receive many votes, thus the posting time difference between the question and answer has been widely explored (Liu et al., 2008, Burel et al., 2012, Chua and Banerjee, 2015, Shah, 2015,

Calefato et al., 2016). Tian et al. (2013) measured the posting time difference between the question and answer on a logarithmic scale, as the time lapse between question and answer varies dramatically (e.g., from a few seconds to one year). Assuming that it is unlikely to be an answer if the answer is provided by the asker, whether the answerer and asker are the same user has also been widely explored (Huang et al., 2007, Tran et al., 2015, Belinkov et al., 2015, Hou et al., 2015, Nicosia et al., 2015, Suggu et al., 2016a,b, Mihaylova et al., 2016, Barrón-Cedeño et al., 2016).

2.4.4.2 User Features

User features include user characteristics, user history features, and user network features. User characteristics capture a user's characteristics, such as a user's registration time and their rank in the community. User history features are extracted from their historical behaviours, which are divided into two categories: user historical question and answer features, and user historical vote features, indicating the user's participation in the community in a detailed way. User network features capture the explicit or implicit interactions among users.

User characteristics features indicate the characteristics of a user, mostly relating to the user's personal information as filled in when the user joined the community, and the user's activity level in the community. The longer a user has been in the community, the more likely that they are experienced in providing high-quality questions and answers. Agichtein et al. (2008), Liu et al. (2008), Dalip et al. (2013), and Arai and Handayani (2013) exploit features based on registration time, such as how long since the user registered, and how long since the user wrote their first question or answer. The number of points or merit badges a user owns has also been widely explored (Blooma et al., 2008, Suryanto et al., 2009, Bian et al., 2009, Shah and Pomerantz, 2010, Blooma et al., 2011, 2012, Dalip et al., 2013, Ponzanelli et al., 2014, Le et al., 2016). Molino et al. (2016) explore the description of the user, such as the presence of a resume, self-description of the user, and the profile picture, while Roochi et al. (2013) explore the number of contacts in the user profile.

User history features are based on the history of a user asking questions, answering questions asked by other users, receiving answers from others to questions asked by themselves, user's votes on other users' historical questions and answers, and other users' votes on the user's historical questions and answer. Here, we divide user history features into two categories: user historical question and answer features, and user historical vote features. User historical question and answer features are extracted from questions asked by the user, answers provided by the user, and answers received from others. To emphasise that the best answers are selected by the asker or through voting, features based on the best answers are classified into user historical vote features, although they can also be classified into user historical question and answer features. Among user's vote features, there are different types of votes in cQA, such as votes for best answer, stars for good questions, thumbs up, thumbs down, abuse, and spam reports. All these votes are treated as user votes without specifying which platform it belongs to, although some types of votes are specific to certain platforms. Comments, ratings, and edits to the questions and answers provided by the user are also classified as user historical vote features.

User historical question and answer features are extracted from a user's history of asking, answering, and receiving answers from others. A user's asking and answering history can indirectly indicate the quality of their future posted content. The number of (resolved) questions asked by the user and answers provided by the user, and the ratio of answers to questions have been widely explored (Jeon et al., 2006, Huang et al., 2007, Liu and Agichtein, 2008, Liu et al., 2008, Bian et al., 2008, Shah and Pomerantz, 2010, Burel et al., 2012, Yao et al., 2013, Arai and Handayani, 2013, Yao et al., 2015, Maleewong, 2016, Le et al., 2016). As the number of questions asked by the user is accumulated over time and it is not fair to compare a newly registered user with an older user, Burel et al. (2012) and Molino et al. (2016) explore post rate, measured by the average number of questions or answers a user posts per day. Agichtein et al. (2008) and Dalip et al. (2013) have exploited features in a detailed way, such as the average length of questions asked by the user, and the average number of answers to questions asked by the user, while

Dalip et al. (2013) explore ranking features, such as the ranking of a user according to the number of questions and answers they have posted in a category or in the community.

A user's activity and reputation measured within a category or community also reflect the content quality posted by the user. A user asking many questions does not necessarily indicate their ability to provide high-quality answers. Burel et al. (2012) have measured the predictability (e.g., asking questions or providing answers) of a user's activity via an entropy score computed on historical questions and answers provided by the user. The number of good or bad answers provided by the user has also been explored (Mihaylova et al., 2016, Suggu et al., 2016a,b). Users can have different expertise levels with regards to different topics. Burel et al. (2012), Roochi et al. (2013), and Molino et al. (2016) measure user's topical reputation via an entropy metric based on the tags associated with questions or categories of the post, while Jeon et al. (2006) and Molino et al. (2016) explore user's category speciality measured by the number of questions and answers that the user has posted in that category.

User historical vote features are based on other users' votes on questions and answers provided by the user, and the user's votes on answers to the question asked by the user. The number (ratio) of the best answers a user has provided has been widely explored (Jeon et al., 2006, Bian et al., 2008, Blooma et al., 2008, Agichtein et al., 2008, Suryanto et al., 2009, Shah and Pomerantz, 2010, Burel et al., 2012, Arai and Handayani, 2013, Maleewong, 2016). Similarly, Agichtein et al. (2008) have explored votes on answers provided by a user, such as the average number of thumbs up answers written by the user have received, and the average number of stars to questions asked by the user. Ponzanelli et al. (2014) explore the number of badges a user has received by providing good answers or asking good questions, an indicator of receiving votes from other users, while Bian et al. (2008) explore the average number of thumbs up received by answers to questions asked by the user. The number of thanks, warnings, and spam reports have also been explored (Le et al., 2016, Agichtein et al., 2008, Ponzanelli et al., 2014).

User network features are extracted from explicit and implicit interactions among users. Explicit interactions reveal the social connectivity of users, such as making friends

with other users. For example, Le et al. (2016) have measured the number of neighbours, and the average degree of their neighbours. Shah (2015) utilise the number of distinct interactions between the asker and answerer, measured by the number of provided and received answers between the asker and answerer. Jurczyk and Agichtein (2007), Suryanto et al. (2009), Bian et al. (2009), and Dalip et al. (2013) exploit asking expertise and answering expertise features, computed via the HITS algorithm (Kleinberg, 1999). The asking expertise (hub value) of a user is derived from the answering expertise of other users in answering questions posted by the asker, and the answering expertise (authority score) of a user is derived from the asking expertise of other users in posting questions.

There have also been studies exploring implicit user relationships in a detailed way, treating asking, answering, and voting relationship differently. Agichtein et al. (2008) construct a graph $G = (V, E)$ based on the interactions between users, where V consists of a set of users and E consists of sets of relationship between users. They identify five types of relationship between users: E_a, E_b, E_s, E_v, and $E_{+/-}$. Here, E_a indicates that the user u answers at least one question asked by the user v; E_b indicates that the user u provides at least one best answer to questions asked by the user v; E_s indicates that the user u gives a star to at least one question asked by the user v; E_v indicates that the user u votes for at least one answer written by the user v as the best answer; and $E_{+/-}$ indicates that the user u gives a thumbs up/thumbs down to an answer written by the user v. Then hub scores and authority scores of vertices V are computed via HITS, which are used to predict the content quality. Similarly, Molino et al. (2016) explore in-degree, PageRank (Page et al., 1998), and HITS authority of a user in constructed graphs. Lui and Baldwin (2009) have also studied author network based features, where nodes are authors, and there is an un-directed edge between authors if these two authors have posted in the same thread at least twice.

2.4.4.3 Thread Level Features

When assessing the quality of an answer, the quality of competing answers within the same thread should also be taken into account. The comparison can be based on the

quotation relationship among answers (such as one answer quoting another answer to support itself), surrounding answers, similarity between answers, answer position, and interaction among users who have posted answers in the same thread.

Quoted content from other answers can provide evidence for the answer of interest. Wanas et al. (2008) explore features based on quotation chunks, such as the number of quotation chunks from earlier posts. Assuming that an answer is more likely to be quoted if the quality of that answer is high, Huang et al. (2007) and Wanas et al. (2008) measure the number of times a given post is quoted, while Huang et al. (2007) explore whether a post quotes the question title and other answers.

Surrounding answers have also been studied as they can signal the quality of the target answer. For example, Chen et al. (2018) perform sentence matching between the comment of interest and surrounding comments to assess the quality of Chinese news comments from a novelty perspective. Xiang et al. (2017) feed a sequence of CNN learned question-answer pair representations into a Bi-LSTM to model answer correlations. Suggu et al. (2016a) and Suggu et al. (2016b) have explored whether the answerer or asker has commented before or after the answer of interest and whether the comment is a question. Assuming that an acknowledgement or further questions by the asker in the thread signal a good answer, previous studies (Barrón-Cedeño et al., 2015, Nicosia et al., 2015, Joty et al., 2015, Filice et al., 2016, Joty et al., 2016, 2018) have also explored whether an answer appears in the proximity of a post provided by the asker. Assuming that a subsequent answer usually provides support or argument to precedent answers from the same user, Huang et al. (2007) measure the distance between two adjacent answers provided by the same user.

The answer of interest can be compared with other answers in the same thread, via similarity between answer pairs and thread level normalisation. Compared to other answers from the same thread, a high-quality answer should contain a balanced ratio between similarity and originality: an answer with a high originality value indicates that the answer is less likely to be relevant to the question; and an answer with a high similarity score indicates that the answer is less likely to provide useful information. Tian et al. (2013)

use cosine similarity to measure the similarity between the answer of interest and other answers to the same question, where cosine similarity score is computed over a vector space uni-gram model weighted by *tf-idf*. Wanas et al. (2008) measure the maximum degree of word overlap between the answer of interest and preceding answers and the number of posts between them. To model the semantic correlations among answers, Zhou et al. (2018) feed a sequence of question-answer pair representations into a BI-LSTM model to select the best answer, where the question-answer pair representation is obtained by concatenating the CNN learned question and answer representations.

Answers from the same thread can also be used to normalise features extracted from the answer of interest (Wanas et al., 2008, Lui and Baldwin, 2009, Burel et al., 2012). For example, Wanas et al. (2008) and Lui and Baldwin (2009) normalise the length of the post of interest by the average length of posts within the same thread. There has been work exploring different ways of normalisation. For example, Burel et al. (2016) explore locally min-max normalisation, order normalisation, and normalised order normalisation. Similarly, Gkotsis et al. (2014) and Calefato et al. (2016) normalise textual features based on order. One concrete example of order normalisation for answer length is that the longest answer in the thread ranks 1, as a longer answer is generally more accurate, elaborate, and has a higher chance of being high quality or selected as the best answer. Guzmán et al. (2016a) and Guzmán et al. (2016b) use a neural machine translation evaluation architecture to compare two answers in response to the same question. Two answers from the same thread play the role of two competing translation hypotheses and the question acts as the translation reference, and the ranking of these two answers in terms of quality is determined in a machine translation scenario.

Answer position in the answer list and feature-value based rank (the rank of a feature value) have been widely studied (Wang et al., 2009, Shah and Pomerantz, 2010, Burel et al., 2012, Tian et al., 2013, Nicosia et al., 2015, Barrón-Cedeño et al., 2015, Joty et al., 2015, 2016, Suggu et al., 2016a,b, Joty et al., 2018). Answer position in the thread reflects the posting sequence of the answer, which indirectly affects the user perception of the answer quality. Shah and Pomerantz (2010), Guzmán et al. (2016a), and Guzmán et al. (2016b) use

reciprocal rank of the answer in the answer list to represent the answer position, while Nicosia et al. (2015), Barrón-Cedeño et al. (2015), and Joty et al. (2015) use $\max(20, i)/20$ to represent the answer position, where i is the position of the answer in the thread sorted by posting time and the value of 20 is arbitrarily chosen. Instead of paying attention to the visual position, Burel et al. (2016) explore featured-value based rank, which is computed as follows: values for a certain feature across all answers within the same thread are extracted; and the rank of a value (in decreasing order) is used as the feature-value based rank. Similarly, Belinkov et al. (2015) employ SVM Rank over textual and non-textual features to obtain rank scores, which are used to predict the quality of answers.

User behaviour at the thread level has also been examined, usually measured through the number of answers a user posts within the same thread and posting interactions among users if the community allows users to post multiple answers in a thread. Observing that most of the posts will be non-informative and irrelevant to the original question if a user has a large number of posts, Tran et al. (2015), Barrón-Cedeño et al. (2016), and Mihaylova et al. (2016) explore the number of posts from the same user within one thread, while Nicosia et al. (2015) use this feature to measure whether a user is particularly active in a thread. Interlacing comments between two users are potential dialogues, indicating low-quality answers. Barrón-Cedeño et al. (2015), Nicosia et al. (2015), Joty et al. (2015), and Joty et al. (2016) identify conversation chains, which are used to extract features, such as whether the comment is at the beginning, middle, and end of the chain. To measure how responsive a user is, Wanas et al. (2008) explore the speed of replies, measured as the time difference between this post and its direct precedent post normalised by the average posting time in the same thread.

2.4.4.4 Usage Features

Usage features include usage statistics of an answer or a thread, such as the number of clicks or page views a thread has received and user dwell time. A large number of clicks results in a high probability of exposure and improvement, thereby leading to an answer of high quality. The number of clicks has been widely explored (Jeon et al., 2006, Agichtein

et al., 2008, Chai et al., 2010, Burel et al., 2012, Dalip et al., 2013). To eliminate spam, Chai et al. (2010) distinguish the total number of views from the total number of distinct user views. To alleviate topical and genre category influence, Agichtein et al. (2008) explore normalised click statistics, such as the average number of views for a given category, and the number of clicks over views. Similarly, the number of print counts, copy counts, and users' recommendation and dis-recommendation have also been explored by Jeon et al. (2006) and Chai et al. (2010).

2.4.5 Discussion

Most hand-crafted features extracted from documents are applicable to measure the quality of documents for all tasks discussed in our work. For example, text length is the most widely explored feature. The presence of misspellings, grammatical errors, and whether the content conforms to the common writing practices are generalised features to measure various types of documents. Misspellings can be identified by comparing words with a predefined dictionary, if the predefined dictionary is large enough; otherwise, n-gram and POS n-gram features can be used to identify misspellings.Word usage and various readability scores are also applicable to measure the quality of documents studied by all these tasks.

There are also features only applicable to specific tasks. For example, the presence or the number of references, images, and URLs can provide additional information for documents, and these features can be used to measure the quality of Wikipedia articles, academic papers, and answers in cQA, but not student essays, as it is uncommon to include references, images, and URLs when students write essays. For example, the presence of emoticons, a sequence of three or more repeat characters, and the number of words longer than 15 characters can only be used to measure content quality in cQA as content from cQA is relatively informal, and the community platform supports the usage of such words and emoticons. Thread level features, such as the similarity between the answer of interest and remaining answers within the same thread are only applicable to measure content quality in cQA. Features extracted from users/contributors' metadata

are applicable to the tasks of quality assessment of Wikipedia articles and content quality assessment in cQA, as this type of information is available for these two tasks. Features used to measure the similarity between an essay and its corresponding prompt can also be adopted to measure the similarity between an answer and its corresponding question, and vice versa.

These hand-engineered features specific to each domain can be used as side-information by concatenating with document representations, which will be described in Sections 4.1 and 4.2.

2.5 End-to-End Models

A variety of neural models have been proposed to tackle these tasks, such as CNNs and bi-directional LSTMs (Bi-LSTMs) (Hochreiter and Schmidhuber, 1997). In this section, we outline neural models explored in different domains from Section 2.5.1 to Section 2.5.4. Then we present a discussion of neural models adopted to assess document quality across different domains in Section 2.5.5.

2.5.1 Quality Assessment of Wikipedia Articles

There have been studies dealing with the quality assessment of Wikipedia articles in an end-to-end fashion. For example, Dang and Ignat (2016c,b) use a distributed memory version of Paragraph2Vec (Le and Mikolov, 2014) to obtain document representations, which are fed into a four-layer feedforward neural network to classify articles based on their quality. Dang and Ignat (2017) exploit two stacked Bi-LSTMs to predict the quality of a Wikipedia article, where the first layer Bi-LSTM learns sentence representations and the second layer Bi-LSTM learns document representations. Wang et al. (2019) and Wang and Li (2020) compare the performance of different neural models, such as CNN and Bi-LSTM, in differentiating high quality articles from lower quality ones, within a feature framework, where hand-crafted features are input to such neural models.

There have also been studies applying neural models over the history of Wikipedia articles to predict their quality. For example, Marrese-Taylor et al. (2019) propose an edit-centric approach to predict quality of revisions for an article. Specifically, the differences between two consecutive versions are fed into a BI-LSTM to classify the quality of an updated version, where the classifier is equipped with an auxiliary model which generates a description for the edit, modelled in a multi-task learning setting. Zhang et al. (2018) apply a BI-LSTM to a sequence of features extracted from metadata and content of each revision of an article, where the last hidden representation from the metadata is concatenated with the document representation learned by another BI-LSTM, to predict the quality of an article.

2.5.2 Academic Paper Rating

Previous studies have rated papers in a modularised way (Yang et al., 2018, Qiao et al., 2018, Leng et al., 2019). For example, Yang et al. (2018) exploit a modularised hierarchical CNN model to predict the acceptance of a paper, where each paper section is treated as a module. An attentive pooling layer is applied to the concatenated representations of sections to obtain paper representations, where section representations are obtained by attention-based CNNs. Similarly, Qiao et al. (2018) feed a sequence of section representations to a BI-LSTM to obtain paper representations, which are used to predict aspect scores. Leng et al. (2019) obtain section representations from semantics, grammar, and novelty views. For each section, a hierarchical bidirectional gated recurrent units (GRU) (Cho et al., 2014) is used to capture semantics; an unsupervised grammaticality model pretrained on Wikipedia is used to capture grammar; a stacked de-noising auto-encoder is used to obtain the textual representations; and the number of papers around this representation's unit Euclidean distance is used to measure novelty. Then an attention layer is applied to weight the importance of these three aspects to predict acceptance of papers. Elsewhere, Huang (2018) uses ResNet-18 (He et al., 2016) to learn visual features from visual renderings of papers, which are used to predict whether a paper in the computer vision area is a conference or workshop paper.

Sentiment analysis over review text or the combination of sentiment analysis and paper content has been explored to predict acceptance of papers (Wang and Wan, 2018, Kang et al., 2018, Ghosal et al., 2019d,c). For example, Wang and Wan (2018) predict acceptance of papers by analysing the sentiment in the review text. They propose a multiple instance learning network to predict the sentiment distribution of each sentence in the review text, where sentiment distributions of all sentences in the review text are aggregated to predict the acceptance of a paper. Furthermore, they adopt an abstract-based memory mechanism to identify sentences with sentiment polarities. Similarly, Ghosal et al. (2019d,c) extract sentiment scores, such as positive, negative, and neutral, from each review sentence using sentiment analyser (Hutto and Gilbert, 2014), where these scores are used in combination with paper and review text representations obtained by applying an universal sentence encoder (Cer et al., 2018) to predict aspect scores and acceptance of papers jointly.

When assessing whether a submission is within the scope of a given venue, paper content, bibliography, and image information have been used to make predictions. For instance, Ghosal et al. (2019b) propose a multi-channel multi-modal neural network to predict whether a paper is an out-of-scope submission, where there are three channels of the paper information, namely, the paper text, bibliography, and images in the paper. For the paper text channel, a universal sentence encoder (Cer et al., 2018) is applied to obtain sentence representations, which are fed into a Bi-LSTM to obtain paper representations; for the bibliography channel, a bag-of-words model is applied to the bibliographic paper titles and venues to obtain bibliographic representations; for the image channel, VGG-16 (Simonyan and Zisserman, 2015) is applied to images in the paper to obtain image representations. Representations learned from these three channels are used to predict whether a paper is within the scope of a target venue.

2.5.3 Essay Scoring

End-to-end models have also been proposed to deal with essay scoring. For example, Taghipour and Ng (2016) explore the utility of recurrent networks, such as basic recurrent

units (RNN) (Elman, 1990), GRUs, and long short-term memory units (LSTMs) to score essays, with experimental results indicating that an LSTM with an average-pooling layer achieves the best performance. To learn the relative weights of words and sentences automatically, Dong et al. (2017) incorporate an attention mechanism into a hierarchical network structure to obtain sentence and essay representations. Alikaniotis et al. (2016) enhance a word embedding model by incorporating the score prediction into word embedding training loss, which is able to capture both local contextual and word usage information, such as the extent of specific words contributing to the essay score. Then the resulting word embeddings are fed into a Bi-LSTM, whose hidden representation at the last step is used to predict the essay score. Cummins and Rei (2018) use a Bi-LSTM to score essays, coupled with the task of grammatical error detection in a multi-tasking learning setting, showing that grammatical error detection can enhance essay scoring.

There have been studies (Tay et al., 2018, Li et al., 2018, Mim et al., 2019) focusing on capturing cohesion and coherence to improve model performance in essay scoring. Cohesion (local coherence) and coherence are essential criterion in essay scoring, which can not be fully captured by neural models, limiting the effectiveness of neural models in predicting essay scores. Thus, models are proposed to measure cohesion and coherence of essays. For example, Mim et al. (2019) propose to use a logical sequence-aware model to score essays. To train such a model, both coherent and incoherent documents are required, where coherent documents are the original essays and incoherent documents are generated by randomly shuffling sentences or discourse indicators (such as *however* and *in my opinion*) in a document. Then the resulting pre-trained model is used as an encoder to obtain document representations to perform essay scoring. Tay et al. (2018) propose to capture local coherence by computing the relevance of two LSTM outputs, which they concatenate with essay representations to score essays. Li et al. (2018) use multi-head self-attention to capture long distance relationships between words, which are passed to an LSTM layer to estimate coherence scores of essays.

There has also been work learning the grading schema, which in turn contributes to essay scoring. For example, to take the rating schema into consideration, Wang et al.

(2018b) propose a reinforcement learning framework with a Bi-LSTM to learn representations of essays, where the evaluation metric QWK is incorporated as the reward and one of the components of the loss function. Similarly, Zhao et al. (2017) use graded essays with different scores to represent the grading criteria specified in the rubric, which are stored in the memory component. The relevance between an ungraded essay and each graded essay in the memory is computed, which is used to score essays.

To develop prompt-independent essay scoring models, where rated essays for a given prompt are unavailable, Jin et al. (2018) propose a two-stage learning framework. Specifically, prompt-independent features are used to select essays for the given prompt with predicted extreme scores, on which a hybrid neural network incorporating semantic, part-of-speech, and syntactic signals is built to score essays responding to a different prompt.

2.5.4 Content Quality Assessment in cQA

Neural models have been studied to deal with content quality assessment in cQA (Wu and Lan, 2016, Fang et al., 2016, Mohtarami et al., 2016, Tan et al., 2016, Zhang et al., 2017). For example, Suggu et al. (2016a) incorporate document representations learned through a CNN with hand-crafted features to predict the quality of answers. Zhou et al. (2015a) and Zhou et al. (2015b) adopt a two-dimensional CNN to learn the semantic relevance between the question and answer pair, which is a distributed representation of each question-answer pair. Then an LSTM is used to model the semantic correlations of answer sequences, aiming to take answer context into consideration. Textual and non-textual features require different types of representations and the correlations between them are highly nonlinear. Hu et al. (2013) leverage a multimodal deep belief net to learn a joint representation from textual and non-textual features. Elsewhere, Li et al. (2019) measure the performance of BERT at answer selection, demonstrating the effectiveness of BERT over this task.

To capture important information in the question and answer, attention mechanisms have been used (Tan et al., 2016, Hsu et al., 2016, Suggu et al., 2016b, Deriu and Cieliebak,

2017, Zhang et al., 2017, Xiang et al., 2017, Wu et al., 2018b). For example, Wu et al. (2018b) propose a multi-dimensional attention mechanism to capture the interactions between the question subject-body pair and question-answer pair, where the question body information is incorporated with the question subject via similarity and disparity. Here, similarity reflects the shared information between the question subject and body; and disparity reflects additional information provided by question body. Deriu and Cieliebak (2017) adopt a Siamese CNN architecture to assess the quality of answers, where two CNNs are used to obtain the question and answer representations respectively. To capture important information in the question and answer, two attention mechanisms are applied to the input and output of CNNs. Assuming that neighbouring words in the answer should be given higher weights if a word appears in both the question and answer, Chen et al. (2017) propose a positional attention model, utilising the position information of shared words.

There has been work assessing the quality of answers in a multi-task setting (Joty et al., 2018, Yang et al., 2019a, Deng et al., 2019, 2020). For example, Yang et al. (2019a) use question categorisation as an auxiliary task, in addition to the primary task of assessing the quality of answers, aiming to learn a category-aware model. With the assumption that generated answer summaries can reduce redundancy and lengthiness of answers of interest, Deng et al. (2020) propose a joint learning framework consisting of two tasks: rating answers and generating answer summaries for the given question and answer pairs. Similarly, Deng et al. (2019) propose to deal with the tasks of answer selection and knowledge base question answering in a multi-task framework. Elsewhere, Yang et al. (2019a) incorporate external commonsense knowledge from a knowledge base into question and answer representation learning to capture important entities and their relations in the question and answer.

Other strategies have also been studied in assessing the quality of answers in cQA (Xiang et al., 2017, Joty et al., 2018, Yang et al., 2019b), such as adversarial training and conditional random fields (CRFs) (Lafferty et al., 2001). Specifically, Yang et al. (2019b) employ a generative model to sample challenging low-quality answers to train a robust

classification model, where low-quality answers are constructed from low-quality answers for the question of interest and answers for other questions. Observing that answer quality predictions depend on not only the answer representation of interest but also label transitions among answers, Xiang et al. (2017) apply a CRF layer to predict the quality for a sequence of answers. Conversely, Joty et al. (2018) adopt a fully-connected CRF to model the dependencies between three relevant tasks, where one of these tasks is to assess the quality of answers.

The relative performance of BERT, Bi-LSTM, and CRF depends on the data/resource setting. In general, for high-resource settings, where pre-trained transformers are available or there is a large number of instances to pretrain transformers from scratch, BERT generally performs well; for low-resource settings or domain-specific tasks, where a pre-trained transformer is unavailable and there is a limited number of instances to pretrain transformers from scratch, Bi-LSTM is a good choice; and CRF is good for tasks where there is structural information between target labels or there is dependency between target labels.

2.5.5 Discussion

Hierarchical neural models have been proposed to tackle tasks in different domains, especially for long documents, such as Wikipedia articles, academic papers, and essays. For Wikipedia articles and essays, a CNN or Bi-LSTM is applied to obtain sentence representations, which are fed into another Bi-LSTM layer to predict the quality of documents (Dang and Ignat, 2017, Dong et al., 2017). For academic papers, section representations are learned by CNNs or Bi-LSTMs, and concatenated section representations are fed into a pooling layer to obtain document representations to predict the acceptance of papers (Yang et al., 2018, Qiao et al., 2018, Leng et al., 2019). The modularised idea has to be customised if applied to Wikipedia articles as the number of sections in Wikipedia articles varies greatly. For answers in cQA, which are relatively shorter than the aforementioned documents, a flat model can be used to learn document representations (Wu and Lan, 2016, Fang et al., 2016, Tan et al., 2016, Zhang et al., 2017).

Attention mechanisms have been incorporated into neural models across different domains, to select salient information from documents, although there are variations among the specific attention mechanisms adopted. For example, in academic paper rating, Yang et al. (2018) focus on attention-based CNNs to obtain section representations, and concatenated section representations are fed into an attentive pooling layer to obtain document representations to predict the acceptance of papers. Similarly, to learn the relative weights of words and sentences automatically, Dong et al. (2017) incorporate an attention mechanism into a hierarchical network structure to learn the relative weights of words and sentences contributing to essay scores. For answers in cQA, Deriu and Cieliebak (2017) apply two attention mechanisms to the input and output of a Siamese CNN architecture to capture important information in the question and answer.

Multi-task learning can be adopted to assess documents from different domains, although there is no such existing work on quality assessment of Wikipedia articles. For example, for academic papers, the main task of predicting acceptances of academic papers can be trained jointly with the task of paper aspect score prediction. For essays, the main task of essay scoring can be learned simultaneously with the task of grammatical error detection. For answers in cQA, the primary goal of predicting answer quality can be learned together with the task of question category classification. The choices of auxiliary tasks depend on not only the task of interest but also their utility in contributing to the primary task, usually verified empirically.

2.6 Document Coherence Measurement

A document of high quality is coherent, indicating that two consecutive sentences are locally well-connected (*cohesion*) and multiple sentences in a paragraph/document globally hold together (*coherence*). Automated document coherence measurement systems or approaches assess documents of any type and written in any language automatically via extracting features from documents, either hand-engineered or learned automatically, where features are obtained at both intra-sentential and inter-sentential levels to reflect

transitions of the logic flow. The relationship between features and coherence scores is learned by a document coherence measurement system, and the trained system is used to predict coherence score of a given document. The performance of a document coherence measurement system is compared to human annotated coherence scores for documents written by humans/machines or gold rankings for documents generated by manipulating operations over coherent documents.

In this section, we first define the task of document coherence measurement in multiple application scenarios and how previous work models it (Section 2.6.1), followed by evaluation metrics in measuring the performance of systems in different application scenarios (Section 2.6.2). Next, hand-crafted features proposed by previous work in multiple application scenarios are summarised in Section 2.6.3. Lastly, end-to-end neural models, either task-specific or task-agnostic, are outlined in Section 2.6.4, noting that there is some overlap with what was discussed in Section 2.5.3.

2.6.1 Task Definition and Modelling

Traditionally documents are written by either native speakers or second language learners or generated by machines (such as machine translation, and summarisation systems), where the coherence of documents is not guaranteed. The performance of systems or approaches measuring the coherence of documents is compared with human annotated coherence scores or an implicit indicator, such as the overall score for an essay acting as a proxy for essay coherence. For such documents, incoherence in these documents occurs due to people/machines' limited ability to output coherent documents. It is also possible to generate incoherent documents by shuffling sentences within a document or inserting text from elsewhere to a coherent document.

In what follows, we first describe related tasks and how they have been modelled in assessing the coherence of documents written by humans/machines, before moving on to discuss the coherence of documents, where incoherent documents are generated by performing manipulations over coherent documents. Finally, we describe peripherally re-

lated tasks and how they have been modelled, such as disfluency detection in spontaneous speech transcription and code-switching detection.

Tasks which assess the coherence of human-written or machine-generated documents include summary coherence rating, readability assessment, and essay scoring. Summary coherence rating assesses the coherence of an automatically generated summary of a document; readability assessment measures whether and to what degree a document is easy to read and understand; and essay scoring scores an essay written by a language learner. Both summary coherence rating (Barzilay and Lapata, 2005, 2008, Guinaudeau and Strube, 2013, Nguyen and Joty, 2017) and readability assessment (Clercq and Hoste, 2016, Mesgar and Strube, 2016, 2018) are modelled as a pairwise ranking problem. Specifically, a prediction is considered correct if the predicted pairwise ranking is the same as that elicited from humans. Essay scoring is generally modelled as a regression problem (Somasundaran et al., 2014, Mesgar and Strube, 2018, Tay et al., 2018).

Tasks proposed to assess model prediction with respect to the coherence of documents include sentence reordering detection, sentence insertion, and paragraph reconstruction, where incoherent documents examined by these tasks are generated by manipulating inherently coherent documents. The task of sentence reordering is to score a document and its permuted versions at the sentence level in terms of their coherence, where the original document is considered to be well-written and coherent, and assigned a higher coherence score than those of its permutations; the task of sentence insertion is to find the correct position to insert a sentence back to a document from which the sentence was taken; and the task of paragraph reconstruction is to recover the original sentence order of a shuffled paragraph given its starting sentence. Incoherent documents used to measure the performance of models are generated by shuffling a single sentence or all sentences of originally coherent documents. Both the task of sentence reordering detection (Barzilay and Lapata, 2005, Elsner et al., 2007, Barzilay and Lapata, 2008, Elsner and Charniak, 2011, Guinaudeau and Strube, 2013, Nguyen and Joty, 2017, Li and Jurafsky, 2017, Mesgar and Strube, 2018) and the task of sentence insertion (Elsner and Charniak, 2011, Putra and Tokunaga, 2017, Nguyen and Joty, 2017, Xu et al., 2019a) are modelled as a

ranking problem. For the task of sentence reordering detection, given a pair of documents consisting of one original document and one permuted version, a prediction is considered to be correct if the score for the original document is higher than the permuted one. For the task of sentence insertion, a prediction is considered to be correct if it assigns the highest coherence score to the text formed by inserting the sentence of interest at the sentence's original position. The task of paragraph reconstruction is tackled in a generative way (Lapata, 2003, Elsner et al., 2007, Li and Jurafsky, 2017, Xu et al., 2019a). Specifically, at each step, the coherence score of placing each remaining sentence to the right of the partially constructed document is computed, which serves as the selection mechanism for beam search to drive the reconstruction process.

Peripherally related tasks have also been proposed in the literature, including disfluency detection in spontaneous speech transcription and code-switching detection. However, these tasks are not designed to measure document coherence or to understand models' performance in document coherence understanding, although phenomena in the document, such as reparandum and language-switching with a sentence/document, can affect the coherence of documents to a certain degree. The task of disfluency detection in spontaneous speech transcription is to detect the *reparandum* and *repair* in spontaneous speech transcriptions to make the text fluent by replacing the *reparandum* with *repair*. An example of disfluent speech text is: *please give me your pen, I mean your pencil.* Here, the *reparandum* is *pen* and the *repair* is *pencil.* The task of code-switching detection is to assign a language label to each word in the document. Code-switching often occurs in bilingual/multilingual communities, which alternate between languages/dialects within a conversation, sentence, or even word. The task of disfluency detection is modelled as a classification problem (Lou et al., 2018). Specifically, the task is to predict whether each word in the document is a *reparandum.* The task of code-switching detection is either formulated as a sequence tagging problem (Chittaranjan et al., 2014, Lin et al., 2014, Barman et al., 2014a, Samih and Maier, 2016, Yirmibesoglu and Eryigit, 2018, Mave et al., 2018, Adouane et al., 2018a) or classification problem (Das and Gambäck, 2014, Barman et al., 2014a, Gella et al., 2014, Barman et al., 2014b, Piergallini et al., 2016).

2.6.2 Evaluation Metrics

Although the aforementioned tasks have been explored in different application scenarios, the same evaluation metrics can be adopted to assess performance of models targeting different tasks. For example, for the tasks of summary coherence rating, readability assessment, and sentence reordering detection, accuracy is widely used to measure model performance. For sentence insertion, an evaluation metric similar to accuracy is adopted, which is computed as the average fraction of sentences per document reinserted into their original position. For the tasks of disfluency detection and code-switching detection, precision, recall, and F_1 are used to evaluate the performance of models. For the task of essay scoring, quadratic weighted kappa is used to measure models' performance. For the task of paragraph reconstruction, Kendall's τ is used to measure models' performance to avoid penalising ordering that maintain relative position between sentences, where local coherence can be kept. Kendall's τ is a non-parametric measure of relationships between two lists of ranked data, computed as:

$$\tau = 1 - \frac{2\#\text{of inversions}}{N \times (N-1)} \tag{2.9}$$

where N is the number of sentences in the documents, and inversions denote the number of interchanges of adjacent sentences to reconstruct the original document.

2.6.3 Hand-crafted Features

Unlike our original four tasks, features used in document coherence measuring are solely based on document content.

In this section, we first describe features engineered to capture local and global coherence in different application settings, including sentence reordering detection, sentence insertion, paragraph reconstruction, summary coherence rating, readability assessment, and essay scoring. Then hand-crafted features proposed to tackle the task of code-switching detection are described.

There have been studies exploring entity matrices or graphs, based on shared entities between sentences, to assess local coherence (Barzilay and Lapata, 2005, Pitler et al., 2010, Louis and Nenkova, 2012, Elsner and Charniak, 2011, Guinaudeau and Strube, 2013, Mesgar and Strube, 2016). For example, Barzilay and Lapata (2005, 2008) use an entity matrix to represent entity transitions across sentences, which reflect distributional, syntactic, and referential information about entities in the document. Then the probabilities of transitions between adjacent sentences are used as features for downstream processing. Later, Elsner and Charniak (2011) have extended the entity grid with entity-specific features, such as syntax features, entity types, and coreference features, to differentiate important entities from less important entities. Elsewhere, Elsner et al. (2007) and Elsner and Charniak (2008) have combined the local and global features to capture both local and global coherence, where local features are captured by an entity-based approach and global features are captured by a hidden Markov model. Pitler et al. (2010) have proposed the use of cosine similarity between sentences, coreference information (such as the number of times that an antecedent for a pronoun is found in the previous sentence, in the same sentence, or neither), and summarisation specific features (such as the number of times that the same sentence appears before a cohesive device in both the summary and source documents), to measure local coherence. Here, cohesive device includes discourse connectives, such as *but* and *because*. Louis and Nenkova (2012) have proposed the use of syntax regularities between sentences adjacent to "communicative goal regularities", to differentiate coherent documents from incoherent ones. Guinaudeau and Strube (2013) and Mesgar and Strube (2016) use a graph to model entity transition sequences, where nodes are sentences. Two nodes are connected if they share at lease one entity or contain a pair of similar words based on cosine similarity, where edges between these two nodes can be weighted based on the number of entities shared, the syntactic roles of the entities, or the distance between two sentences representing these two nodes. Then the average out-degree of this graph is used as the local coherence score or the frequencies of all k-node subgraphs are input to a classifier, such as a random forest.

Language dependent and independent features have been proposed to tackle the task of code-switching detection (Chittaranjan et al., 2014, Das and Gambäck, 2014, Barman et al., 2014b, Piergallini et al., 2016, Samih and Maier, 2016, Samih et al., 2016, Yirmibesoglu and Eryigit, 2018, Mave et al., 2018). For example, Chittaranjan et al. (2014), Mave et al. (2018), and Yirmibesoglu and Eryigit (2018) use both language independent features, such as character/word n-grams, length of words, and contextual information, and language dependent features, such as lexicons, monolingual part-of-speech taggers, and features designed especially for a language. These features are input to CRFs when the task is modelled as a sequence tagging problem, and classifiers, such as SVM, or logistic regression, when the task is modelled as a classification problem. A comprehensive summary can be found in Solorio et al. (2014) and Molina et al. (2016), which summarise participating systems in the first and second shared tasks on language identification in code-switched data, respectively.

2.6.4 End-to-End Models

Neural network models have been proposed to measure local and global coherence (Nguyen and Joty, 2017, Li and Jurafsky, 2017, Mesgar and Strube, 2018, Logeswaran et al., 2018). For example, in addition to entity-specific features, Nguyen and Joty (2017) apply a convolutional neural network to the entity grid to capture long-range entity transitions. On the other hand, Li and Jurafsky (2017) propose to detect the coherence at a clique level, where a clique is a sequence of consecutive sentences in a text. Assuming the original text is coherent, they obtain a positive clique by selecting a sub-sequence of text and a corresponding negative clique by replacing the middle sentence with a random sentence from elsewhere in the text. Then a concatenation of constituent sentence representations from an LSTM is fed into a *sigmoid* layer to obtain the probability of being a coherent clique. To capture local coherence, Mim et al. (2019) adopt the most commonly used idea in sentence reordering detection to generate incoherent documents by randomly shuffling sentences or discourse indicators (such as *however* and *in my opinion*) in a document. Then a logical sequence-aware model is trained over both the coherent and incoherent

documents, which is later used as an encoder to obtain essay representations to perform essay scoring. Elsewhere, Ji and Smith (2017) use a recursive neural model built on a discourse dependency tree which is transformed from the rhetorical structure theory tree (RST: Mann and Thompson (1988)), to focus on important content from the perspective of RST, where the RST tree is obtained by applying a discourse parser to the text.

Models have also been proposed to capture coherence by computing the correlation between two hidden representations (Tay et al., 2018, Mesgar and Strube, 2018). For example, Tay et al. (2018) propose to capture local coherence by computing the similarity of two LSTM outputs, which is denoted as a neural coherence feature. These neural coherence features are in turn concatenated with essay representation obtained via average-pooling over LSTM outputs to assign a score for a given essay. Mesgar and Strube (2018) propose a model capturing not only local coherence but also change patterns of semantic information in the document. Specifically, the most similar hidden states from an LSTM in two consecutive sentences are used to capture the salient semantic concepts that connect the sentences, denoted as local coherence, and the similarity scores between adjacent semantic concepts are used to represent patterns of semantic information changes as the document develops. Li et al. (2018) use multi-head self-attention to capture long distance relationships between words, which are passed to an LSTM layer to estimate coherence scores of essays.

Approaches have also been proposed to detect disfluency in spontaneous speech transcripts. For example, Lou and Johnson (2017) use a Bi-LSTM based language model to score candidate disfluency analyses generated by a noisy channel, where scores from the language model are used as features to re-rank these candidate disfluency analyses. Later, Lou et al. (2018) propose to augment the convolutional neural network with an auto-correlational kernel, which can detect the rough copy dependencies between disfluent words, thus improving performance.

To detect languages in code-switched text, neural models have been explored variously (Mave et al., 2018, Samih et al., 2016, Kocmi and Bojar, 2017, Adouane et al., 2018a,b, Samih et al., 2016). For example, Samih et al. (2016) and Mave et al. (2018) use a Bi-LSTM to

obtain representations at the word and character levels, which are concatenated together to identify languages in code-switched text. Kocmi and Bojar (2017) and Adouane et al. (2018b) use a bidirectional gated recurrent unit to predict the language at a character level. To alleviate the problem of lacking enough annotated code-switched data, Adouane et al. (2018a) and Adouane et al. (2018b) inject background knowledge, in the form of lexicons, pre-trained sub-word embeddings, or hidden representations from a language model trained on an unlabelled corpus, into RNN and CNN representations. Furthermore, Samih et al. (2016) combine automatically-learned features from a neural network with hand-crafted features, which are fed into a CRF to predict languages in code-switched text.

2.7 Summary

In this chapter, we have thoroughly reviewed document quality assessment across four domains (quality assessment of Wikipedia articles, academic paper rating, essay scoring, and content quality assessment in cQA), in addition to document coherence measurement. For documents from different domains, we discussed document quality assessment from the following aspects: task definition, task modelling, evaluation metrics, hand-crafted features, and end-to-end models, and included a general discussion for each aspect.

Next, we address the four research questions outlined in Section 1.2, focusing on complementing textual features with visual features (Chapter 3), enhancing neural models with hand-crafted features (Chapter 4), modelling the uncertainty of document quality assessment systems (Chapter 5), and proposing a novel task to evaluate document coherence modelling, along with a large-scale dataset (Chapter 6).

Chapter 3

Multimodal Document Quality Assessment

In this chapter, we address the first research question – whether a neural model based on visual renderings of documents is effective in predicting the quality of documents and whether the performance of a quality assessment system can be improved by using both the textual and visual representations. To answer this question, we propose to use Inception V3 model (INCEPTION, Szegedy et al. (2016)), a widely used visual model in computer vision. INCEPTION achieves superior performance than models learned from textual data alone over a newly constructed Wikipedia dataset, and competitive results with models learned from textual data alone over the academic paper reviewing dataset (Kang et al., 2018). We further propose a joint model to predict document quality by combining visual and textual features. This model yields improvement over both the Wikipedia and academic paper reviewing datasets, indicating that visual and textual features complement each other, and the general applicability of our proposed method. We also investigate what kinds of features a neural model learns by modelling document quality assessment and feature learning as a multi-task learning problem, providing interpretability of both the textual and visual models. Lacking visual renderings in existing document quality assessment datasets, we construct a large-scale Wikipedia dataset

consisting of textual data, visual renderings, and quality class labels, and supplement an existing academic paper reviewing dataset with corresponding visual representations.

We first present the motivation for utilising the INCEPTION model and proposing the JOINT model to predict the quality of documents in Section 3.1, followed by a brief summary of related work in Section 3.2. Next, we detail our proposed method: the JOINT model consisting of the visual model INCEPTION and textual model BI-LSTM in Section 3.3. We describe experimental results in Section 3.4 and model interpretability in Section 3.5. We finish this chapter by discussing applicability of our proposed method to other domains in Section 3.6 and presenting a summary of this chapter in Section 3.7.

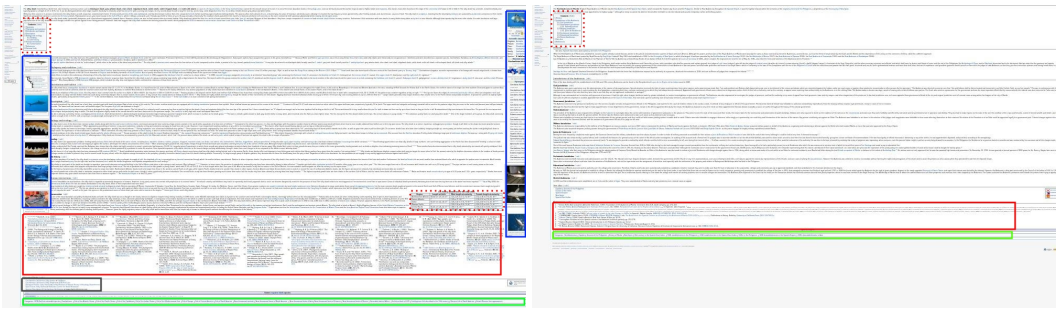
This chapter is based on and extends the following papers:

> Aili Shen, Bahar Salehi, Timothy Baldwin and Jianzhong Qi. 2019. A Joint Model for Multimodal Document Quality Assessment. In *Proceedings of the 18th Joint Conference on Digital Libraries (JCDL 2019)*, pages 107–110, Urbana-Champaign, USA.
>
> Aili Shen, Bahar Salehi, Jianzhong Qi and Timothy Baldwin. 2020. A General Approach to Multimodal Document Quality Assessment. *Journal of Artificial Intelligence Research (JAIR)* (accepted in June 2020).

3.1 Motivation

The task of document quality assessment is to automatically assess a document according to some predefined inventory of quality labels. This can take many forms, including essay scoring (quality = language quality, coherence, and relevance to a topic), job application filtering (quality = suitability for role + visual/presentational quality of the application),[1] or answer selection in community question answering (quality = actionability + relevance of the answer to the question). In this work, we focus on document quality assessment in

[1]There is, of course, the possibility that visual models can capture various superfluous indicators in this context particularly, given the presentational diversity of job application materials.



(a) Featured article (b) Lower quality article

Figure 3.1 Visual renderings of two example Wikipedia documents with different quality labels (not intended to be readable).

two contexts: Wikipedia document quality classification, and predicting whether a paper submitted to a conference will be accepted.

Automatic quality assessment has obvious benefits in terms of time savings and tractability in contexts where the volume of documents is large. In the case of dynamic documents (possibly with multiple authors), such as in the case of Wikipedia, it is particularly pertinent, as any edit potentially has implications for the quality label of that document (around 10 English Wikipedia documents are edited per second[2]). Furthermore, when the quality assessment task is decentralised (as in the case of Wikipedia and academic paper assessment), quality criteria are often applied inconsistently by different people, where an automatic document quality assessment system could potentially reduce inconsistencies and enable immediate author feedback.

Current studies on document quality assessment mainly focus on textual features. For example, Warncke-Wang et al. (2015) examine features such as the article length and the number of headings to predict the quality class of a Wikipedia article. In contrast to these studies, we propose to examine the effectiveness of using visual features, based on a visual rendering of the document. Figure 3.1 illustrates our intuition, relative to Wikipedia articles. Without reading the text, we can tell that the article in Figure 3.1a is most likely to have a higher quality than that in Figure 3.1b, as it has a variety of images (in blue box), extensive references (in red box), and a list of categories (in green box). Based on

[2] `https://en.wikipedia.org/wiki/Wikipedia:Statistics`

this intuition, we aim to answer the following questions: *Can the quality of documents be captured by visual renderings of documents? If so, can we achieve better accuracy at document quality assessment by complementing textual features with visual features?*

To answer the aforementioned questions, we propose to fine-tune an INCEPTION (Szegedy et al., 2016) model over visual renderings of documents, aiming to examine the utility of visual renderings. We further propose to combine textual features with visual features, based on a visual rendering of the document. Our visual model is based on fine-tuning an Inception V3 model (Szegedy et al., 2016) over visual renderings of documents, while our textual model is based on a hierarchical BI-LSTM. We combine the two into a joint model. To test the effectiveness of our proposed joint model, we perform experiments on five datasets: a novel Wikipedia dataset constructed for this work, three arXiv subsets split based on subject category (Kang et al., 2018), and a dataset consisting of academic papers in computer vision (Huang, 2018). Experimental results on the visual renderings of documents show that visual features, such as images and visual layout, can be captured by an image classifier, at a level comparable to a text classifier. When we combine the two models, we achieve state-of-the-art results over four out of five of the datasets.

Modern-day neural network models are highly complex with millions or billions of parameters, hindering the interpretability as to what kinds of features they have implicitly learned. This raises the following question: *what kinds of features does a textual model/visual model learn in predicting document quality?* To answer this question, we train our model to jointly predict document quality and a rich set of hand-crafted features, via multi-task learning. In doing so, we are able to analyse the behaviour of neural networks in terms of their ability to reproduce features that have been shown to have utility in document quality assessment. Experimental results show that both the visual and textual model are generally more adept at learning structural features than readability scores. Analysing the results in detail, we find that the visual model is better at capturing structural features while the textual model is better at learning text readability scores.

3.2 Related Work

Here, we briefly describe related work in quality assessment of Wikipedia articles and academic paper rating. In this work we focus exclusively on the content of the document itself. Metadata related to the document, such as the revision history and editor–article network in Wikipedia, and co-authorship network in academic paper rating, are left for future exploration.

3.2.1 Wikipedia Article Quality Assessment

Quality assessment of Wikipedia articles is the task of assigning a quality class label, see Table 2.1 in Section 2.1.1, to a given Wikipedia article, mirroring the quality assessment process that the Wikipedia community carries out manually. Various approaches have been proposed that use features from the article itself, meta-data features (e.g., the editors, and Wikipedia article revision history), or a combination of the two (see Section 2.4.1). Article-internal features capture information such as whether an article is properly organised, with supporting evidence, and with appropriate terminology. For example, Lipka and Stein (2010) use writing styles represented by binarized character tri-gram features to identify featured articles. Warncke-Wang et al. (2013) and Warncke-Wang et al. (2015) explore the number of headings, images, and references in the article. Dang and Ignat (2016a) further explore nine readability scores, such as the percentage of difficult words in the document, to measure the quality of the article, and observe performance improvements. Meta-data features, which are indirect indicators of article quality, are usually extracted from revision history, and the interaction between editors and articles. For example, one heuristic that has been proposed is that higher-quality articles have more edits (Dalip et al., 2014, 2017). Wang and Iwaihara (2011) use the percentage of registered editors and the total number of editors of an article. Article–editor dependencies have also been explored. For example, Stein and Hess (2007) use the authority of editors to measure the quality of Wikipedia articles, where the authority of editors is determined by the quality of articles they edit.

Deep learning approaches to predicting Wikipedia article quality have also been proposed (see Section 2.5.1). For example, Dang and Ignat (2016c) use a version of DOC2VEC (Le and Mikolov, 2014) to represent articles, and feed the document embeddings into a four hidden layer neural network. Dang and Ignat (2017) exploit two stacked BI-LSTMs to learn document representations. Wang and Li (2020) compare the performance of different neural models (such as CNNs and LSTMs), in differentiating high-quality Wikipedia articles from low-quality ones.

The aforementioned approaches either do not take visual features into account or use statistics (the number of) of visual features, without paying attention to the detailed visual content and other unobserved features captured in the visual renderings but not in the textual content. Also they do not perform any analysis of the learned representations, making it difficult to understand what the models have learned. In our work, we not only propose to use a neural model widely adopted in computer vision to learn visual features but also further propose a joint model consisting of textual and visual models to predict the quality of Wikipedia articles, along with an analysis of neural models' behaviour in reproducing structural features and readability scores.

3.2.2 Academic Paper Rating

Academic paper rating is a relatively new task in NLP/AI, with the basic formulation being to automatically predict whether a paper is accepted/rejected by a conference, based on the naive assumption that any paper published at a workshop was/would have been rejected by a conference. Kang et al. (2018) explore hand-crafted features, such as the length of the title, whether specific words (such as *outperform*, *state-of-the-art*, and *novel*) appear in the abstract, and an embedded representation of the abstract as input to different downstream learners, such as logistic regression, decision tree, and random forest. Yang et al. (2018) exploit a modularised hierarchical CNN, where each paper section is treated as a module. For each paper section, they train an attention-based CNN, and an attentive pooling layer is applied to the concatenated representation of each section, which is then fed into a softmax layer. Louis and Nenkova (2012) use syntactic patterns to differentiate

conference papers from workshop papers, where syntactic features are extracted from abstracts, introductions, and related work. Elsewhere, Huang (2018) uses ResNet-18 (He et al., 2016) to learn visual features only (similar to our visual baseline, which we introduce later), to predict whether a paper in computer vision area is a conference or workshop paper.

Previous studies in academic paper rating focus on either textual features alone (see Section 2.4.2 and Section 2.5.2)or visual features alone (see Huang (2018)), without exploring their interactions in rating academic papers. Furthermore, what kinds of hand-crafted features a model can learn is not studied, resulting in models that lack interpretability. In our work, we propose a joint model to make use of both textual and visual features in rating academic papers, along with interpreting models' behaviour in capturing hand-crafted features in a multi-task setting. The primary focus of our research, however is the Wikipedia setting.

3.2.3 Multimodal Document Processing

There have been studies exploring what kinds of features a model can learn when complementing textual information with visual information, but not in the context of document quality assessment. For example, Bruni et al. (2014) complement the distributional representations of words with distributional representations of visual words (visual units obtained by discretizing the image content) extracted from a multimodal document containing both text and images. Their proposed multimodal method consists of two sub-models: a text model and a visual model. Each sub-model obtains semantic representations derived from treating words/visual words as a bag-of-words, ignoring order.

In contrast, in addition to predicting the quality of documents, our work explores not only what kinds of features each sub-model can learn but also their correlation with hand-crafted features. Further, our textual sub-model takes word order into account.

3.3 The Proposed JOINT Model

We treat document quality assessment as a classification problem, i.e., given a document, we predict its quality class (e.g., whether an academic paper should be accepted or rejected). The proposed model is a joint model that integrates visual features learned through a visual model, with textual features learned through a textual model. In this section, we present the details of the visual and textual embeddings, and describe how we combine the two. We return to discussing hyper-parameter settings and the experimental configuration in the next section.

3.3.1 Visual Embedding Learning

A wide range of models have been proposed for image classification, such as VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2016), Inception V3 (Szegedy et al., 2016), and Xception (Chollet, 2017). The only work we are aware of that has used visual renderings of documents to assess document quality is the very recent arXiv paper of Huang (2018), which uses visual features only (similar to our INCEPTION baseline in the Wikipedia section), to predict whether a paper is a conference or workshop paper. In our work, we use Inception V3 pre-trained on ImageNet[3] ("INCEPTION" hereafter) to obtain visual embeddings of documents, noting that any image classifier could be applied to our task. The input to INCEPTION is a visual rendering (screenshot) of a document, and the output is a visual embedding, which we will later integrate with our textual embedding.

As it is difficult to decide what types of convolution to apply to each layer (such as 3×3 or 5×5), the basic Inception model applies multiple convolution filters in parallel and concatenates the resulting features, which are fed into the next layer. This has the benefit of capturing both local features through smaller convolutions and abstracted features through larger convolutions. INCEPTION is a hybrid of multiple Inception models of different architectures. To reduce computational cost, INCEPTION also modifies the basic

[3] http://www.image-net.org/

model by applying a 1×1 convolution to the input and factorising larger convolutions into smaller ones.

3.3.2 Textual Embedding Learning

We adopt a bidirectional LSTM model to generate textual embeddings for document quality assessment ("BI-LSTM" hereafter). The input to BI-LSTM is a textual document, and the output is a textual embedding, which we will later integrate with the visual embedding. For BI-LSTM, each word is represented as a word embedding (Bengio et al., 2003), and an average-pooling layer is applied to the word embeddings to obtain the sentence embedding, which is fed into a bi-directional LSTM. Then an average-pooling layer is applied to obtain document representations.

We detail our model as illustrated in Figure 3.2. First, an average-pooling layer is applied to word embeddings within a sentence to obtain a representation of each sentence. Each word is represented as a word embedding (Bengio et al., 2003), which is a continuous, real-valued vector.

Second, we use a bidirectional LSTM to obtain a representation of each article. Suppose that an article contains n sentences: $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$, the bidirectional LSTM contains a forward LSTM which reads the article from sentence $\boldsymbol{s}_1$ to $\boldsymbol{s}_n$, and a backward LSTM which reads the article from sentence $\boldsymbol{s}_n$ to $\boldsymbol{s}_1$. Given a sentence $\boldsymbol{s}_j$, we can obtain its hidden state $\boldsymbol{h}_j$ by concatenating the forward output $\overrightarrow{\boldsymbol{h}}_j$ and the backward output $\overleftarrow{\boldsymbol{h}}_j$, i.e., $\boldsymbol{h}_j = [\overrightarrow{\boldsymbol{h}}_j, \overleftarrow{\boldsymbol{h}}_j]$.

Finally, a max-pooling layer is applied to select the most salient features among sentences, which are used as the textual embeddings.

3.3.3 The Joint Model

The proposed joint model ("JOINT" hereafter) combines the visual and textual embeddings (output of INCEPTION and BI-LSTM) via a feed-forward layer and softmax over the

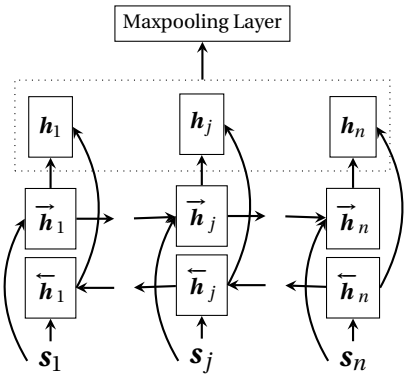


Figure 3.2 Illustration of the textual model (BI-LSTM), a component of our JOINT model.

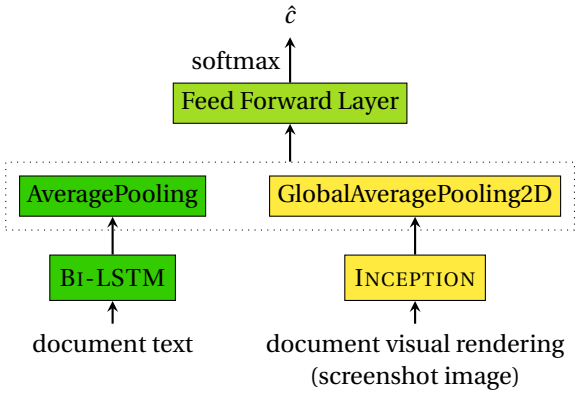


Figure 3.3 Illustration of our proposed model (JOINT), which consists of textual and visual models.

document label set, as shown in Figure 3.3. We optimise our model based on cross-entropy loss.

3.4 Experiments

In this section, we first describe our experimental settings. Then we report experimental results over five datasets: (1) Wikipedia, (2) three arXiv subsets, and (3) CVPG. In each case, we first describe the dataset, then present results, including multiple baseline approaches in each case.

3.4.1 Experimental Setting

As discussed above, our model has two main components — BI-LSTM[4] and INCEPTION— which generate textual and visual representations, respectively. For the BI-LSTM component, each article is divided into sentences and tokenized using NLTK (Bird, 2006). Words appearing more than 20 times are retained when building the vocabulary. All other words are replaced by the special `UNK` token. We use the pre-trained GloVe (Pennington et al., 2014) 50-dimensional word embeddings to represent words. For words not in GloVe, word embeddings are randomly initialised based on sampling from a uniform distribution $U(-1,1)$. All word embeddings are updated in the training process. We set the LSTM hidden layer size to 256. The concatenation of the forward and backward LSTMs thus gives us 512 dimensions for the document embedding. A dropout layer is applied at both the sentence and document level, respectively, with a probability of 0.5.

For INCEPTION, we adopt data augmentation techniques in the training with a "nearest" filling mode, a zoom range of 0.1, a width shift range of 0.1, and a height shift range of 0.1. As the original screenshots are of size 1,000×2,000 pixels, they are resized to 500×500 to feed into INCEPTION, where the input shape is (500, 500, 3). A dropout layer is applied with a probability of 0.5. Then, a GlobalAveragePooling2D layer is applied, which produces a 2,048 dimensional representation.

For the JOINT model, we get a representation of 2,560 dimensions by concatenating the 512 dimensional representation from BI-LSTM with the 2,048 dimensional representation from INCEPTION. The dropout layer is applied to the two components with a probability of 0.5. For BI-LSTM, we use a mini-batch size of 128 and a learning rate of 0.001. For both INCEPTION and the JOINT model, we use a mini-batch size of 16 and a learning rate of 0.0001. All hyper-parameters were set empirically over the development data, and the models are optimised using the Adam optimiser (Kingma and Ba, 2015).

In the training phase, the weights in INCEPTION are initialised by parameters pre-trained on ImageNet, and the weights in BI-LSTM are randomly initialised (except for the

[4]We adopt BI-LSTM rather than BI-LSTM$_{\mathrm{H}}$ (introduced later) in our JOINT model as BI-LSTM achieves better performance than BI-LSTM$_{\mathrm{H}}$ in general, which can be observed in our experimental results.

word embeddings). We train each model for 50 epochs. However, to prevent overfitting, we adopt early stopping, where we stop training the model if the performance on the development set does not improve for 20 epochs. For evaluation, we use (micro-)accuracy, following previous studies (Dang and Ignat, 2016a, Kang et al., 2018, Huang, 2018).

One may argue that GloVe word embeddings are pre-trained on Wikipedia 2014[5] and Gigaword 5[6], meaning that there is overlap between the pre-training data and our dataset. We note, however, that GloVe is pre-trained via an unsupervised objective, aimed at capturing word relations in the vector space instead of relations between word embeddings and quality labels. Therefore, the dataset that GloVe is pre-trained upon is less likely to provide clues as to quality categories. Furthermore, the version of Wikipedia documents we used in our experiments is different to that used to pre-train GloVe.

3.4.2 Wikipedia

3.4.2.1 Dataset

The Wikipedia dataset consists of articles from English Wikipedia, with quality class labels assigned by the Wikipedia community. Wikipedia articles are labelled with one of six quality classes, in descending order of quality: Featured Article ("FA"), Good Article ("GA"), B-class Article ("B"), C-class Article ("C"), Start Article ("Start"), and Stub Article ("Stub"). A description of the criteria associated with the different classes can be found in the Wikipedia grading scheme page.[7] The quality class of a Wikipedia article is assigned by Wikipedia reviewers or any registered user, who can discuss through the article's talk page[8] to reach a consensus. We constructed the dataset by first crawling all articles from each quality class repository, e.g., we get FA articles by crawling articles from the FA repository: `https://en.wikipedia.org/wiki/Category:Featured_articles`. This resulted in around 5K FA, 28K GA, 212K B, 533K C, 2.6M Start, and 3.2M Stub articles.

[5]The Wikipedia dump was released in 2014.
[6]`https://catalog.ldc.upenn.edu/LDC2011T07`
[7]`https://en.wikipedia.org/wiki/Template:Grading_scheme`
[8]`https://en.wikipedia.org/wiki/Help:Talk_pages`

Class	Train	Dev	Test	Total
FA	4000	500	500	5000
GA	4000	500	500	5000
B	4000	500	455	4955
C	4000	500	467	4967
Start	4000	500	451	4951
Stub	4000	500	421	4921
Total	24000	3000	2794	29794

Table 3.1 Statistics of the Wikipedia dataset used to benchmark our proposed model (JOINT).

In this work, we focus on examining the feasibility and actionality in evaluating the quality of Wikipedia articles. To simplify this task, we construct a balanced dataset, as techniques for imbalanced datasets are required to deal with a dataset with skewed class distributions, which will complicate the task. We acknowledge that a dataset that mimics the real dataset distribution is desirable and necessary, and that once results over the balanced dataset have improved further, future research should incorporate such evaluation. As the number of FA in Wikipedia is limited, we randomly sampled 5,000 articles from each quality class and removed all redirect pages, resulting in a dataset of 29,794 articles. As the wikitext contained in each document contains markup relating to the document category such as *{Featured Article}* or *{geo-stub}*, which reveals the quality label, we removed such information. We additionally randomly partitioned this dataset into training, development, and test splits based on a ratio of 8:1:1. Details of the dataset are presented in Table 3.1.

We generate a visual representation of each document via a 1,000×2,000-pixel screenshot of the article by running a PhantomJS script over the rendered version of the article,[9] ensuring that the screenshot and wikitext versions of the article are the same version. Any direct indicators of document quality (such as the FA indicator, which is a bronze star icon in the top right corner of the web page) are removed from the screenshot.

[9] `https://github.com/ariya/phantomjs/blob/master/examples/rasterize.js`

3.4.2.2 Baseline Approaches

We compare our models against the following six baselines:

- MAJORITY: the model labels all test samples with the majority class of the training data.
- BENCHMARK: a benchmark method from the literature (Dang and Ignat, 2016a), which uses structural features and readability scores as features to build a random forest classifier. Detailed description of these features can be found in the Hand-Crafted Features section.
- DOC2VEC: Doc2Vec (Le and Mikolov, 2014, Lau and Baldwin, 2016) to learn 500d document embeddings, and a 4-layer feed-forward classification model on top of this, with 2000, 1000, 500, and 200 dimensions, respectively.
- $\text{INCEPTION}_{\text{FIXED}}$: the frozen INCEPTION model, where only parameters in the last layer are fine-tuned during training.
- BI-LSTM: first derive a sentence representation by averaging across words in a sentence, then feed the sentence representation into a biLSTM and an average-pooling layer over the output sequence to learn a 512d document level representation, which is used to predict document quality.
- $\text{BI-LSTM}_{\text{H}}$: the hierarchical attention model of Yang et al. (2016), which incorporates hierarchical structure and an attention mechanism into a GRU (Cho et al., 2014), and achieves improved performance over six document classification datasets, such as Yelp (Tang et al., 2015) and Amazon (Zhang et al., 2015). Thus we also compare our proposed models with a BI-LSTM, with a hierarchical structure and attention mechanism ("$\text{BI-LSTM}_{\text{H}}$" hereafter). $\text{BI-LSTM}_{\text{H}}$ first generates a sentence embedding by applying a biLSTM to words in a sentence, then an attention mechanism to outputs of the BI-LSTM to weight words based on their importance in the sentence; then the sequence of sentences is fed into another BI-LSTM, where another attention layer is used to weight outputs based on the importance of sentences

in the document. Here, document tokenization, word embedding initialisation and updating are the same as in BI-LSTM. We set the BI-LSTM hidden layer size for both the sentence and document levels to 50, which gives us a 100-dimensional sentence and document embedding. The context vectors for both words and sentences also have a dimensionality of 100. A dropout layer is applied at both the sentence and document levels, with a probability of 0.5. All other hyper-parameters are set as in BI-LSTM except that a mini-batch size of 64 is used in BI-LSTM$_{\mathrm{H}}$.

The hyper-parameters of BENCHMARK, DOC2VEC, and BI-LSTM are based on the corresponding papers, except that we fine-tune the feed forward layer of DOC2VEC on the development set and train the model for 300 epochs.

3.4.2.3 Experimental Results

Table 3.2 shows the performance of the different models, in the form of the average accuracy on the test set (along with the standard deviation) over 10 runs, with different random initializations.

On Wikipedia, we observe that the performance of BI-LSTM$_{\mathrm{H}}$, BI-LSTM, INCEPTION, and JOINT is much better than that of all four baselines. INCEPTION achieves 2.9% higher accuracy than BI-LSTM and 2.2% higher accuracy than BI-LSTM$_{\mathrm{H}}$. The performance of JOINT achieves an accuracy of 59.4%, which is at least 4.6% higher than using textual features alone (BI-LSTM and BI-LSTM$_{\mathrm{H}}$) and 2.4% higher than using visual features alone (INCEPTION). Based on a two-tailed Wilcoxon signed-rank test, the difference between JOINT and the single-modality models is statistically significant ($p < 0.05$). This shows that the textual and visual features complement each other, achieving state-of-the-art results in combination.

		MAJORITY	BENCHMARK	DOC2VEC	INCEPTION$_{\text{FIXED}}$	BI-LSTM$_{\text{H}}$	BI-LSTM	INCEPTION	JOINT
Wikipedia		16.7%	46.7±0.34%	23.2±1.41%	43.7±0.51%	54.8±0.68%	54.1±0.47%	57.0±0.63%	**59.4±0.47%**†
arXiv	cs.ai	92.2%	92.6%	73.3±9.81%	92.3±0.29%	92.2±1.28%	91.5±1.03%	92.8±0.79%	**93.4±1.07%**†
	cs.cl	68.9%	75.7%	66.2±8.38%	75.0±1.95%	73.3±3.41%	76.2±1.30%	76.2±2.92%	**77.1±3.10%**
	cs.lg	67.9%	70.7%	64.7±9.08%	73.9±1.23%	76.8±2.11%	**81.1±0.83%**	79.3±2.94%	79.9±2.54%
CVPG		79.33%	N/A	72.3±10.04%	82.9±0.40%	87.7±0.39%	88.2±0.49%	86.9±0.87%	**88.8±0.90%**

Table 3.2 Performance comparisons between our proposed model (JOINT) and baselines. The best result for each dataset is indicated in **bold**, and marked with "†" if it is significantly better than the second best result (based on a two-tailed Wilcoxon signed-rank test; $p < 0.05$). The results of BENCHMARK on the arXiv dataset are from the original paper, where the standard deviation values were not reported. All neural models except for INCEPTION$_{\text{FIXED}}$ have larger standard deviation values on arXiv than Wikipedia and CVPG, which can be explained by the small size of the arXiv test set.

Subject	Accepted	Train	Dev	Test	Total
cs.ai	10%	3682	205	205	4092
cs.cl	30%	2374	132	132	2638
cs.lg	32%	4543	252	253	5048

Table 3.3 Statistics of the arXiv dataset. "Accepted" indicates the proportion of accepted papers in the given subject.

3.4.3 arXiv

3.4.3.1 Dataset

The arXiv dataset (Kang et al., 2018) consists of three subsets of academic papers (submitted in the years of 2007 – 2017) under the arXiv repository of Computer Science (cs), from the three subject areas of: Artificial Intelligence (cs.ai), Computation and Language (cs.cl), and Machine Learning (cs.lg). In line with the original dataset formulation (Kang et al., 2018), a paper is considered to have been accepted (i.e. is positively labeled) if it satisfies one of the following criteria: (1) the submission matches one of the bibliographic entries in the DBLP directory; and (2) the submission title matches an accepted paper in one of ACL, EMNLP,NAACL, EACL, TACL, NeurIPS, ICML, ICLR, and AAAI conferences. Failing this, it is considered to be rejected (noting that, in practice, some of the papers may not have been submitted to any of these conferences). The median numbers of pages for papers in cs.ai, cs.cl, and cs.lg are 11, 10, and 12, respectively. To make sure each page in the PDF file has the same size in the screenshot, we crop the PDF file of a paper to the first 12; we pad the PDF file with blank pages if a PDF file has less than 12 pages, using the PyPDF2 Python package.[10] We then use ImageMagick[11] to convert the 12-page PDF file to a single 1,000×2,000 pixel screenshot. Table 3.3 details this dataset, where the "Accepted" column denotes the percentage of positive instances (accepted papers) in each subset.

[10] `https://pypi.org/project/PyPDF2/`
[11] `https://www.imagemagick.org/script/index.php`

3.4.3.2 Baseline Approaches

We compare our models against the six baselines mentioned above except that: (1) BENCHMARK is the method of Kang et al. (2018), who use hand-crafted features, such as the number of references and TF-IDF weighted bag-of-words in the abstract, to build a classifier based on the best of logistic regression, multi-layer perception, and AdaBoost; and (2) we fine-tune the feed forward layer of DOC2VEC on the development set and train the model for 50 epochs.

3.4.3.3 Experimental Results

Once again, the JOINT model achieves the highest accuracy on cs.ai and cs.cl by combining textual and visual representations (at a level of statistical significance for cs.ai). This, again, confirms that textual and visual features complement each other, and together they achieve state-of-the-art results. On cs.lg, JOINT achieves a 0.6% higher accuracy than INCEPTION by combining visual features and textual features, but BI-LSTM achieves the highest accuracy. One characteristic of cs.lg documents is that they tend to contain more equations than the other two arXiv datasets, and preliminary analysis suggests that the BI-LSTM is picking up on a correlation between the volume/style of mathematical presentation and the quality of the document. Surprisingly, $\text{INCEPTION}_{\text{FIXED}}$ is better than MAJORITY and BENCHMARK over the arXiv cs.lg subset, which verifies the usefulness of visual features, even when only the last layer is fine-tuned. Table 3.2 also shows that INCEPTION and BI-LSTM achieve similar performance on arXiv, showing that textual and visual representations are equally discriminative: INCEPTION and BI-LSTM are indistinguishable over cs.cl; BI-LSTM achieves 1.8% higher accuracy over cs.lg, while INCEPTION achieves 1.3% higher accuracy over cs.ai. Comparisons between $\text{BI-LSTM}_{\text{H}}$ and INCEPTION show the superiority of INCEPTION: INCEPTION achieves 0.6%, 2.9%, and 2.5% higher accuracy over cs.ai, cs.cl, and cs.lg, respectively.

For cs.ai, MAJORITY, BENCHMARK, and $\text{INCEPTION}_{\text{FIXED}}$ outperform or are competitive with BI-LSTM and $\text{BI-LSTM}_{\text{H}}$, in large part because of the class imbalance in this dataset (90% of papers are rejected).

Class	Train	Dev	Test	Total
Accepted	3856	783	978	5617
Rejected	1005	251	247	1503
Total	4861	1034	1225	7120

Table 3.4 Statistics of the CVPG dataset, in terms of the number of documents that were accepted or rejected by a conference.

3.4.4 CVPG

3.4.4.1 Dataset

The CVPG dataset (Huang, 2018) consists of conference papers ("accepted" papers) and workshop papers ("rejected" papers) from top-tier computer vision conferences: six CVPR and three ICCV proceedings from 2013 to 2018. In line with Huang (2018), a paper is considered to be accepted if it is from the main conference, and otherwise is considered to have been rejected (from the main conference).[12] Following Huang (2018), we crop/pad the PDF file of a paper to 8 pages if a paper has more/less than 8 pages.[13] Then we use ImageMagick to convert the 8-page pdf file to a single 1,000×2,000 pixel screenshot. Main conference papers and workshop papers from CVPR 2017 are used as the validation set; main conference papers and workshop papers from CVPR 2018 are the test set; the remaining papers are the training set. Full statistics of the dataset are shown in Table 3.4.

3.4.4.2 Baseline Approaches

Over CVPG, we compare our models against the same baselines used in Wikipedia. The BENCHMARK baseline is not provided for CVPG, as it was not possible for us to extract hand-crafted features from the parsed PDF files with high precision: we first extracted the textual content from each paper using the Science Parse library,[14] then extracted

[12] The CVPG dataset is constructed from the CVPR and ICCV conferences rather than arXiv, not mimicking the real word scenario. It is the dataset construction process conducted by Huang (2018) that results in such a distribution.

[13] Workshop papers with less than 7 pages are removed, as they make the classification task trivial for the visual model.

[14] `https://github.com/allenai/science-parse`

hand-crafted features from the parsed content. However, the Science Parse library makes mistakes, resulting in missing sections — such as the abstract — in the extracted content, making it difficult for us to extract hand-crafted features.[15] The hyper-parameters of the baselines are the same as for Wikipedia except that we train DOC2VEC for 50 epochs.

3.4.4.3 Experimental Results

In the case of the CVPG dataset, BI-LSTM$_{\text{H}}$, BI-LSTM, INCEPTION, and JOINT achieve substantial improvements over MAJORITY and DOC2VEC. The usefulness of visual features learned by INCEPTION$_{\text{FIXED}}$ (fine-tuning only the last layer of the network) is once again verified over this dataset: INCEPTION$_{\text{FIXED}}$ obtains superior performance over MAJORITY and DOC2VEC. The JOINT model achieves the best performance by combining textual features and visual features, which demonstrates that textual features and visual features complement each other. Similar to that over cs.lg, BI-LSTM outperforms INCEPTION over the CVPG dataset. Preliminary analysis over the CVPG dataset indicates that mathematical presentations are quite common in CVPG documents, despite being less prevalent than that in cs.lg and with greater diversity, thereby BI-LSTM achieving better performance by picking up the correlation between the mathematical presentation and the quality of a document. Furthermore, images in CVPG documents are quite prevalent, contributing to the superior performance of JOINT by combining the textual and visual representations.

3.4.5 Summary of Results across Datasets

The Model performance across the different datasets is not directly comparable, as these datasets are intrinsically different: (1) Wikipedia is a six-class classification task, while arXiv and CVPG are binary; and (2) the proportion of positive (accepted) instances in cs.ai, cs.cl, cs.lg, and CVPG differ considerably (10%, 30%, 32%, and 80%, respectively). Having said this, we can compare the performance relative to the respective baselines and within task. As shown in Table 3.2, INCEPTION and BI-LSTM are competitive with each

[15] For arXiv, it contains additional review files including the abstract, making it possible for us to extract hand-crafted features mainly from the abstract.

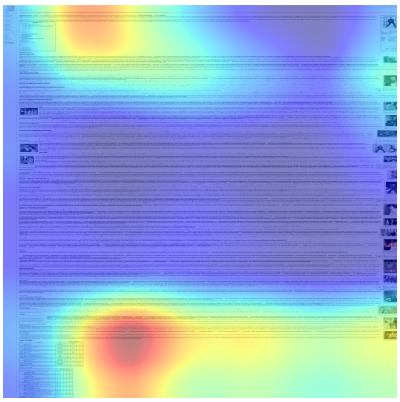
other across all these datasets: INCEPTION and BI-LSTM achieve the same performance over cs.cl; BI-LSTM is superior to INCEPTION over cs.lg and CVPG, while INCEPTION outperforms BI-LSTM over Wikipedia and cs.ai. The JOINT model achieves state-of-the-art performance over all the datasets except for cs.lg. This affirms the complementarity of visual and textual features. The superior performance of the JOINT model over the different datasets also attests to the general applicability of our model.

3.4.6 Analysis

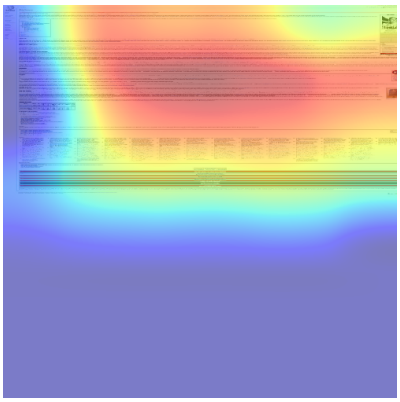
In this section, we first analyse the performance of INCEPTION and JOINT. We also analyse the performance of different models on different quality classes. Additionally, we compare the high-level representations learned by the different models through visualisation. Our analysis is based on the result over the Wikipedia dataset, and we leave the analysis over the arXiv and CVPG datasets to future work.

3.4.6.1 INCEPTION

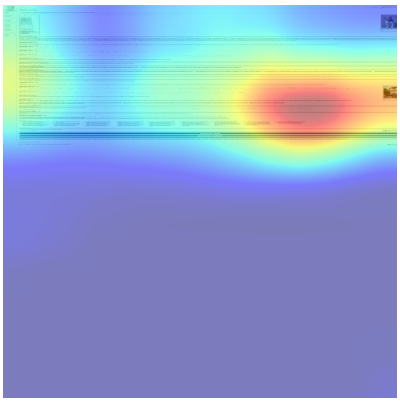
To better understand the performance of INCEPTION, we generated the gradient-based class activation map (Selvaraju et al., 2017), by maximising the outputs of each class in the penultimate layer, as shown in Figure 3.4. From Figure 3.4a and Figure 3.4b, we can see that INCEPTION identifies the two most important regions (one at the top corresponding to the table of contents, and the other at the bottom, capturing both document length and tables) that contribute to the FA class prediction, and a region in the upper half of the image that contributes to the GA class prediction (capturing images and the length of the article body). From Figure 3.4c and Figure 3.4d, we can see that the most important regions in terms of B and C class prediction capture images (down the left and right of the page, in the case of B and C), and document length and references. From Figure 3.4e and Figure 3.4f, we can see that INCEPTION finds that images in the top right corner are the strongest predictor of Start class prediction, and (the lack of) images/the link bar down the left side of the document are the most important for Stub class prediction.



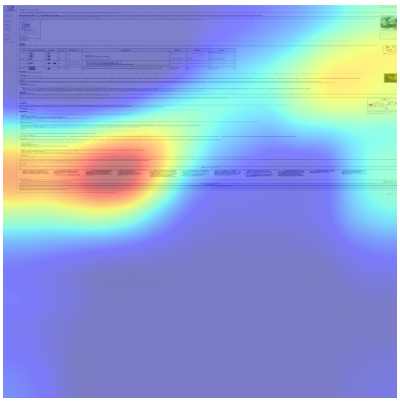
(a) FA



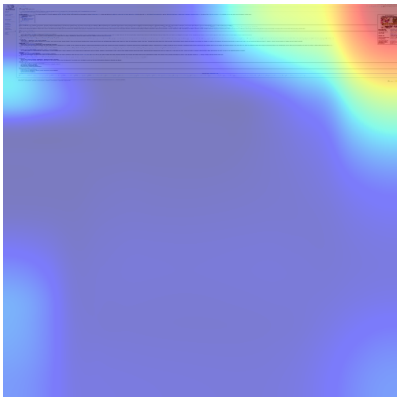
(b) GA



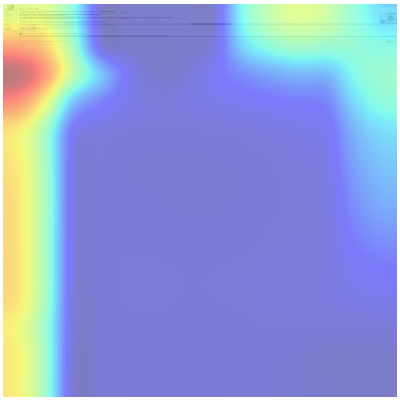
(c) B



(d) C



(e) Start



(f) Stub

Figure 3.4 Heatmap overlapped onto screenshots of each Wikipedia quality class. Best viewed in color.

Quality	FA	GA	B	C	Start	Stub
FA	397	83	20	0	0	0
GA	112	299	65	22	2	0
B	23	53	253	75	44	7
C	5	33	193	124	100	12
Start	1	6	36	85	239	84
Stub	0	0	6	7	63	345

Table 3.5 Confusion matrix of the JOINT model on Wikipedia. Rows are the actual quality classes and columns are the predicted quality classes. The diagonal (gray cells) indicates correct predictions.

3.4.6.2 JOINT

Table 3.5 shows the confusion matrix of JOINT on Wikipedia. We can see that more than 50% of documents for each quality class are correctly classified, except for the C class where more documents are misclassified into B. Analysis shows that when misclassified, documents are usually misclassified into adjacent quality classes, which can be explained by the Wikipedia grading scheme, where the criteria for adjacent quality classes are more similar.[16]

We also provide a breakdown of precision ("$\mathscr{P}$"), recall ("$\mathscr{R}$"), and F1 score ("$\mathscr{F}_{\beta=1}$") for BI-LSTM, INCEPTION, and JOINT across the quality classes in Table 3.6. We can see that JOINT achieves the highest accuracy in 11 out of 18 cases. It is also worth noting that all models achieve higher scores for FA, GA, and Stub articles than B, C and Start articles. This can be explained in part by the fact that FA and GA articles must pass an official review based on structured criteria, and in part by the fact that Stub articles are usually very short, which is discriminative for INCEPTION, and JOINT. All models perform worst on the B and C quality classes. It is difficult to differentiate B articles from C articles even for Wikipedia contributors. As evidence of this, when we crawled a new dataset including talk pages with quality class votes from Wikipedia contributors, we found that among articles with three or more quality labels, over 20% percent of B and C articles have inconsistent

[16] Suggesting that ordinal regression should boost accuracy, but preliminary experiments with various methods led to no improvement over simple classification.

Quality	Metric	BI-LSTM	INCEPTION	JOINT
FA	$\mathscr{P}$	**76.6**	74.8	73.8
	$\mathscr{R}$	72.0	68.2	**79.4**
	$\mathscr{F}_{\beta=1}$	74.2	71.3	**76.5**
GA	$\mathscr{P}$	51.3	57.7	**63.1**
	$\mathscr{R}$	**59.8**	59.0	**59.8**
	$\mathscr{F}_{\beta=1}$	55.2	58.3	**61.4**
B	$\mathscr{P}$	37.6	41.8	**44.2**
	$\mathscr{R}$	42.4	44.0	**55.6**
	$\mathscr{F}_{\beta=1}$	39.9	42.9	**49.2**
C	$\mathscr{P}$	36.3	38.9	**39.6**
	$\mathscr{R}$	27.0	**36.0**	26.6
	$\mathscr{F}_{\beta=1}$	31.0	**37.4**	31.8
Start	$\mathscr{P}$	48.2	49.4	**53.3**
	$\mathscr{R}$	44.8	**57.2**	53.0
	$\mathscr{F}_{\beta=1}$	46.4	53.0	**53.1**
Stub	$\mathscr{P}$	71.9	**83.3**	77.0
	$\mathscr{R}$	78.9	78.2	**81.9**
	$\mathscr{F}_{\beta=1}$	75.2	**80.7**	79.4

Table 3.6 Precision ("$\mathscr{P}$"), recall ("$\mathscr{R}$"), and F1 ("$\mathscr{F}_{\beta=1}$") of BI-LSTM, INCEPTION, and JOINT on Wikipedia.

votes from Wikipedia contributors, whereas for FA and GA articles the number is only 0.7%.

We further visualise the learned document representations of BI-LSTM, INCEPTION, and JOINT in the form of a t-SNE plot (van der Maaten and Hinton, 2008) in Figure 3.5. The degree of separation between Start and Stub achieved by INCEPTION is much greater than for BI-LSTM, with the separation between Start and Stub achieved by JOINT being the clearest among the three models. INCEPTION and JOINT are better than BI-LSTM at separating Start and C. JOINT achieves slightly better performance than INCEPTION in separating GA and FA. We can also see that it is difficult for all models to separate B and C, which is consistent with the findings of Tables 3.5 and 3.6.

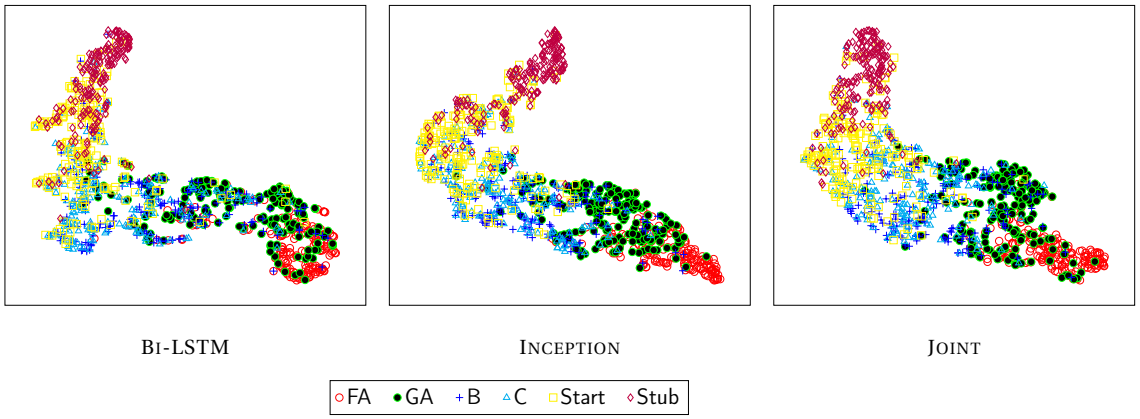


Figure 3.5 t-SNE scatter plot of Wikipedia article representations (representations from the penultimate layer of each model, based on 200 random samples from each quality class; best viewed in colour).

3.5 Model Interpretability

Although neural networks have achieved competitive or state-of-the-art performance across various tasks, such as document quality assessment and essay scoring, they have been criticised for lacking explainability. In this section, we explore what kinds of features BI-LSTM and INCEPTION implicitly learn, by allowing them to learn the quality of documents and hand-crafted feature values simultaneously. In other words, we model this as a multi-task learning problem: (1) one task is to learn the document quality; and (2) the second task is to learn the value of a given feature. We first detail 21 features used in the literature (Dang and Ignat, 2016a) to assess the quality of Wikipedia articles, then present experimental results of multi-task learning over the Wikipedia dataset. We also study the performance difference of BI-LSTM, INCEPTION, and JOINT by combining neural network-learned features with hand-crafted features.

3.5.1 Hand-crafted Features

Following Dang and Ignat (2016a), we use structural features and readability scores as hand-crafted features for quality class prediction. The structural features capture the structure of articles, and the readability scores reflect writing style. These are extracted

Structural Features	Readability Scores
Article length in bytes (*Length*)	Flesch reading score (Kincaid et al., 1975) (*Flesch*)
Number of references (*References*)	Flesch-Kincaid grade level (Kincaid et al., 1975) (*Flesch-Kincaid*)
Number of links to other Wikipedia pages (*Pagelinks*)	Smog index (McLaughlin, 1969) (*Smog*)
Number of citation templates (*Citation*)	Coleman-Liau index (Coleman and Liau, 1975) (*Coleman-Liau*)
Number of non-citation templates (*Non-citation*)	Automated readability index (Smith and Senter, 1967) (*Readability Index*)
Number of categories linked in the text (*Categories*)	Difficult words (Chall and Dale, 1995) (*Difficult Words*)
Number of images / length of the article (*Images*)	Dale-Chall score (Dale and Chall, 1948) (*Dale-Chall*)
Article having an infobox or not (*Infobox*)	Linsear write formula (Chen, 2012) (*Linsear*)
Number of level 2 headings (*Level2*)	Gunning-Fog index (Gunning, 1969) (*Gunning-Fog*)
Number of level 3+ headings (*Level3+*)	Information noise score (Zhu and Gauch, 2000) (*Infonoise*)
	Readability consensus (*Consensus*)

Table 3.7 A list of hand-crafted features explored in the multi-task setting. Here, the text in parentheses (e.g., "*Length*") is a descriptor for each feature, which will be used hereafter.

from Wikipedia articles using the open-source packages wikiclass[17] and textstat.[18] The features are listed in Table 3.7.

The structural features reflect article quality in different ways. For example, *article length* captures how much content an article contains (with the expectation that articles that do not contain much content are usually of low quality). The *number of references*, *number of links to other Wikipedia pages*, and *number of citation templates* show how the article editors support their content by using information from different sources, making the article more reliable and of higher quality. The *number of level 2 and level 3+ headings* reflects how the content is organised. Usually, Wikipedia articles of high quality have appropriate *number of level 2 and level 3+ headings*.

Readability scores reflect the use of language and how easy an article is to read. *Flesch reading score*, *Flesch-Kincaid grade level*, *Smog index*, and *Linsear write formula* use the average syllable per word or the number of polysyllables with different weight values to measure how difficult a text is to understand. Both *Coleman-Liau index* and *Automated readability index* use the average word length with different weight values to measure the readability of texts. *Readability consensus*, which is the estimated average school grade level required to understand the content, is computed by averaging across all readability scores in Table 3.7 except for *difficult words*. *Difficult words*, *Dale-Chall score*, and *Gunning-Fog index* use the number of difficult words or percentage of difficult words to measure the comprehension difficulty of a text. Here, a word is considered to be difficult if it is not in a list of 3,000 common English words that fourth-grade American students can reliably understand.

3.5.2 Experimental Settings

In the multi-task learning scenario, we train a single model for each feature, i.e., learn 21 different models, one for each feature. The settings of BI-LSTM and INCEPTION are the same as previously, except that the loss is computed differently. With the sole

[17] `https://github.com/wiki-ai/wikiclass`
[18] `https://pypi.python.org/pypi/textstat/0.5.1`

exception of *Infobox*, the features are continuous, and so are modelled as regression tasks. The combined loss is thus the (unweighted) sum of the document quality cross-entropy (between actual quality and predicted quality) and feature-level mean squared error (between the actual and predicted feature value). In each case, we standardise the feature by subtracting the mean and scaling to unit variance. In the case of *Infobox*, the target variable is binary (i.e., does the article contain an infobox or not), and so the combined loss is simply the sum of the cross-entropy for the document quality label and *Infobox* feature value.

3.5.3 Experimental Results

Table 3.8 summarises the experimental results for the feature learning task, in terms of the Pearson correlation (all features other than *Infobox*) or accuracy (*Infobox*) at predicting each feature. Predictably, INCEPTION outperforms BI-LSTM for all structural features, with the one exception of *Non-citation*, and the results achieved by INCEPTION are significantly better for 8 out of 10 structural features. Also predictably, INCEPTION achieves a much better performance at learning *Images* than BI-LSTM. We can observe that BI-LSTM is superior to INCEPTION at learning readability scores, where the results achieved by BI-LSTM are statistically better than those of INCEPTION for 8 out of 11 readability scores. Both BI-LSTM and INCEPTION achieve better performance at learning structural features than learning readability scores except for *Difficult Words* and *Infonoise*. The higher correlation in *Difficult Words* and *Infonoise* can be explained by the fact that they are highly correlated with article length, which is easy to learn for both BI-LSTM and INCEPTION. On the other hand, both BI-LSTM and INCEPTION find it difficult to learn *Linsear* and *Consensus*, as these two features are based on syllable counts or the average of multiple readability scores, which are more subtle to model.

Table 3.9 provides the results for predicting document quality in the multi-task setting. We can observe that the performance of BI-LSTM is slightly improved by explicitly learning hand-crafted features and document quality simultaneously for 14 out of 21 features, while the performance of INCEPTION benefits less from multi-task learning (only 2 features

	Length	*References*	*Pagelinks*	*Citation*	*Non-citation*	*Categories*	*Images*
BI-LSTM	0.8945	0.8542	0.8486	0.8397	**0.8256**†	0.6901	0.3278
INCEPTION	**0.9288**†	**0.8573**	**0.8861**†	**0.8764**†	0.8036	**0.8804**†	**0.8358**†
	Infobox	*Level2*	*Level3+*	*Flesch*	*Flesch-Kincaid*	*Smog*	*Coleman-Liau*
BI-LSTM	0.8530	0.7211	0.7460	**0.6209**†	**0.5488**†	**0.5368**†	0.6646
INCEPTION	**0.8954**†	**0.8769**†	**0.9311**†	0.5580	0.4602	0.3730	**0.6685**
	Readability Index	*Difficult Words*	*Dale-Chall*	*Linsear*	*Gunning-Fog*	*Infonoise*	*Consensus*
BI-LSTM	0.6163	0.9330	**0.7351**†	**0.3238**†	**0.6270**†	**0.8273**†	**0.2212**†
INCEPTION	**0.6379**†	**0.9765**†	0.6732	0.2755	0.5248	0.8199	0.1870

Table 3.8 Pearson correlation (r) for each predicted feature, for BI-LSTM and INCEPTION. The best result for each feature is indicated in bold, and marked with "†" if the difference is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$).

	Length	*References*	*Pagelinks*	*Citation*	*Non-citation*	*Categories*	*Images*
BI-LSTM	53.9%	**54.3%**	**54.4%**	**54.6%**	54.0%	54.0%	53.8%
INCEPTION	56.6%	56.9%	**57.1%**	56.8%	56.9%	56.7%	56.6%
	Infobox	*Level2*	*Level3+*	*Flesch*	*Flesch-Kincaid*	*Smog*	*Coleman-Liau*
BI-LSTM	**54.3%**	**54.6%**†	**54.2%**	**54.2%**	**54.2%**	**54.4%**	54.0%
INCEPTION	**57.2%**	56.8%	56.4%	57.0%	57.0%	56.9%	56.5%
	Readability Index	*Difficult Words*	*Dale-Chall*	*Linsear*	*Gunning-Fog*	*Infonoise*	*Consensus*
BI-LSTM	**54.2%**	54.1%	54.1%	**54.4%**	**54.2%**	**54.2%**	**54.4%**
INCEPTION	56.7%	56.9%	56.5%	56.7%	56.9%	56.5%	56.9%

Table 3.9 Performance comparisons between BI-LSTM and INCEPTION over the Wikipedia dataset in the multi-task setting. Results achieved by BI-LSTM and INCEPTION in the multi-task setting which are better than those for the corresponding single task setting are indicated in bold, and marked with "†" if the difference is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$).

help in predicting document quality). Surprisingly, some features (such as *length*) harm the performance of both BI-LSTM and INCEPTION. Examining the Pearson correlation between each feature and the document quality, we observe no relation between the correlation figures and performance over document quality assessment. For example, the correlation between *Length* and the quality label is 0.61, whereas *Length* harmed the performance of both BI-LSTM and INCEPTION; the correlation between *Smog* and the quality labels is −0.10, and yet *Smog* improved the performance of BI-LSTM and compromised the performance of INCEPTION.

3.5.4 Experiments Incorporating Hand-crafted Features

Observing that models are not able to learn hand-crafted features fully, we propose to combine network learned features with hand-crafted features, which will be detailed in Chapter 4.1. To concatenate hand-crafted features with features learned by our models, we first normalise the hand-crafted features as above for Wikipedia (by subtracting the mean and scaling to unit variance, for each feature) and by min–max normalisation for arXiv.[19] It is worth mentioning that we incorporate 21 dataset-specific features in Table 3.7 for Wikipedia, and 14 dataset-specific features for arXiv. As explained in CVPG section, extracting hand-crafted features is non-trivial for CVPG, which is the reason we don't report such results for CVPG.

Table 3.10 presents a comparison of results over Wikipedia and arXiv for our basic models, and models using hand-crafted features as side information. We can see that BI-LSTM$^+$, INCEPTION$^+$, and JOINT$^+$ achieve better results than their corresponding base models over Wikipedia and 2 out of 3 subsets of arXiv. Specifically, all models achieve statistically significant improvements over Wikipedia, and slightly better results (not at a level of significance) over 2 out of 3 of the arXiv subsets. We hypothesise that the hand-crafted features for Wikipedia are more powerful in capturing document quality than those for arXiv, in large part because of the editorial guidelines associated with Wikipedia documents, as compared to arXiv papers where writing styles and document structures vary much more.

3.6 Applicability to Other Domains

Our proposed method can also be used to score essays and assess the quality of content in cQA, while the effectiveness of our proposed method may vary.

For essay scoring, visual renderings of essays are most likely to be able to capture the paragraph structure if this information is kept in the dataset, thus may lead to small

[19] The decision to use min–max normalisation for arXiv was a purely empirical one, in that it led to better results, whereas in the Wikipedia case, z-scoring performed better.

		BI-LSTM	BI-LSTM$^+$	INCEPTION	INCEPTION$^+$	JOINT	JOINT$^+$
Wikipedia		54.1±0.47%	**57.2±0.45%**†	57.0±0.63%	**58.8±0.71%**†	59.4±0.47%	**62.5±0.51%**†
arXiv	cs.ai	91.5±1.03%	**92.1±1.06%**	92.8±0.79%	92.8±0.81%	93.4±1.07%	93.1±0.95%
	cs.cl	76.2±1.30%	**76.8±1.67%**	76.2±2.92%	**76.7±2.09%**	77.1±3.10%	**77.2±2.59%**
	cs.lg	81.1±0.83%	80.0±2.30%	79.3±2.94%	**79.5±1.28%**	79.9±2.54%	**81.2±1.05%**

Table 3.10 Performance comparisons between BI-LSTM, INCEPTION, and JOINT after incorporating hand-crafted features. BI-LSTM$^+$, INCEPTION$^+$, and JOINT$^+$ are the models that concatenate hand-crafted features with features learned by BI-LSTM, INCEPTION, and JOINT. Results achieved by BI-LSTM$^+$, INCEPTION$^+$, and JOINT$^+$ are indicated in **bold**, if they are better than those achieved by BI-LSTM, INCEPTION, and JOINT, respectively, and additionally with "†" if the difference is statistically better (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$).

improvement in essay scoring. Otherwise, our proposed method may not be able to capture useful information in predicting scores of essays. In this regard, essay scoring may not be able to benefit from our proposed method.

For content quality assessment in cQA, visual renderings of answers can provide further information, such as whether an answer contains codes and link references, whether there are comments made to the answer, and the reputation of answers, etc. Thus, the performance of quality assessment systems in cQA can be improved by incorporating visual features. One thing worth mentioning is that we have to avoid including the best answer mark and votes information in constructing visual renderings of answers.

3.7 Summary

In this chapter, we have answered the first research questions: *Can the quality of documents be captured by visual renderings of documents? If so, can we achieve better accuracy at document quality assessment by complementing textual features with visual features?* The answer is yes. We proposed to use INCEPTION to capture visual features, such as font choices, images, and visual layout, which are not captured in textual content. The effectiveness of utilising visual renderings of documents is demonstrated over Wikipedia and arXiv datasets. We presented JOINT, a model combining textual and visual representations. Experimental results over Wikipedia, arXiv, and CVPG not only underline the feasibility of

assessing document quality via visual features, and the complementarity of visual and textual document representations for quality assessment, but also demonstrate the general applicability of our proposed model. In the multi-task setting, we observe that our visual model outperforms the textual model in learning structural features, while the textual model is better at learning readability scores, further verifying the complementarity of textual and visual features. This work is based on the observation that visual renderings of documents can capture visual features, leading to a new way of capturing implicit quality indicators, thus resulting in better performance in predicting the quality of documents.

In this chapter, we mainly focus on utilising visual renderings of documents and combining them with textual representations to improve the performance of models in quality assessment of documents, while in the next chapter, we turn our attention to incorporating hand-crafted features into neural network learned representations, thereby enhancing models' performance in predicting the quality of documents based only on textual content.

Chapter 4

Enhancing Neural Models with Hand-crafted Features

Chapter 3 has focused on utilising a visual model to capture visual features not captured by a textual model and utilising learned visual and textual representations to predict document quality. In this chapter, we shift our focus to making use of hand-crafted features, in the context of neural models, where the utility of hand-crafted features alone has been demonstrated by various traditional methods (Blumenstock, 2008, Lipka and Stein, 2010, Warncke-Wang et al., 2013, 2015, Halfaker and Taraborelli, 2015, Dang and Ignat, 2016a). This chapter addresses the second research question of whether neural models can benefit from hand-crafted features and in which way, if so. Although neural models such as LSTMs and CNNs have achieved state-of-the-art or competitive results over a range of text classification/regression tasks, with the advantage of learning document representations automatically, the ability of neural models to capture the same information as carried by hand-crafted features has not been studied.

We explore two methods to make use of hand-crafted features in the context of neural models: (1) simply concatenating hand-crafted features with learned representations (Section 4.1); and (2) utilising hand-crafted features to guide model learning by explicitly attending to feature indicators in model training (Section 4.2). Both methods improve

the performance of neural models in quality assessment of Wikipedia articles, where the latter outperforms the first method.

This chapter is based on and extends the following papers:

> Aili Shen, Jianzhong Qi and Timothy Baldwin. 2017. A Hybrid Model for Quality Assessment of Wikipedia Articles. In *Proceedings of the 2017 Australasian Language Technology Workshop (ALTW 2017)*, pages 43–52, Brisbane, Australia.
>
> Aili Shen, Bahar Salehi, Jianzhong Qi and Timothy Baldwin. 2019. Feature-guided Neural Model Training for Supervised Document Representation Learning. In *Proceedings of the 2019 Australasian Language Technology Workshop (ALTW 2019)*, pages 47–51, Sydney, Australia.

4.1 Enhancing Neural Models via Concatenating Hand-crafted Features with Learned Representations

In this section, we describe our proposed hybrid method by concatenating hand-crafted features with learned representations. We first describe the motivation for the proposed model in Section 4.1.1, followed by related work in quality assessment of Wikipedia articles in Section 4.1.2. Next, we describe our proposed hybrid model in Section 4.1.3, and experimental results in Section 4.1.4. Lastly, we describe the applicability of our proposed method to other domains in Section 4.1.5.

4.1.1 Motivation

As discussed in Section 2.4.1, most previous studies rely heavily on feature engineering for quality assessment of Wikipedia articles (Blumenstock, 2008, Warncke-Wang et al., 2013, 2015, Dang and Ignat, 2016a), but more recent work using neural networks (Dang and Ignat, 2016c,b) has been explored to automatically learn document representations, which are used to predict the quality of Wikipedia articles.

Studies, such as Dang and Ignat (2016c) and Dang and Ignat (2016b), which use neural networks achieve promising results in quality assessment of Wikipedia articles, even though the performance is not at the state-of-the-art level, compared to a simple RF model using hand-crafted features. Specifically, Dang and Ignat (2016c) use the distributed memory version of *Paragraph Vector* (Le and Mikolov, 2014) to represent articles, and the learned representations of articles are fed into a four hidden layer neural network to classify articles based on their quality. Intuitively, we would expect that a model which takes the order between sentences into consideration should achieve better performance, which motivates our work.

Additionally, the performance achieved by Dang and Ignat (2016c) is less competitive compared to an RF model using hand-crafted features, indicating that neural network learned high-level representations are unable to automatically capture information carried by hand-crafted features, thus resulting in less competitive results in predicting document quality. Thus, a neural network should benefit from concatenating hand-crafted features with neural network learned high-level representations. This provides another motivation for our work.

In this section, we propose a hybrid model to assess the quality of Wikipedia articles, by concatenating hand-crafted features with BI-LSTM learned high-level representations.

4.1.2 Related Work

As discussed in Section 2.4.1, a number of approaches for quality assessment of Wikipedia articles have been proposed, which can be classified into three categories: (i) meta-data based approaches; (ii) article internal feature based approaches; and (iii) meta-data and article internal feature based approaches. We only focus on reviewing studies based on the article itself.

Article internal features refer to features derived from the articles themselves. Various such features have been used for Wikipedia article quality assessment. Blumenstock (2008) uses article length as a metric to assess the quality of Wikipedia articles. A high accuracy is achieved in separating featured articles from non-featured articles despite the

simplicity of this approach. Lipka and Stein (2010) use writing styles, which is represented by exploiting binarized character tri-gram features, to identify featured articles. Warncke-Wang et al. (2013) propose a model to assess article quality, which includes five features extracted from article text: completeness, informativeness, number of headings, and ratio of number of references to article length. Warncke-Wang et al. (2015) propose a model including 11 structural features (such as number of references) and use *Random Forest* (RF) to classify Wikipedia articles by their quality classes. Dang and Ignat (2016a) further add nine readability scores (such as Flesch reading-ease score) to the structural features and use RF to classify Wikipedia articles based on their quality. Based on these last two studies, an online *Objective Revision Evaluation Service* (ORES) has been built to measure the quality of Wikipedia articles (Halfaker and Taraborelli, 2015). ORES requires revision ID of Wikipedia articles as its input parameter. Our experimental datasets do not contain such information. Thus, we compare with the model proposed by Dang and Ignat (2016a) instead of ORES in the experiments.

Dang and Ignat (2016c) use DOC2VEC (Le and Mikolov, 2014) to represent articles, and the learned representations of articles are fed into a four hidden layer neural network to classify articles based on their quality.

However, Dang and Ignat (2016c) do not consider the order of sentences, which may affect article quality. If the sentences in an article are not ordered in a logical way, it is more difficult for reviewers and readers to understand this article and the quality of this article should be lower. Our approach differs from that of Dang and Ignat (2016c) in that we use a *long short-term memory* (LSTM) (Hochreiter and Schmidhuber, 1997), which considers the order between sentences, to learn the high-level representation of Wikipedia articles. Furthermore, although neural networks can learn features from the article content, it requires a large training dataset. Due to limited availability of labelled Wikipedia articles, we integrate the neural network learned high-level representation with manually developed features, which can lead to better quality prediction of Wikipedia articles even with a relatively small set of training data (e.g., 20,432 articles).

4.1.3 The Proposed Hybrid Model

We formulate quality assessment of Wikipedia articles as a multi-class classification problem, i.e., a model assigns a quality indicating label to a given Wikipedia article.

The proposed classification model is a hybrid model that integrates both neural network learned high-level representation and hand-crafted features. We first describe the LSTM-based model to obtain high-level representation of Wikipedia articles, followed by hand-crafted features. Next, we detail how to integrate neural network learned high-level representation of Wikipedia articles with hand-crafted features.

4.1.3.1 High-level Representation

We adopt a bidirectional LSTM model to summarise information from both directions to learn high-level representation of Wikipedia articles; we call the model "BI-LSTM". The input to BI-LSTM is the text of an article, and the output is the high-level representation of the article, which we integrate later with manually developed features. For BI-LSTM, each word is represented as a word embedding (Bengio et al., 2003), and an average-pooling layer is applied to the word embeddings to obtain the sentence embedding, which is fed into a bi-directional LSTM. Then an average-pooling layer is applied to obtain document representations (see Section 3.3.2).

4.1.3.2 Hand-crafted Features

Following Dang and Ignat (2016a), we use structural features and readability scores as the manually developed features for quality class prediction. Dang and Ignat (2016a) demonstrate the utility of readability scores in predicting the quality of Wikipedia articles, thus we adopt readability scores in this work.[1] The structural features can capture the structure information of articles, and the readability scores can reflect writing styles. We re-enumerate these features in Table 4.1, as concatenating these features with neural model

[1] Theoretically, readability scores should contribute little to predicting the quality of Wikipedia articles as Wikipedia articles are targeted at adult readers. The intuition as to why they are beneficial in assessing the quality of Wikipedia articles is not immediately self-evident, and further analysis is required to better understand their interaction with document quality.

Structural Features	Readability Scores
article length in bytes	Flesch reading score
the number of references	Flesch-Kincaid grade level
the number of links to other Wikipedia pages	Smog index
the number of citation templates	Coleman-Liau index
the number of non-citation templates	Automated readability index
the number of categories linked in the text	difficult words
the number of images / length of the article	Dale-Chall score
information noise score	Linsear write formula
article having an infobox or not	Gunning-Fog index
number of level 2 headings	
number of level 3+ headings	

Table 4.1 A list of hand-crafted features concatenated with learned representations.

learned high-level representations is one of the contributions in this work, although they have been described in Section 3.5.1.

The structural features reflect article quality in different ways. For example, the *article length* implies how much content an article contains and articles that do not contain much content are usually considered to be of low quality. *The number of references*, *number of links to other Wikipedia pages*, and *number of citation templates* show how the article editors support their content by using information from different sources, which makes the article more reliable and of higher quality. *The number of level 2 and level 3+ headings* reflect how the content is organised. Usually, Wikipedia articles of high quality have appropriate *numbers of level 2 and level 3+ headings*.

Readability scores reflect the use of language and how easy to read an article is. *Flesch reading score*, *Flesch-Kincaid grade level*, *Smog index*, and *Linsear write formula* use the average syllable per word or the number of polysyllables with different weight values to measure how difficult a text in English is to understand. Both *Coleman-Liau index* and *Automated readability index* use the average word length with different weight values to measure the readability of texts. *Difficult words*, *Dale-Chall score*, and *Gunning-Fog index* use the number of difficult words or percentage of difficult words to measure the comprehension difficulty of a text. A word is considered difficult if it is not in a list of 3000 common English words that fourth-grades American students can reliably understand.

Manually developed features are extracted using open-source packages *wikiclass*[2] and *textstat*[3]: *wikiclass* is used to extract structural features and *textstat* is used to compute readability scores.

4.1.3.3 The Proposed Hybrid Model Bi-LSTM$^+$

The proposed hybrid model, denoted as "Bi-LSTM$^+$", concatenates the neural network learned high-level representation $\boldsymbol{f}_l$ with manually developed features $\boldsymbol{f}_m$, which results in the combined feature vector $\boldsymbol{f}$. The combined features $\boldsymbol{f}$ are used to classify Wikipedia articles according to their quality. This is done by feeding the combined feature vector $\boldsymbol{f}$ into a linear layer and softmax layer to predict the probability distribution over quality classes. We use the cross-entropy between ground truth distribution and predicted distribution as the loss function to train our model.

4.1.4 Experiments

We report the results of an empirical study on the proposed hybrid model in this section.

4.1.4.1 Experimental Setting

Document preprocessing is as described in Section 3.4.1. Hyper-parameters of the proposed model are tuned on a development set. We set the word embedding dimension to 50 and the hidden size of Bi-LSTM$^+$ to 256. Then the concatenation of forward and backward LSTM gives us 512 dimensions for the article representation. The feedforward neural network produces the neural network learned high-level representation $\boldsymbol{f}_l$, which is a real-valued vector with 40 dimensions. Concatenating with manually developed features $\boldsymbol{f}_m$, which is a real-valued normalised vector with 20 dimensions, we obtain the combined features of $\boldsymbol{f}$ with 60 dimensions for each article, which are used for Wikipedia article quality classification. A linear layer and softmax layer are applied on the combined features $\boldsymbol{f}$, which produces the predicted distribution $\hat{\boldsymbol{c}}$. To save training time, articles with

[2] `https://github.com/wiki-ai/wikiclass`
[3] `https://pypi.python.org/pypi/textstat/0.3.1`

	20K	30K	60K
FA	2,414	4,920	9,908
GA	3,160	4,891	9,898
B	3,201	4,913	9,913
C	3,318	4,907	9,907
Start	4,096	4,910	9,910
Stub	4,243	4,915	9,915
Total	20,432	29,456	59,451

Table 4.2 Statistics of Wikipedia datasets used to benchmark our proposed model (BI-LSTM$^+$).

more than 350 sentences are clipped and only the first 350 sentences are used. During training, we use a mini-batch size of 128. We use Adam optimiser (Kingma and Ba, 2015) to train the model with a learning rate of 0.001. Dropout layers are applied to the input of BI-LSTM$^+$ and the neural network learned high-level representation $\boldsymbol{f}_l$ with a probability of 0.5.

4.1.4.2 Datasets

In the experiments, three different datasets with different numbers of Wikipedia articles are used: 20K dataset[4], 30K dataset[5], and 60K dataset. The Wikipedia articles in these datasets contain manually labeled quality classes, which are used as the ground truth in the experiments. The 20K dataset is provided by Warncke-Wang et al. (2015). The 30K dataset is provided by the Wikimedia Foundation. The 60K dataset is obtained by combining the 30K dataset with newly crawled articles of different quality classes. We wrote a python script to crawl articles from each quality class and talk pages were eliminated from the crawled data, which resulted in 5K articles from each quality class. We crawl about 5K articles from each quality class because there are only about 5K FA articles from the featured article category and we want the dataset to be evenly distributed. Table 4.2 summarises the quality class distributions of the three datasets.

[4]This dataset is available at: `http://figshare.com/articles/English_Wikipedia_Quality_Assessment_Dataset/1375406`

[5] This dataset is available at: `https://datasets.wikimedia.org/public-datasets/enwiki/`

	20K	30K	60K
RF	**63.70%**	58.63%	61.71%
DOC2VEC	59.84%	54.98%	61.46%
BI-LSTM	56.04%	54.36%	65.16%
BI-LSTM$^+$	63.59%	**58.98%**	**68.17%**†

Table 4.3 Performance comparisons between our proposed model (BI-LSTM$^+$) and baselines.

4.1.4.3 Experimental Results

We compare the proposed model BI-LSTM$^+$ with two state-of-the-art approaches RF (Dang and Ignat, 2016a) and DOC2VEC (Dang and Ignat, 2016c). RF only uses the structural features and readability scores as features to build a random forest as the article quality classifier. DOC2VEC uses Paragraph Vector to learn article representation and builds a classification model on the representation. The hyper-parameters of RF and DOC2VEC are based on their corresponding papers. We also compare with a model using only BI-LSTM learned high-level representation for the classification, denoted as BI-LSTM.

Table 4.3 shows the experimental results. We see that on the 20K and 30K datasets, BI-LSTM$^+$ and RF have very close performance: RF has a 0.11% higher accuracy on the 20K dataset while BI-LSTM$^+$ has a 0.35% higher accuracy on the 30K dataset. However, on the larger 60K dataset, BI-LSTM$^+$ gains a 6.5% higher accuracy than that of RF. The performance gain of BI-LSTM$^+$ is statistically significant ($p < 0.01$) on the 60K dataset, which is emphasised using a † symbol. DOC2VEC and BI-LSTM have a lower accuracy than that of BI-LSTM$^+$ in all three datasets.

4.1.4.4 Analysis

The performance of the proposed hybrid model BI-LSTM$^+$ is analysed in the following three aspects: (1) impact of hand-engineered features on dataset of different sizes; and (2) impact of hand-engineered features on models using different number of sentences in each article.

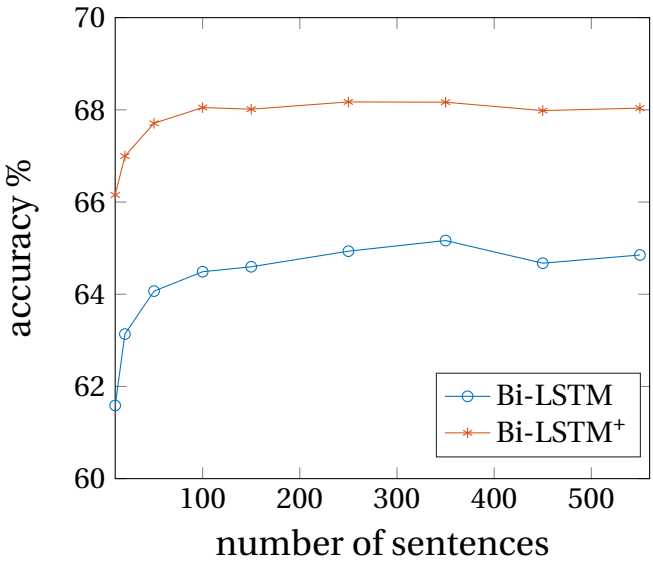


Figure 4.1 Performance of BI-LSTM and BI-LSTM$^+$ on the 60K dataset, varying the number of sentences.

Impact of manually developed features on dataset of different sizes: BI-LSTM$^+$ and RF have very close performance on the 20K and 30K datasets, which shows the effectiveness of manually developed features in article quality classification over smaller datasets. Meanwhile, the better performance of BI-LSTM$^+$ on the 60K dataset highlights the advantage of a neural network based model when there is more training data. Further, by comparing BI-LSTM with BI-LSTM$^+$, we find that BI-LSTM$^+$ still benefits from incorporating hand-crafted features, even though the improvement decreases as the dataset size gets larger: the hand-crafted features produce an accuracy improvement of 7.55% on the 20K dataset, 4.62% on the 30K dataset, and 3.01% on the 60K dataset. This can possibly be explained by that as the dataset size increases, the neural network can learn more robust features from the context and hence the performance improvement of BI-LSTM$^+$ from the manually developed features decreases, but BI-LSTM is still unable to fully capture information carried by the hand-crafted features.

Impact of manually developed features on models using different number of sentences in each article: To further justify the use of manually developed features, we vary the number of sentences fed into the models BI-LSTM and BI-LSTM$^+$, and the accuracy of the two

models on the 60K dataset is compared. We feed from the first 10 to the first 550 sentences of each article into BI-LSTM and BI-LSTM$^+$, respectively. From Figure 4.1, we can see that the performance gain of BI-LSTM$^+$ over BI-LSTM is the greatest (up to 4.57%) when the number of sentences is below 100. As the number of sentences continues to increase, the performance gain becomes smaller. This can possibly be explained by the fact that when the number of sentences per article is small, there are less features that can be learned by the neural network. The performance of BI-LSTM$^+$ can be better compensated by the manually developed features. As more sentences are processed, more features are learned by the neural network, and the contribution of manually developed features become less. We also see that, beyond 350 sentences, the performance of BI-LSTM$^+$ does not increase any more. This is because the neural network may forget the previously-learned features, which are more important, if there is too much content to learn from. Thus, we use 350 sentences per article as our default setting in the experiments.

4.1.4.5 Discussion

It is inappropriate to compare performance across different datasets, even with ones of the same size. We perform an experiment on the newly crawled 30K dataset. The accuracy of RF, DOC2VEC, BI-LSTM, and BI-LSTM$^+$ is 65.53%, 67.46%, 75.34%, and 76.33%, respectively. The accuracy of all approaches on the newly crawled 30K dataset is higher (ranging from 6.9% to 20.98%) than that on the 30K dataset provided by the Wikimedia Foundation. One reason for the performance difference is that there still are some noisy data points in the 30K dataset provided by the Wikimedia Foundation, even though we remove articles with obvious problems (e.g., an empty article is labelled as an FA article). Another reason is that there maybe some qualitative differences between the 30K dataset provided by the Wikimedia Foundations and our newly crawled 30K dataset. For example, there maybe more articles whose quality is at the boundary of adjacent classes in the 30K dataset than that of our newly crawled 30K dataset and more articles being misclassified into their adjacent classes, which results in poor performance of all approaches on the 30K dataset.

In the future, we are also interested in exploring different and effective ways of combining hand-crafted features with BI-LSTM representations. For example, we can feed both document and hand-crafted features into the model to predict both the quality of documents and reconstruct hand-crafted features, similar to Wu et al. (2018a) in exploring the utility of hand-crafted features in dealing with sequence labelling tasks.

4.1.5 Applicability to Other Domains

The proposed hybrid method that concatenates hand-crafted features engineered from the document itself with neural network learned representations can also be applied to essay scoring, academic paper rating, and content quality assessment in cQA, although the utility of our proposed method may vary across different domains.

For these three domains, especially essay scoring and content quality assessment in cQA, there are a variety of studies focusing on engineering features from the document itself before the emerging popularity of neural models, referring to Chapter 2.4 for an extensive summary of hand-crafted features explored by previous studies. Concatenating these already existing hand-crafted features with neural network learned document representations can boost the performance of models in predicting the quality of documents. Actually, Suggu et al. (2016a) and Suggu et al. (2016b) incorporate hand-crafted features with document representations learned through a CNN/LSTM to predict the quality of answers in cQA.

Furthermore, hand-crafted features concatenated with learned representations are not limited to features engineered from the document itself, although we only experiment with features from the document itself. Meta-features retrieved from other sources such as Wikipedia revision history, information of authors in academic paper rating, and the posting time of answers in cQA, can also be utilised to enhance models' capabilities to predict the quality of documents.

4.2 Enhancing Neural Models via Feature-guiding

We now turn our attention to the second contribution of this chapter: utilising hand-crafted feature indicators to guide network learning by attending to feature indicators associated with hand-crafted features present in the document. In this section, we first present the motivation for our proposed approach in Section 4.2.1, followed by related work in Section 4.2.2. Next, we detail our proposed method Section 4.2.3, along with experiments over quality assessment of Wikipedia article in Section 4.2.4 and online petition signature prediction in Section 4.2.5. Lastly, we discuss utility of our proposed method to other domains in Section 4.2.6, before concluding this chapter.

4.2.1 Motivation

The motivation underlying this work is that when hand-crafted features are represented by numerical vectors and concatenated with neural network representations, there is no information on what kind of feature each value represents. To make better use of hand-crafted features, we propose a feature-guided neural training method that guides the network to map feature indicators onto (explicit or implicit) features in the document. We evaluate the effectiveness of the proposed method over two datasets for two different tasks: (1) quality assessment of Wikipedia articles, and (2) popularity prediction of online petitions. Taking state-of-the-art approaches for the respective tasks, we achieve consistent improvements when using our model.

The closest work to our approach is the label-guided model training by Wang et al. (2018a). They embed words and labels in the same embedding space, and compute a label-based attention score between a word and all possible labels, which is used to weight word embeddings in obtaining document representations. Our work differs in two aspects: (1) Wang et al. (2018a) capture direct associations between labels and words, while we use the proxy of (potentially much higher-level) hand-crafted features to guide network learning; and (2) our method does not rely on the target variable being closely related to the semantics of a document, leading to better generalisability.

4.2.2 Related Work

Studies employing neural models either eschew hand-crafted features or simplistically use hand-crafted features as side information. For example, Dang and Ignat (2017) propose to use a bidirectional LSTM ("BI-LSTM") to classify Wikipedia articles by their quality classes, and in Section 4.1 we concatenate structural features (e.g., article length) and readability scores with BI-LSTM-learned document representations for the same task. Subramanian et al. (2018) hand-engineer a set of features (e.g., the ratio of indefinite and definite articles), and concatenate them with CNN-learned document representations to predict the popularity of online petitions. Wu et al. (2018a) explore the utility of hand-crafted features in sequence labelling via multi-task training and an auto-encoder. Specifically, Wu et al. (2018a) concatenate hand-crafted features with character representations learned via a CNN and word embeddings. These representations are then fed into a BI-LSTM to identify named entities (with the help of a CRF) and re-construct the hand-crafted features in the output simultaneously, which is achieved by combining an auto-encoder loss with the NER loss. Suggu et al. (2016b) combine features, such as the presence of URLs and exclamation marks in an answer, with CNN/BI-LSTM learned document representations to predict the quality of an answer in cQA platforms (e.g., Stack Overflow[6]).

4.2.3 Methodology

Figure 4.2 is an illustration of our proposed approach, in the context of a stacked BI-LSTM, where two BI-LSTMs are applied to obtain the sentence and document level representations, respectively. Note that our method is not limited to LSTMs, as we show in Section 4.2.4 and Section 4.2.5.

A hand-crafted feature can consist of multiple feature indicators. For example, having *level 3+ headings* is one of the features in quality assessment of Wikipedia articles. This hand-crafted feature consists of two feature indicators (tokens) {"===", "===="}. We embed

[6] `https://stackoverflow.com/`

the document and feature indicators into a shared space. Then, as indicated in Figure 4.2, for each feature indicator, we compute cosine similarity between the feature indicator and word embeddings, followed by average-pooling to obtain a sentence score:

$$\text{score} = \frac{1}{N}\sum_{j=1}^{N}\frac{\mathbf{F}\mathbf{V}_j}{\|\mathbf{F}\|_2\|\mathbf{V}_j\|_2}. \tag{4.1}$$

Here, $\mathbf{F}$ and $\mathbf{V}_j$ are embeddings of the feature indicator and the jth word in a sentence, respectively; $\|\mathbf{F}\|_2$ and $\|\mathbf{V}_j\|_2$ are the ℓ_2 norms of $\mathbf{F}$ and $\mathbf{V}_j$, respectively; and N is the number of words in a sentence. All scores based on the feature indicator are concatenated with sentence representations $\mathcal{Z}_1$, which are learned through a Bi-LSTM layer (f_1). Then, another Bi-LSTM layer (f_2) is applied to the concatenated sentence representations to obtain document representation $\mathcal{Z}_2$, which is followed by a dense layer (f_3) to compute $\boldsymbol{y}$.

The score computed in Equation 4.1 is for a single-token feature indicator. If a hand-crafted feature consists of multiple feature indicators (tokens, e.g., {"===", "===="}), the score becomes:

$$\text{score} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N}\sum_{j=1}^{N}\frac{\mathbf{F}_i\mathbf{V}_j}{\|\mathbf{F}_i\|_2\|\mathbf{V}_j\|_2}. \tag{4.2}$$

Here, M is the number of feature indicators in a hand-crafted feature, and $\mathbf{F}_i$ is the word embedding of the ith feature indicator in the feature.

For example, the feature *level 3+ headings* is one of structural features described in Table 3.7, which consists of two feature indicators {"===", "===="}. To obtain the similarity score for the feature *level 3+ headings*, we first compute the similarity score between each feature indicator "==="/"====" and word embeddings in the sentence, then apply average-pooling to obtain the similarity score for each feature indicator. We obtain the similarity score of *level 3+ headings* by averaging similarity scores among the feature indicators at the sentence level. Finally, the feature score is concatenated with the sentence representation, which is fed into a latter layer.

While we do not experiment with this, it is also possible to first average feature indicator embeddings and then compute the sentence score by Equation 4.1. This way, we can efficiently reduce the computation of similarity scores for hand-crafted features with

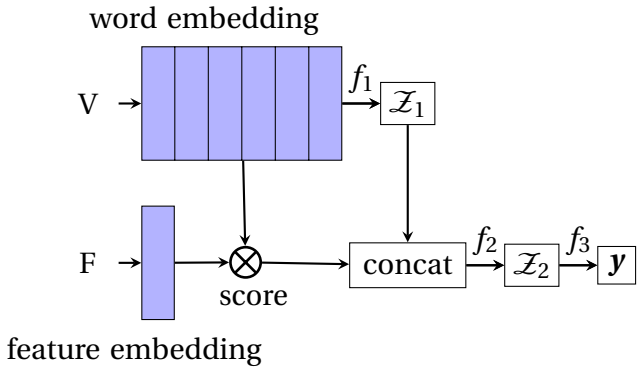


Figure 4.2 Illustration of our proposed method (`Feature-guided`). Here: $\otimes$ denotes cosine similarity between the feature indicator and word embeddings; f_1 and f_2 denote Bi-LSTM layers; f_3 denotes a dense layer; $\mathcal{Z}_1$ and $\mathcal{Z}_2$ denote sentence and document representations, respectively; $\boldsymbol{V}$ and $\boldsymbol{F}$ are the document input and feature indicators, respectively; and $\boldsymbol{y}$ is the target output.

a large number of feature indicators. In the experiments, we use Equation 4.2, as the maximum number of feature indicators in a given hand-crated feature is less than 1,000, and less than 10 in most cases.

4.2.4 Experiments on Quality Assessment of Wikipedia Articles

To test the effectiveness of our proposed method, we experiment with a Wikipedia document quality assessment task (Dang and Ignat, 2017), and online petition signature prediction task (Subramanian et al., 2018), as detailed below. The reasons we chose these particular tasks are as follows. First, extensive domain-specific feature engineering had taken place in each case, that we could use as the basis of our feature indicators. Second, strong neural benchmarks have been established, based on extensive experimentation with both neural and non-neural models.

We aim to explore the relative gains of our proposed method relative to the current state-of-the-art for the task, which in both cases is not based on contextualised embeddings. For BERT (Devlin et al., 2019) or other contextualised encoders, the same word in different contexts will end with different embeddings, leading to localised representations

of feature indicators. As such, the proposed method is not directly applicable to models such as BERT, and novel research would be required to adapt the method to such models.

Following Dang and Ignat (2017) and Chapter 3, we formulate the quality assessment of Wikipedia articles as a multi-class classification problem, and all models are trained to minimise cross-entropy loss. We report average `accuracy` and standard deviation over 10 runs.

4.2.4.1 Dataset

The Wikipedia dataset (see Section 3.4.2.1) consists of 29,794 English Wikipedia articles and their corresponding quality labels: FA, GA, B, C, Start, and Stub, in descending order of document quality. The dataset is class-balanced and partitioned into training, development, and test splits (8:1:1). Statistics of the dataset is shown in Table 3.1 in Chapter 3. Documents are relatively long, and processed in a hierarchical manner, by constructing sentence representations, and composing these into a document representation.

4.2.4.2 Model Configuration

We apply our proposed approach (`Feature-guided`) over the four models detailed below. In each case, we contrast with two baselines: (1) `Vanilla`, makes no use of hand-crafted features; and (2) `Side-information` which uses the hand-crafted features as side information, by concatenating them with learned representations in the penultimate layer.

1. CNN_Bi-LSTM: apply convolution kernels with width 2, 3, and 4 (32 for each width size) to word embeddings within a sentence, and a tanh activation function to each; pass the output of the filters through a Bi-LSTM.

2. Average_Bi-LSTM (Chapter 4.1): average word embeddings to get the sentence representation, and run a Bi-LSTM over the sequence of sentence representations.

3. Stacked_Bi-LSTM: feed the word embeddings in a sentence through a Bi-LSTM, and the output through a max-pooling layer; finally, apply another Bi-LSTM over the sentence representations.

4. STACKED_BI-LSTM_ATT (Yang et al., 2016): use a hierarchical STACKED_BI-LSTM, except that an attention mechanism with a context size of 100 is applied to the output of each BI-LSTM to weight words/sentences based on their importance in the sentence/document.

A max-pooling layer is applied to the output of the BI-LSTM at the sentence level for all models except STACKED_BI-LSTM_ATT to get the document level representation, which is followed by two dense layers, one with a ReLU activation and one without any activation function. For all models, dropout layers are applied at both the sentence and document levels with a rate of 0.5 during training. For both CNN_BI-LSTM and AVERAGE_BI-LSTM, the BI-LSTM cell size is set to 256. For STACKED_BI-LSTM and STACKED_BI-LSTM_ATT, the cell size is set to 32 and 256 for the sentence and document level BI-LSTM, respectively.

Similar to Section 3.4.1 and Section 4.1.4.1, we use 50-dimensional pre-trained word embeddings from GloVe (Pennington et al., 2014).[7] For OOV words, the word embeddings are randomly initialised based on sampling from a uniform distribution $U(-1,1)$. All word embeddings are updated in the training process. We use a mini-batch size of 128 and a learning rate of 0.01. We train each model for 50 epochs. To prevent over-fitting, early stopping is adopted. All hyper-parameters are set empirically over the development data, and the models are optimised using Adam (Kingma and Ba, 2015).

Hand-crafted features used to guide network learning here include: (1) reference indicators; (2) links to other Wikipedia page indicators; (3) citation template indicators; (4) non-citation template indicators; (5) categories linked in the article indicators; (6) image indicators; (7) infobox indicators; (8) level 2 heading indicators; and (9) level 3+ heading indicators. These features are from Dang and Ignat (2016a) and Table 3.7. Hand-crafted features in `Side-information`, described next, are based on counting the number of appearances of such feature indicators.

[7] We fine-tuned hyper-parameters over the development set for quality predictions of Wikipedia articles. 50-dimensional embeddings were chosen because `Vanilla` performs the best under this setting (meaning the baseline without features is as strong as possible).

Model	CNN_Bi-LSTM	Average_Bi-LSTM	Stacked_Bi-LSTM	Stacked_Bi-LSTM_att
`Vanilla`	57.12±0.58%	57.91±0.81%	57.60±0.65%	56.70±1.21%
`Side-information`	57.24±0.47%	59.04±0.33%	57.97±0.74%	57.44±0.62%
`Feature-guided`	**58.10±0.50%**†	**59.90±0.45%**†	**58.30±0.71%**	**58.30±0.65%**†

Table 4.4 Accuracy (%) over Wikipedia dataset for our proposed method (`Feature-guided`) and various baselines. The best result is in **bold**, and marked with "†" if the improvement is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$).

4.2.4.3 Results

The experimental results are presented in Table 4.4. We can see that our method outperforms both `Vanilla` and `Side-information` across all four network architectures, at a level of statistical significance for 3 out of the 4 models. It is worth noting that the performance of Stacked_Bi-LSTM_att is worse than that of Stacked_Bi-LSTM for both `Vanilla` and `Side-information`, due to attention in Stacked_Bi-LSTM_att not being as effective as max-pooling on this task. However, the performance of Stacked_Bi-LSTM_att is not worsened by incorporating the attention for our method, indicating that our feature-guided learning can guide the network to learn better.

4.2.4.4 Further Results

In Equations (4.1) and (4.2), sentence scores are computed by averaging cosine similarities between the feature indicator and word embeddings, thereby paying attention to feature indicator tokens or tokens similar to feature indicators. We further explore how our proposed method works if sentence scores are based on the maximum similarity score between the feature indicator and word embeddings rather than the average score, thereby only capturing the most salient information. We also experiment with combining the afore-

Model	CNN_Bi-LSTM	Average_Bi-LSTM	Stacked_Bi-LSTM	Stacked_Bi-LSTM_att
`Vanilla`	57.13±0.95%	58.27±0.64%	57.61±0.97%	57.17±0.87%
`Side-information`	57.34±0.69%	59.33±0.58%	57.81±0.89%	57.71±0.79%
`Feature-guided`^{mean}	**57.98±0.64%**	**59.61±0.91%**	**58.40±0.86%**	**58.16±0.74%**
`Feature-guided`^{max}	56.94±0.84%	58.73±0.96%	57.35±0.73%	57.30±0.78%
`Feature-guided`^{combine}	57.91±0.50%	59.53±0.58%	57.90±0.54%	57.86±0.74%

Table 4.5 Accuracy (%) over Wikipedia dataset for our proposed method `Feature-guided` with different pooling strategies. The best result is in **bold**.

mentioned two strategies. The averaging strategy is denoted as `Feature-guided`^{mean}; the maximum strategy is denoted as `Feature-guided`^{max}; and the method adopting these two strategies is denoted as `Feature-guided`^{combine}.

Table 4.5 shows the experimental results adopting different strategies, across different architectures. We can see that `Feature-guided`^{mean} achieves the best performance, noting that the exact numbers are different from those in Table 4.4, due to the randomness of model initialisation. `Feature-guided`^{max} achieves slightly better performance than `Vanilla` within the architectures of Average_Bi-LSTM and Stacked_Bi-LSTM_att, while it is the opposite for CNN_Bi-LSTM and Stacked_Bi-LSTM, indicating the limited capacity of using the maximum operation within our proposed method. The performance of `Feature-guided`^{combine} is close to that of `Feature-guided`^{mean}, although minor deterioration is observed, which is worth exploring in the future.

4.2.5 Experiments on Online Petition Signature Prediction

To further verify the effectiveness of our proposed method, we perform experiments over the task of online petition signature prediction, i.e., to predict the popularity of a given online petition.

We formulate signature count prediction as a regression problem, and all models are trained to minimise mean squared error. For evaluation, average mean absolute error (MAE) and mean absolute percentage error (MAPE) over 10 runs are reported. Here, MAPE is calculated as $\frac{100}{n}\sum_1^n \frac{|\hat{y}_i - y_i|}{y_i}$, where $\hat{y}_i$ and y_i are the predicted and ground-truth signature counts.

4.2.5.1 Dataset

The online petition dataset (Subramanian et al., 2018) consists of 10,950 UK petitions and their corresponding signature counts. Following Subramanian et al. (2018), we chronologically split the data into training, dev, and test (8:1:1).

4.2.5.2 Model Configuration

We again compare our proposed `Feature-guided` approach with `Side-information` and `Vanilla`, in the form of a state-of-the-art CNN with convolution kernels of width 1, 2, and 3 (100 kernels for each width size) over word embeddings, with a ReLU applied to each. The outputs are passed through two dense layers, one with a tanh activation function and one with an ELU activation function, to obtain the final output.

A dropout layer is applied to the output of the convolution filters at a rate of 0.5 during training. We use a mini-batch size of 32, and a learning rate of $1e-4$. All other hyper-parameters are the same as in the Wikipedia setting, except that the early stopping is based on MAE.

Hand-crafted features used to guide network learning here include: (1) indefinite vs. definite articles; (2) P1 singular and plural, P2, and P3 singular and plural pronouns; (3) subjective, positive, and negative words; and (4) biased words. Specifically, subjective, positive, and negative words are based on General Inquirer lexicon (Stone et al., 1962), which categorises words based on their valence. Biased words are from the bias lexicon (Recasens et al., 2013). These features are from Subramanian et al. (2018). Hand-crafted features in `Side-information` are based on counting the number of appearances of such feature words.

Model	MAE	MAPE
Vanilla	1.44	38.1
Side-information	1.45	39.0
Feature-guided	**1.42**†	**36.2**†

Table 4.6 Performance of our proposed method (Feature-guided) over online petitions. The best result is indicated in **bold**, and marked with "†" if the improvement is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$).

4.2.5.3 Results

Table 4.6 summarises our results. We observe that our approach benefits Vanilla and Side-information once again, at a level of significance in terms of both MAE and MAPE. It is worth noting that Side-information performs worse than Vanilla, as it over-fits to features only present in the training data. In comparison, our method improves the model performance even in this case, as it identifies words semantically related to the feature indicators. For example, the word *increase*, not in the predefined list of positive words, has a high similarity score (> 0.8) with positive words *help*, *hope*, *give*, and *allow*.

4.2.6 Applicability to Other Domains

Our proposed method of guiding network learning by attending to feature indicators associated with hand-crafted features can be applied to other domains as long as feature indicators in these domains are easy to define and detect. Our method can be applied directly to other domains if the utility of hand-crafted features in improving performance has been demonstrated and it is feasible to define feature indicators associated with these hand-crafted features; otherwise, modifications are required to make our method suitable for other domains. For example, for content quality assessment in cQA, it is easy and simple to identify feature indicators for some features, such as the number of links and email addresses in answers, while it is difficult and complex to identify feature indicators for other features, such as various readability scores.

An alternative is to use keywords as proxies of hand-crafted features to guide network learning by attending to keywords or words similar to keywords in the document. In

academic paper rating, keywords can be selected from the paper title and abstract, which can be used to guide the network to obtain better paper representations. In essay scoring, we can select topic words in the prompt as keywords, which in turn can be used to guide network learning. In content quality assessment in cQA, keywords from the question can be used to attend to important words in the answer, thus leading to better prediction of document quality. Here, keywords can be identified based on information gain or external resources.

4.3 Summary

In this chapter, we addressed the second research question of verifying the utility of hand-crafted features in neural networks and incorporating such features with learned representations based on different approaches: (1) concatenating hand-crafted features with neural model learned representations; and (2) utilising hand-crafted features to guide document representation learning by explicitly attending to features indicators present in the documents. While these two proposed methods differ in how they make use of hand-crafted features, their superiority over baselines has been demonstrated by experiments across different datasets, confirming our hypotheses and motivations that neural models can benefit from existing hand-crafted features.

In the next chapter, we turn our attention to measuring the uncertainty in document quality predictions and discussing its potential application in the real world, considering the inherent subjectivity of quality assessment in evaluating the performance of quality assessment systems.

Chapter 5

Modelling Uncertainty in Collaborative Document Quality Assessment

The focus of Chapter 3 and Chapter 4 was on enhancing performance of neural models, either by learning visual features or by incorporating hand-crafted features into neural networks. In this chapter, we address the third research question – how we can take the inherent subjectivity of quality assessment into consideration in evaluating the performance of quality assessment systems, to imitate people's disagreement over the task. We base our approaches on two methods: Gaussian processes (GPs) and random forests (RFs), both of which can provide predictive distributions, making it feasible to quantify their uncertainty over the predictions. Lacking a large-scale dataset consisting of multiple quality labels from different people, we first construct such a dataset, which enables the imitation of people's disagreement over the inherent subjectivity of document quality assessment. Experimental results show that both GPs and RFs can yield competitive results in predicting the quality of Wikipedia articles, with the advantage of providing an estimate of uncertainty in their predictions. We additionally evaluate our methods in the context of a semi-automated document quality class assignment decision-making process, where there is asymmetric risk associated with overestimates and underestimates of document quality. Our experiments suggest that GPs provide more reliable estimates in this context.

We first present the motivation for modelling uncertainty in collaborative document quality assessment in Section 5.1, followed by related work in Section 5.2. Next, we describe problem definition and dataset collection in Section 5.3 and methods explored in this work in Section 5.4. Lastly, we present experimental results under different scenarios in Section 5.5, before discussing applicability of our work to other domains in Section 5.6 and finishing this chapter.

This chapter is based on and extends the following paper:

> Aili Shen, Daniel Beck, Bahar Salehi, Jianzhong Qi and Timothy Baldwin. 2019. Modelling Uncertainty in Collaborative Document Quality Assessment. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 191–201, Hong Kong, China.

5.1 Motivation

The volume of textual web content generated collaboratively – through sites such as Wikipedia, or community question answering platforms such as Stack Overflow — has been growing progressively. Such collaborative paradigms rise the necessity for document quality assessment: how to ensure documents are reliable and useful to end users.

Given the volume of such documents, and velocity with which they are being produced, there has been recent interest in *automatic* quality assessment using machine learning techniques (Dang and Ignat, 2016a, Dalip et al., 2017). However, previous work has treated this problem using off-the-shelf predictors, which fail to take into account two key aspects. First, any quality rating is inherently *subjective*: different end users can heavily disagree on the quality of a document. For example, as shown in Figure 5.1, the Wikipedia article *Warden Head Light*[1] is assigned to different labels from different Wikipedia Projects:[2] B (in the green block) by Wikipedia Project *Lighthouses*, and Start (in the orange block) by

[1] `https://en.wikipedia.org/w/index.php?title=Warden_Head_Light&oldid=759074867`

[2] A Wikipedia Project is a group of Wikipedia contributors who work together to improve Wikipedia articles that they are interested in.



Figure 5.1 A screenshot of the "Warden Head Light" Talk page. Wikipedia Project *Lighthouses* assigns a B-class quality label to this article, while Wikipedia Project *Australia* assigns a Start-class quality label.

Wikipedia Project *Australia*;[3] among a 30K datatset we collected, there are 7% such articles (even including high-quality articles), where contributors disagree over the article quality. Second, previous work has ignored *decision-making* procedures (such as expert reviewing, and featuring articles on the Wikipedia main page) that are impacted by the results of the prediction, which can vary in non-trivial ways.

In this work, we address these two gaps by modelling the *uncertainty* in the quality labels by treating predictions as probability distributions. In order to obtain these distributions, we experiment with both Bayesian models (Gaussian Processes, GPs: Rasmussen

[3] We return to describe the full label set in Section 5.3.

and Williams (2006)) and frequentist, ensemble-based methods (Random Forests, RFs: Breiman (2001)), applying them to English Wikipedia articles. Our results show that these approaches are competitive with the state-of-the-art in terms of predictive performance, while also providing estimates of uncertainty in the form of predictive distributions.

As a case study on the utility of uncertainty estimates, we analyse a typical Wikipedia scenario, where articles with predicted high quality are sent to expert reviewers to confirm their status. Such reviewing procedures are costly: if a low-quality article is predicted to be a featured article (the highest quality in Wikipedia), the triggered manual review can substantially waste time and human effort. Conversely, if a high-quality article is predicted to be of a lower-quality class, there is no cost to the editor community.[4] This is an example of asymmetric risk, where underestimates and overestimates have different penalties. In this work, we show how to use uncertainty estimates from predictions in order to make a quality prediction that minimises this asymmetric risk.

5.2 Related Work

Most work on quality assessment does not take the subjectivity of the task into consideration. See Section 2.4.1 and Section 2.5.1 for a comprehensive survey of related work on quality assessment of Wikipedia articles.

Uncertainty has been measured in other tasks. For example, Beck et al. (2016) explore prediction uncertainty in machine translation quality estimation (QE), where post-editing rate is the dependent variable. In QE, the post-editing rate — which is computed by dividing the post-editing time by the length of the translation hypothesis — is a positive real value. The performance of a GP model was studied in both underestimate and overestimate scenarios. Beck and Cohn (2017) and Beck (2017) employ GPs to model text representation noise in emotion analysis, where Pearson's correlation and NLPD are used as the evaluation metrics. Our work is different from these two studies as we model the

[4] Although there may be an opportunity cost (in terms of not showcasing high-quality articles), and the potential de-motivation of the associated editors.

subjectivity of quality assessment explicitly, which can mimic people's different opinions over the quality of a document.

There is also a rich body of work on identifying trustworthy annotators and predicting the correct underlying labels from multiple annotations (Hovy et al., 2013, Cohn and Specia, 2013, Passonneau and Carpenter, 2014, Graham et al., 2017, Paun et al., 2018). For example, Hovy et al. (2013) propose an item-response model to identify trustworthy annotators and predict the true labels of instances in an unsupervised way. However, our task is to measure the uncertainty of a model over its predictions (as distinct from attempting to learn the "true" label for an instance from potentially biased/noisy annotations), in addition to correctly predicting the gold label, in the context of assessing the quality of Wikipedia articles. Additionally, we have a rich representation of the data point (i.e., document) that we are attempting to label, whereas in work on interpreting multiply-annotated data, there is little or no representation of each data point that has been annotated. Finally, we do not have access to the IDs of annotators across documents, and thus cannot model annotator reliability or bias.

5.3 Preliminaries

In this section, we detail the specific scenario addressed in this study: quality assessment of Wikipedia articles. We also describe the procedure to construct our dataset.

5.3.1 Problem Definition

In line with previous work (Warncke-Wang et al., 2015, Dang and Ignat, 2016a,c, 2017), we consider six quality classes of Wikipedia articles, ordered from highest to lowest: Featured Article ("FA"), Good Article ("GA"), B-class Article ("B"), C-class Article ("C"), Start Article ("Start"), and Stub Article ("Stub"). A description of the quality grading criteria can be found in the Wikipedia grading scheme page.[5]

[5] https://en.wikipedia.org/wiki/Template:Grading_scheme

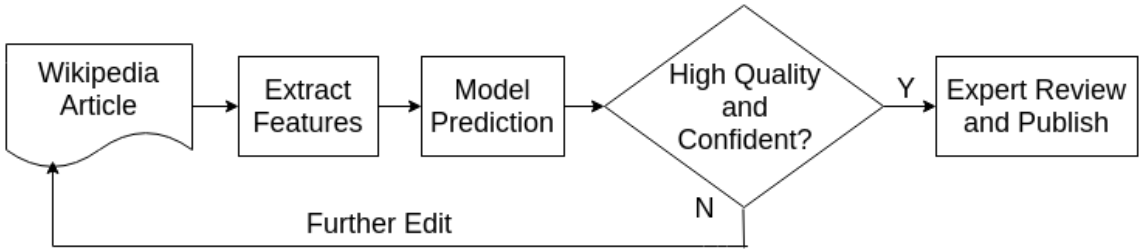


Figure 5.2 Model application and decision-making procedure.

The quality assessment process over a Wikipedia article is done in a collaborative way, through discussions on the corresponding article's Talk page.[6] Wikipedia contributors also carefully review articles that are GA and FA candidates. In particular, FA articles are eligible to appear on the main page of the website. A reliable automatic quality assessment model should take these decision making aspects into account.

In this work, our aim is to predict the quality of Wikipedia articles, paired with an estimate of uncertainty over each prediction, which we evaluate in a risk-aware decision making scenario. Figure 5.2 summarises model application and actions depending on the uncertainty: (1) quality-indicative features are first extracted from a Wikipedia article; (2) a model predicts the article quality and provides an indication of how confident it is of its prediction; and (3) different actions are taken based on the predicted quality and confidence value, such as expert review and featuring on the Wikipedia main page if an article is predicted to be FA/GA with high confidence.

5.3.2 Data Collection

We constructed an evaluation dataset by collecting articles from Wikipedia in a balanced way. From each quality class, we crawled 5K articles from its corresponding repository.[7] As mentioned in Section 5.1, quality assessment is subjective and multiple editors/Wikipedia Projects may disagree when assigning a quality label to an article. We can observe this

[6] Such as the Talk page for the "Warden Head Light" article: `https://en.wikipedia.org/wiki/Talk:Warden_Head_Light`

[7] For example, we obtain FA articles by crawling pages from the FA repository: `https://en.wikipedia.org/wiki/Category:Featured_articles`

behaviour by inspecting an article's corresponding Talk page, which records quality labels from different Wikipedia Projects.[8] For roughly 7% of the articles, there is a disagreement between editors/Wikipedia Projects. Take Figure 5.1 in Section 5.1 as an example, although the *primary* label of the Wikipedia article *Warden Head Light* is B, two other quality labels are assigned to it: B class (in the green block) by Wikipedia Project *Lighthouses*, and Start class (in the orange block) by Wikipedia Project *Australia*. Since we are interested in investigating how an automatic quality assessment system performs when there is a disagreement, we also crawl these *secondary* labels when building our dataset. Finally, we remove markup that relates to the document quality classes, such as *{Featured Article}* or *{geo-start}*, to alleviate any overt indication of the quality label in the text of the article.

The resulting dataset contains 29,097 Wikipedia articles, which we partition into two subsets for separate evaluation in Section 5.5: (1) *consistent* articles, where *primary* and *secondary* labels fully agree; and (2) *inconsistent* articles, where there is disagreement among *secondary* labels, with at least one of them agreeing with the *primary* label. We emphasise that we keep *all secondary* labels for the latter, without performing any label aggregation (e.g., voting). Our aim is to make quality predictions as close as possible to the *primary* labels while also providing uncertainty estimates of such predictions: lower uncertainty over *consistent* articles and higher uncertainty over *inconsistent* articles. The dataset is then stratified into training, development, and test sets, as detailed in Table 5.1.

5.4 Methods

A key aspect of the task is the ordinal nature of the quality labels, e.g., a Start article is close in quality to a C, but much worse than an FA. Surprisingly though, most previous studies (Dang and Ignat, 2016a,c, 2017) formulate the problem as multi-class classification and use accuracy as the evaluation metric as mentioned in Section 2.3.4, such modelling and evaluation procedures completely disregard the ordinal nature of the labels, which in turn does not correspond to real world scenarios: the cost of mispredicting a Start article as C

[8] Different Wikipedia articles can be rated by different Wikipedia Projects. And the number of quality labels in a Talk page depends on how many Wikipedia Projects rate this article.

		Train	Dev	Test	Total
FA	consistent	3956	470	538	4998
	inconsistent	28	4	2	
GA	consistent	3887	468	495	4878
	inconsistent	16	6	6	
B	consistent	3138	400	416	4843
	inconsistent	702	85	102	
C	consistent	3036	382	381	4523
	inconsistent	570	69	85	
Start	consistent	3725	451	472	4924
	inconsistent	223	28	25	
Stub	consistent	3863	470	492	4931
	inconsistent	83	12	11	
Total	—	23227	2845	3025	29097

Table 5.1 Statistics of the Wikipedia dataset consisting of *consistent* and *inconsistent* articles, which are used to benchmark models' performance in providing uncertainty estimates of their predictions.

is different to mispredicting it as an FA (standard classification metrics such as accuracy assume equal cost for all mispredictions).

To better address the scenarios we are interested in, we treat quality assessment as a regression problem, in terms of both modelling and evaluation. In order to do this, we encode the quality class labels as real values by mapping them to the interval $[-2.5, 2.5]$ with increments of 1. These labels are -2.5, -1.5, -0.5, 0.5, 1.5, and 2.5, respectively, where higher values indicate higher quality.[9] We perform this step to be able to use off-the-shelf regression models, while also centering the labels. The remainder of this section details the regression methods we use, as well as two types of features we employ to represent each article. Both methods provide uncertainty estimates through *predictive distributions* (Gaussian distributions in our case).

[9] Having equal intervals is a heuristic: we discuss this limitation in Section 5.7.

5.4.1 Gaussian Processes

A principled approach to obtain predictive distributions is to use GPs (Rasmussen and Williams, 2006), a Bayesian non-parametric framework widely considered the state-of-the-art for regression (Hensman et al., 2013). Given a latent function f, which explains the relationship between an input vector $\boldsymbol{x}$ and its corresponding output value y, the model assumes that f is distributed according to a GP, i.e.,

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})),$$

where $m(\boldsymbol{x})$ is a mean function, and $k(\boldsymbol{x}, \boldsymbol{x'})$ is a covariance or *kernel* function.

Following common practice, we fix the mean function to zero, as our output values are centered. Most of the information obtained from the training data can be encoded in the kernel function, of which the most common one is the Radial Basis Function (RBF), defined as

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \sigma_v \exp\left(-\frac{1}{2}\sum_{j=1}^{d}\frac{1}{\ell_j^2}(x_j - x'_j)^2\right),$$

where σ_v is the variance hyperparameter controlling the scale of the labels, and ℓ_j is the lengthscale for the jth dimension of the input. The lengthscales are learned by maximising the marginal likelihood (Rasmussen and Williams, 2006), resulting in a feature selection procedure known as *Automatic Relevance Determination* (ARD): lower lengthscales indicate features with higher discriminative power. We use this procedure to perform a feature analysis in Section 5.5.1. Besides the RBF, we also experiment with a range of other kernels used in GP models: see Rasmussen and Williams (2006, Chap. 4) for details.

Standard GP inference takes $O(n^3)$ time, where n is the number of instances in the training data. As this is prohibitively expensive given the size of our dataset, we employ Sparse GPs (Titsias, 2009, Gal et al., 2014), a scalable extension that approximates an exact GP by using a small set of latent *inducing points*. These are learned by maximising a variational lower bound on the marginal likelihood: see Titsias (2009) for details.

5.4.2 Random Forests

As an alternative method to obtain predictive distributions, we use RFs (Breiman, 2001), which are ensembles of decision trees (regression trees in our case). Each tree is trained on a bootstrapped sample of the training set and within each tree, a random subset of features is used when splitting nodes. To obtain predictive distributions, we assume that the individual tree predictions follow an "empirical" Gaussian distribution. The mean and the variance are computed from the full set of predictions obtained by the RF. While this approach is less principled — since there is no reason to believe the distribution over individual predicted values is Gaussian — it can work well in practice. For instance, RFs have been used before to obtain uncertainty estimates in the context of Bayesian Optimisation (Hutter et al., 2011).

5.4.3 Features and Preprocessing

Hand-crafted features used in this work include 11 structural features and 10 readability scores (see Section 3.5.1). Features from the revision history, such as the number of revisions and the article–editor network, are indirect quality indicators, thus we only focus on direct quality indicators from the content itself.

5.5 Experimental Study

In this section, we detail four sets of experiments: (1) intrinsic comparison of our methods with respect to their predictive distributions; (2) comparative experiments with the state-of-the-art with respect to point estimates only; (3) experiments measuring the performance of our methods in a transfer setting, where the goal is to predict *secondary* labels; and (4) a case study where automatic labels are used to filter articles for manual revisions and we use distributions to incorporate risk in the quality predictions. All our models are trained on the *primary* labels but we explicitly report our results on two different test sets: one with *consistent* and one with *inconsistent* labels, as explained in Section 5.3.2.

5.5.1 Intrinsic Evaluation

Our first set of experiments evaluates the performance of methods intrinsically, with respect to their predictive distributions.

5.5.1.1 GP Settings

We use GP models from the *GPflow* toolkit (Matthews et al., 2017). In particular, we use a Sparse GP with 300 inducing points, which are initialised with k-means clusters learned on the training set and we explore different kernels (RBF, Arccosine (Cho and Saul, 2009), Rational Quadratic (RQ), Matérn 32, and Matérn 52). These are the hyper-parameters that we have used in our experiments.

5.5.1.2 RF Settings

We use the RF implementation in *scikit-learn* (Pedregosa et al., 2011), with 300 trees and a maximum depth of 40, fine-tuning over the development set. All other hyperparameters are set to default values.

5.5.1.3 Evaluation Metrics

Standard metrics to evaluate regression models such as Root Mean Squared Error (RMSE) and Pearson's correlation (r) are only based on *point estimate* predictions. These are not ideal for our setting since we aim to assess predictive *distributions* instead. For such settings, Candela et al. (2005) proposed the Negative Log Predictive Density ("NLPD") as an alternative metric, which is commonly used in the literature (Chalupka et al., 2013, Hernández-Lobato and Adams, 2015, Beck et al., 2016) to evaluate probabilistic regression models.

Given a test set containing the input and its reference score $(\boldsymbol{x}_i, y_i)$, NLPD is defined as

$$\text{NLPD} = -\frac{1}{n}\sum_{i=1}^{n} \log p(\hat{y}_i = y_i | \boldsymbol{x}_i),$$

	NLPD (*consistent*)	NLPD (*inconsistent*)
RF	**0.978**†	1.642
GP_{RBF}	1.224	**1.364**†
GP_{arc0}	1.280	1.460
GP_{arc1}	1.266	1.428
GP_{arc2}	1.286	1.426
$GP_{Matérn32}$	1.275	1.427
$GP_{Matérn52}$	1.275	1.425
GP_{RQ}	1.271	1.442

Table 5.2 Performance comparisons between RF and GPs with different kernels in predicting the primary quality label. GP_{arc0}, GP_{arc1}, and GP_{arc2} denote GP using an Arccosine kernel with orders of 0, 1, and 2. $GP_{Matérn32}$, $GP_{Matérn52}$, and GP_{RQ} denote GP using Matérn32, Matérn52, and RQ, respectively. The best result (the lower the better) is indicated in **bold**, and marked with "†" if the improvement is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$).

where n is the number of test samples and $p(\hat{y}_i|\boldsymbol{x}_i)$ is the predictive distribution for input $\boldsymbol{x}_i$. For Gaussian distributions, NLPD penalises both overconfident wrong predictions and underconfident correct predictions, where a lower NLPD indicates better performance.

Both log likelihood and cross entropy relate to NLPD in the context of evaluating point estimate predictions, although they are most commonly used as loss functions to optimise model parameters rather than evaluation metrics.

5.5.1.4 Results

Table 6.3 shows the average NLPD over 10 runs on both test sets. Clearly, RFs outperform GPs on the consistent set while the opposite happens on the inconsistent set. In general, this shows that GPs tend to give more conservative predictive distributions compared to RFs. This is beneficial when there is label disagreement, and therefore, high uncertainty over the labels. However, it also translates into worse performance when labels are consistent, where the higher confidence obtained by RFs give better results. In terms of kernels, we obtained the best results using RBF for both test sets.

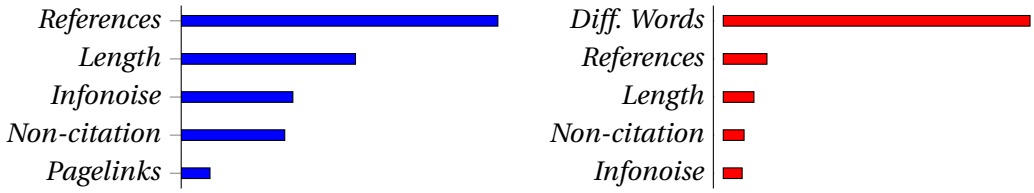


Figure 5.3 Feature importance values in the GP model (left) and RF model (right). A full description of all features can be found in Dang and Ignat (2016a) and Table 3.7.

5.5.1.5 Feature Analysis

To find out which features contribute most to the performance of our models, we analyse the lengthscales from GP and the feature weights from RF. Figure 5.3 shows the top five features in each model: four of those shared by both GP and RF, indicating their importance as quality indicators in both methods. In particular, the length of an article is a consistently good indicator: a possible explanation is that short articles lack enough content to be considered high quality articles. The presence of structured indicators such as number of references and non-citation templates is also interesting, as it is evidence of the contribution of non-textual information in the quality assessment process.

5.5.2 Point Estimate Comparison

5.5.2.1 Experimental Settings

In terms of point estimates, the state-of-the-art in our task is a neural model based on a BI-LSTM architecture used in Chapter 4.1. As we model quality prediction of Wikipedia articles as a regression problem, where a linear transformation is used as the final layer instead of a softmax, the neural network model does not provide predictive distributions, limiting their applicability in the scenarios which we focus on in this work. However, to put our proposed methods into perspective, we compare their performance with these neural models in terms of point estimates. Specifically, we use the mean of the distributions for both GP and RF as predictions and use standard regression metrics (RMSE and Pearson's r correlation) to assess the performance of our models against BI-LSTM.

	consistent		*inconsistent*	
	RMSE	r	RMSE	r
BI-LSTM^p	**0.795**†	**0.897**†	0.951	0.522
BILSTM^r	0.810	0.891	0.936	**0.548**†
RF	0.805	0.892	0.942	0.527
GP_{RBF}	0.822	0.887	**0.932**	0.545

Table 5.3 Point estimate comparison results.

For the GP model, we restrict our evaluation in this section (and in the remainder of this chapter) to the one with an RBF kernel, as it performed significantly better in Section 5.5.1. We compare with two BI-LSTM models: (1) with pre-trained word embeddings, using GloVe (Pennington et al., 2014) ("BI-LSTM^p" hereafter); and with randomly initialised word embeddings ("BILSTM^r" hereafter). See Chapter 4.1.4.1 for a detailed description of all hyperparameters.

5.5.2.2 Results

From Table 5.3, we see that while BI-LSTM^p outperform our methods in the *consistent* set, the difference is small and we obtain good results nevertheless. In particular, correlation is close to 0.9 for all methods. Therefore, we can see that GPs and RFs obtain comparable results with the state-of-the-art while providing additional information through the predictive distributions.

The importance of having distributions as predictions becomes clear when we see the results for the *inconsistent* set, in Table 5.3. Here, not only do GPs perform on par with the BI-LSTM models, but the overall correlation is much lower (between 0.52 and 0.55). This highlights the harder task of predicting quality labels under disagreement, which further motivates the additional uncertainty information coming from predictive distributions.

5.5.3 Prediction of Secondary Labels

As explained in Section 5.3.2, the *inconsistent* articles are ones where the *primary* label is in disagreement with the *secondary* ones, from different Wikipedia Projects. In this

section, we assess how our models fare under a transfer scenario, where the goal is to predict these secondary labels. Such a scenario can be useful, for instance, if we want to incorporate information from Projects to decide on the quality of a document.

5.5.3.1 Evaluation Metrics

To measure the performance with *secondary* labels as references, one option is to aggregate the labels of an article into a single one (through voting or averaging, for instance) and use that value as the reference. Instead, we opt to embrace the disagreement, and propose a *weighted* extension of NLPD, namely wNLPD, which we define as

$$\text{wNLPD} = -\frac{1}{n}\sum_{i=1}^{n}\log\sum_{j=1}^{m_i} w_j\, p(\hat{y}_i = y_j|\boldsymbol{x_i}),$$

where m_i is the number of *secondary* labels for article $\boldsymbol{x}_i$, and w_j is the weight for label j. If we have prior information about the reliability of some label sources (for instance, different Wikipedia Projects), one can plug this information into the weights. Here we assume equal reliability and use uniform weights $w_j = \text{freq}(j)/m_i$, where freq is the count of label j among *secondary* labels. The metric degrades to standard NLPD when labels are consistent. We also evaluate point estimate performance using a similar weighting scheme for RMSE (which we denote as wRMSE).

5.5.3.2 Results

Table 5.4 summarises the results. Notice that in this setting we only report results for the *inconsistent* test set, as the *consistent* one has no disagreement (and therefore, numbers would match the ones in Section 5.5.1). Here we also see that GPs achieve significantly better performance than RFs, although by a much lower margin compared to the results on the *primary* labels (Table 6.3). This reflects the harder aspect of this setting. We hypothesise we can obtain better performance in this scenario by incorporating the *secondary* labels at training time, which we leave for future work.

	wNLPD	wRMSE
RF	1.460	1.158
GP_{RBF}	**1.412**†	**1.152**†

Table 5.4 Performance comparisons between RF and GP with the RBF kernel in predicting the secondary quality labels.

5.5.4 Case Study: Quality Prediction as Filtering for Manual Revision

As mentioned in Section 5.1, one use for a quality prediction system is to filter documents for manual revision. In the case of Wikipedia, such revisions are mandatory for articles to be assigned as a Good Article or a Featured Article. This incurs in an *asymmetric risk*: the cost of mispredicting an article as GA and FA is higher than other labels, as these trigger expensive, manual labour. Such a scenario can be modelled through *asymmetric loss functions* (Varian, 1975).

If a quality model provides predictive distributions, one can obtain *optimal* quality decisions under an asymmetric loss function through the framework of Minimum Bayes Risk (MBR). This setting has been studied before by Christoffersen and Diebold (1997) and more recently applied by Beck et al. (2016) in the context of machine translation post-editing. However, these assume a regression scenario. While we employ regression models in our work for ease of modelling, the final decisions in the pipeline are discrete (although still ordinal).

To adapt the MBR framework into our case, we first define the risk $\delta(q)$ of predicting the quality label q as

$$\delta(q) = \sum_{\hat{q}} \mathscr{L}(\hat{q}, q)\, p(\hat{q}|\boldsymbol{x}),$$

where $\mathscr{L}(\hat{q}, q)$ is an (asymmetric) loss function over the discrete quality labels and $p(\hat{q}|\boldsymbol{x})$ is the *discretised* probability of quality label $\hat{q}$ for document $\boldsymbol{x}$ under one of our proposed models. We detail these two terms below.

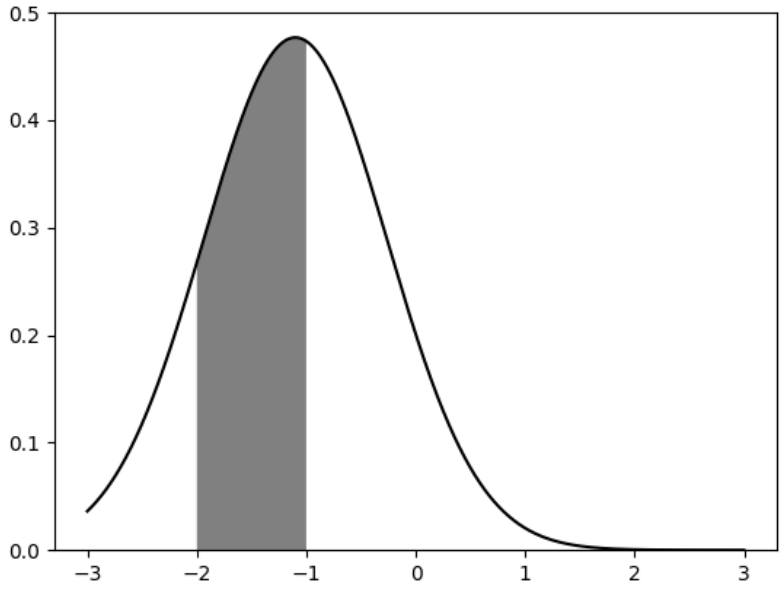


Figure 5.4 Discretisation of a continuous predictive distribution. The shaded area shows the probability of quality label Start (−1.5).

5.5.4.1 Discretized Distribution

Given a predictive distribution obtained by a regression model we can discretize it by using the cumulative density function (cdf). Define $\ell(\hat{q})$ as the real value which we encode quality label $\hat{q}$, as described in Section 5.4. With this, we obtain the discretized probability mass function

$$p(\hat{q}|\boldsymbol{x}) = \begin{cases} 1 - \text{cdf}(\ell(\hat{q}) - 0.5) & \text{if } \hat{q} = \mathsf{FA} \\ \text{cdf}(\ell(\hat{q}) + 0.5) & \text{if } \hat{q} = \mathsf{Stub} \\ \text{cdf}(\ell(\hat{q}) + 0.5) - \text{cdf}(\ell(\hat{q}) - 0.5) & \text{otherwise,} \end{cases}$$

where the cdf is obtained from the predictive distribution. As we only consider Gaussian distributions for predictions, we can easily use off-the-shelf implementations to obtain the cdf. Figure 5.4 gives an example of how to obtain the probability of an instance being predicted to be Start (−1.5).

5.5.4.2 Asymmetric Loss Function

To incorporate asymmetry into the quality label prediction, we define it as

$$\mathscr{L}(\hat{q}, q) = \begin{cases} 0 & \text{if } \hat{q} = q \\ \alpha & \text{if } \hat{q} \neq q, \hat{q} \notin S, q \in S \\ 1 & \text{otherwise} \end{cases},$$

where S is a high-risk label set and $\alpha > 1$ is the penalty associated with a higher risk prediction. In our scenario, we set $S = \{\mathsf{GA}, \mathsf{FA}\}$, as these are the labels we want to give larger penalties when there is a misprediction. Notice that this loss is only an example tailored to our specific setting: other scenarios might warrant different definitions.

5.5.4.3 Evaluation

Under a deployment scenario, one would evaluate $\delta(q)$ for all possible 6 labels and choose the one with minimum risk. However, since in our case we have access to true test set labels, we can just average all $\delta(q)$ for the *gold* labels q in order to assess the models we are interested in (GPs and RFs). As in the previous sections, we report average results over 10 runs for both *consistent* and *inconsistent* test sets.

5.5.4.4 Results

In Figure 5.5 we plot the average risk while varying the penalty cost α. As in the intrinsic evaluation result, we see that GPs tend to perform better in the *inconsistent* set. On the other hand, the results in the *consistent* set are very similar, and no conclusions can be made about which method performs best. Overall, the results are favourable towards GPs but the inconclusive results for *consistent* labels shows that there is room for improvements in uncertainty modelling, which we leave for future work.

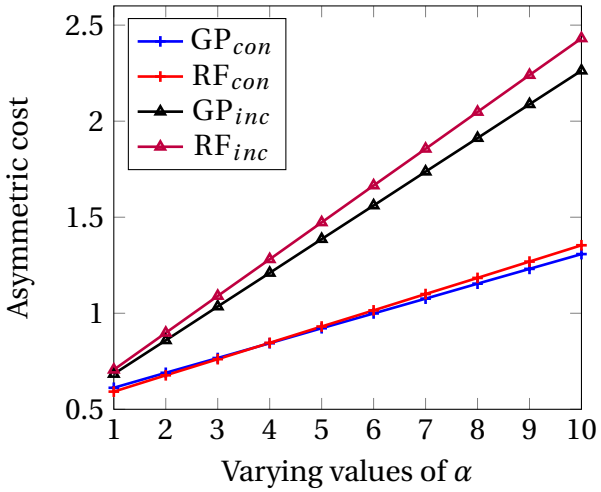


Figure 5.5 Asymmetric risk with varying values of α (best viewed in colour). Here, GP$_{con}$ and RF$_{con}$ denote risk values achieved by GP and RF over *consistent* articles, respectively; GP$_{inc}$ and RF$_{inc}$ denote risk values achieved by GP and RF over *inconsistent* articles.

5.6 Applicability to Other Domains

Both GPs and RFs, along with the evaluation metrics proposed in this work, can be used to measure the uncertainty in document quality assessment, where a given document can receive different quality scores from different humans, such as student essays and academic papers. For student essays, there is usually more than one examiner involved in scoring a given essay, where examiners can agree/disagree over the quality of essays. For academic papers, there are multiple reviewers involved in reviewing a given paper, hence resulting in multiple scores with regards to each aspect assigned to the same paper. For questions and answers in cQA, we can not measure the uncertainty of quality assessment systems unless different scores can be obtained from users, while the voting mechanism in cQA limits to a single quality score for each answer.

5.7 Summary

In this chapter, we have addressed the third research question of how we can take the inherent subjectivity of document quality assessment into consideration in evaluating

the performance of quality assessment systems. Specifically, we use GPs and RFs, as both of them can provide predictive distributions, enabling us to measure the uncertainty of Wikipedia article quality assessment. Utilising the NLPD evaluation metric, we find that both GPs and RFs are less certain over *inconsistent* articles, where people tend to disagree over their quality, and GPs are more conservative in their predictions over such articles. We also propose an asymmetric cost, which takes the prediction uncertainty into consideration, imitating a real world scenario where decision-making processes based on model predictions can lead to different costs. Experimental results show that GPs are a better option if overestimates are heavily penalised.

In the next chapter, we turn our attention to detecting whether there are intruder sentences in documents and identifying the span of any such intruder sentences, given that they interrupt the coherence of documents, thereby lowering the quality of documents.

Chapter 6

Evaluating Document Coherence Modelling

The focus of Chapter 3 to Chapter 5 was on the overall quality of a document. In this chapter, we shift our focus to a particular aspect of document quality – document coherence. We address the fourth research question – can a quality assessment system detect whether there are intruder sentences in documents and identify the span of any such intruder sentences, given that they interrupt the coherence of documents, thereby lowering their quality. Detecting such intruder/incoherent sentences can potentially improve document quality. While pre-trained language models ("LMs") have driven impressive gains over morpho-syntactic and semantic tasks, their ability to model discourse and pragmatic phenomena is less clear. As a step towards a better understanding of their discourse modelling capabilities, we propose a sentence intrusion detection task. We formulate this task as a sentence-level classification problem, in predicting whether there is an "intruder" sentence in a document, and if so, which single sentence it is. Lacking a dataset for this task, we introduce INSteD, a novel **in**coherent **s**en**te**nce **d**etection dataset that contains 170,000+ documents constructed from English Wikipedia and CNN News. Our experiments show that pre-trained LMs perform impressively in in-domain evaluation, but experience a substantial drop in the cross-domain setting, indicating limited general-

isation capacity. Further results over a novel linguistic probe dataset show that there is substantial room for improvement, especially in the cross-domain setting.

We first describe the motivation for proposing the task of sentence intrusion detection in Section 6.1, followed by related work in Section 6.2. Next, we present how our INSteD dataset is constructed in Section 6.3 and dataset analysis in Section 6.4. Models and experimental results are described in Section 6.5 and Section 6.6. Lastly, we present linguistic probe dataset construction and experimental results in Section 6.7, before finishing this chapter.

This chapter is based on and extends the following paper:

> Aili Shen, Meladel Mistica, Bahar Salehi, Hang Li, Timothy Baldwin and Jianzhong Qi. 2020. Evaluating Document Coherence Modelling. In *Transactions of the Association for Computational Linguistics (TACL)* (under review).

6.1 Motivation

Rhetorical relations refer to the transition of one sentence to the next in a span of text Mann and Thompson (1988), Asher and Lascarides (2003). They are important as a discourse device that contributes to the overall coherence, understanding, and flow of the text. These relations span a tremendous breadth of types, including *contrast*, *elaboration*, *narration*, and *justification*. These connections allow us to communicate cooperatively in understanding one another Grice (2002), Wilson and Sperber (2004). The ability to understand such coherence (and conversely detect incoherence) is potentially beneficial for downstream tasks, such as storytelling Fan et al. (2019), Hu et al. (2020b), recipe generation Chandu et al. (2019), document-level text generation Park and Kim (2015), Holtzman et al. (2018), and essay scoring Tay et al. (2018), Li et al. (2018).

However, there is little work on document coherence understanding, especially examining the capacity of pre-trained LMs to model the coherence of longer documents. To address this gap, we examine the capacity of pre-trained language models to capture document coherence, focused around two research questions: (1) do models truly capture the

(1) Mark Ferguson (born 21 May 1990) is an Irish handballer, currently playing in Dublin, Ireland.
(2) **It is a twelve time Asian Champion, the tournament has been won by any other nation only twice.**
(3) Previously playing for his university team ITB, in the 2013/14 League his club Lughnasa HC came 3rd ...
(4) Mark has been involved in the National Team from 2011 and played in Ireland's first ever European qualifiers ...
(5) The following year Ireland took part in the 2016 Men's European Championship Qualification ...

Figure 6.1 An excerpt of an incoherent document, with the "intruder" sentence indicated in **bold**.

intrinsic properties of document coherence? and (2) what types of document incoherence can/can't these models detect?

We propose the sentence intrusion detection task: (1) to determine whether a document contains an intruder sentence (coarse-grained level); and (2) to identify the span of any intruder sentence (fine-grained level). We restrict the scope of the intruder text to a single sentence, noting that in practice, the incoherent text could span multiple sentences, or alternatively be sub-sentential.

Existing datasets in document coherence measurement Chen et al. (2019), Clercq et al. (2014), Lai and Tetreault (2018), Mim et al. (2019), Pitler and Nenkova (2008), Nguyen and Joty (2017) are unsuitable for our task: they are either too small for training large-scale models, or do not specify the span of incoherent text. For example, in the dataset of Lai and Tetreault (2018), each document is assigned a coherence score, but the span of incoherent text is not specified. There is thus a need for a large-scale dataset which includes annotation of the position of intruder text.

In this work, we introduce a dataset consisting of English documents from two domains: Wikipedia articles (106K) and CNN news articles (72K). This dataset fills a gap in research pertaining to document coherence: our dataset is large in scale, includes both coherent and incoherent documents, and has mark-up of the position of any intruder sentence. Figure 6.1 is an example document with an intruder sentence. Here, the high-

lighted sentence reads as though it should be an elaboration of the previous sentence, but clearly exhibits an abrupt change of topic and the pronoun *it* cannot be readily resolved.

This paper makes the following contributions: (1) we propose the sentence intrusion detection task, and examine how pre-trained LMs perform over the task and hence at document coherence understanding; (2) we construct a large-scale dataset from two domains — Wikipedia and CNN news articles — that consists of coherent and incoherent documents, and is accompanied with the positions of intruder sentences, to evaluate in both in-domain and cross-domain settings; (3) we examine the behaviour of models and humans, to better understand the ability of models to model the intrinsic properties of document coherence; and (4) we further hand-craft adversarial test instances across a variety of linguistic phenomena to better understand the types of incoherence that a given model can detect.

6.2 Related Work

We first review tasks relevant to our proposed task, then describe existing datasets used in coherence measurement, and finally discuss work on dataset artefacts and linguistic probes.

6.2.1 Document Coherence Measurement

Coherence measurement has been studied across various tasks, such as sentence reordering detection Barzilay and Lapata (2005), Elsner et al. (2007), Barzilay and Lapata (2008), Elsner and Charniak (2011), Li and Jurafsky (2017), Putra and Tokunaga (2017), sentence insertion Elsner and Charniak (2011), Putra and Tokunaga (2017), Xu et al. (2019a), paragraph reconstruction Lapata (2003), Elsner et al. (2007), Li and Jurafsky (2017), Xu et al. (2019a), Prabhumoye et al. (2020), summary coherence rating Barzilay and Lapata (2005), Pitler et al. (2010), Guinaudeau and Strube (2013), Nguyen and Joty (2017), readability assessment Guinaudeau and Strube (2013), Mesgar and Strube (2016, 2018), and essay scoring Mesgar and Strube (2018), Somasundaran et al. (2014), Tay et al. (2018). These

tasks differ from our task of intruder sentence detection as follows. First, sentence reordering detection assigns coherence scores to a document and its sentence-permuted versions, where the original document is considered to be well-written and coherent and permuted versions incoherent. Incoherence is introduced by shuffling sentences, while our intruder sentences are selected from a second document, and there is only ever a single intruder sentence per document. Second, sentence insertion aims to find the correct position to insert a removed sentence back into a document. Paragraph reconstruction aims to recover the original sentence order of a shuffled paragraph given its first sentence. These two tasks do not consider sentences from outside of the document of interest. Third, in summary coherence rating, readability assessment, and essay scoring, coherence is just one dimension of the overall document quality measurement.

Various methods have been proposed to capture local and global coherence, while our work aims to examine the performance of existing pre-trained LMs in document coherence understanding. To assess local coherence, traditional studies have used entity matrices, e.g. to represent entity transitions across sentences (Barzilay and Lapata, 2005, 2008). Guinaudeau and Strube (2013) and Mesgar and Strube (2016) use a graph to model entity transition sequences. Sentences in a document are represented by nodes in the graph, and two nodes are connected if they share the same or similar entities. Neural models have also been proposed Ji and Smith (2017), Li and Jurafsky (2017), Li et al. (2018), Mesgar and Strube (2018), Mim et al. (2019), Nguyen and Joty (2017). For example, Tay et al. (2018) capture local coherence by computing the similarity of the output of two LSTMs (Hochreiter and Schmidhuber, 1997), which they concatenate with essay representations to score essays. Li et al. (2018) use multi-headed self-attention to capture long distance relationships between words, which are passed to an LSTM layer to estimate essay coherence scores. Xu et al. (2019a) use the average of local coherence scores between consecutive pairs of sentences as the document coherence score.

Another relevant task is disfluency detection in spontaneous speech transcription (Johnson and Charniak, 2004, Lou et al., 2018). This task detects the reparandum and repair in spontaneous speech transcriptions to make the text fluent by replacing the

reparandum with the repair. Also relevant is language identification in code-switched text Adouane et al. (2018a,b), Mave et al. (2018), Yirmibesoglu and Eryigit (2018), where disfluency is defined at the language level (for a monolingual speaker, e.g.). Lau et al. (2015) and Warstadt et al. (2019) predict sentence-level acceptability (how natural a sentence is). However, none of tasks are designed to measure document coherence, although sentence-level phenomena can certainly impact on document coherence.

6.2.2 Document Coherence Datasets

There exist a number of datasets targeted at discourse understanding. For example, Alikhani et al. (2019) construct a multi-modal dataset for understanding discourse relations between text and imagery, such as elaboration and exemplification. In contrast, we focus on discourse relations in a document at the inter-sentential level. The Penn Discourse Treebank Miltsakaki and Kukich (2004), Prasad et al. (2008) is a corpus of coherent documents with annotations of discourse connectives and their arguments, noting that inter-sentential discourse relations are not always lexically marked Webber (2009).

The most relevant work to ours is the discourse coherence dataset of Chen et al. (2019), which was proposed to evaluate the capabilities of pre-trained LMs in capturing discourse context. This dataset contains documents (18K Wikipedia articles and 10K documents from the Ubuntu IRC channel) with fixed sentence length, and labels documents only in terms of whether they are incoherent, without considering the position of the incoherent sentence. In contrast, our dataset: (1) provides more fine-grained information (i.e. the sentence position); (2) is larger in scale (over 170K documents); (3) contains documents of varying length; (4) incorporates adversarial filtering to reduce dataset artefacts (see Section 6.3); and (5) is accompanied with human annotation over the Wikipedia subset, allowing us to understand behaviour patterns of machines and humans.

6.2.3 Dataset Artefacts

Also relevant to this research is work on removing artefacts in datasets Zellers et al. (2019), McCoy et al. (2019), Zellers et al. (2018). For example, based on analysis of the SWAG dataset Zellers et al. (2018), Zellers et al. (2019) find artefacts such as stylistic biases, which correlate with the document labelling and mean that naive models are able to achieve abnormally high results. Similarly, McCoy et al. (2019) examine artefacts in an NLI dataset, and find that naive heuristics which are not directly related to the task can perform remarkably well. We incorporate the findings of such work in the construction of our dataset.

6.2.4 Linguistic Probes

Adversarial training has been used to craft adversarial examples to obtain more robust models, either by manipulating model parameters (white-box attacks) or minimally editing text at the character/word/phrase level (black-box attacks). For example, Papernot et al. (2018) provide a reference library of adversarial example construction techniques and adversarial training methods.

As we aim to understand the linguistic properties that each model has captured, we focus on black-box attacks Sato et al. (2018), Cheng et al. (2018), Liang et al. (2018), Yang et al. (2020), Samanta and Mehta (2017). For example, Samanta and Mehta (2017) construct adversarial examples for sentiment classification and gender detection by deleting, replacing, or inserting words in the text. For a comprehensive review of such studies, see Belinkov and Glass (2019).

There is also a rich literature on exploring what kinds of linguistic phenomena a model has learned Hu et al. (2020a), Hewitt and Liang (2019), Hewitt and Manning (2019), Chen et al. (2019), McCoy et al. (2019), Conneau et al. (2018), Gulordava et al. (2018), Peters et al. (2018), Tang et al. (2018), Blevins et al. (2018), Wilcox et al. (2018), Kuncoro et al. (2018), Tran et al. (2018), Belinkov et al. (2017). The basic idea is to use learned representations to predict linguistic properties of interest. Example linguistic properties are

subject–verb agreement or syntactic structure, while representations can be word or sentence embeddings. For example, Marvin and Linzen (2018) construct minimal sentence pairs, consisting of a grammatical and ungrammtical sentence, to explore the capacity of LMs in capturing phenomena such as subject–verb agreement, reflexive anaphora, and negative polarity items. In our work, we hand-construct intruder sentences which result in incoherent documents, based on a broad range of linguistic phenomena.

6.3 Dataset Construction

6.3.1 Dataset Desiderata

To construct a large-scale, low-noise dataset that truly tests the ability of systems to detect intruder sentences, we posit five desiderata:

1. **Multiple sources:** The dataset should not be too homogeneous in terms of genre or domain, and should ideally test the ability of models to generalise across domain.

2. **Defences against hacking:** Human annotators and machines should not be able to hack the task and reverse-engineer the labels by sourcing the original documents.

3. **Free of artefacts:** The dataset should be free of artefacts, that allow naive heuristics to perform well.

4. **Topic consistency:** The intruder sentence, which is used to replace a sentence from a coherent document to obtain an incoherent document, should be relevant to the topic of the document, to focus the task on coherence and not simple topic detection.

5. **KB-free:** Our goal is *NOT* to construct a fact-checking dataset; the intruder sentence should be determinable based on the content of the document, without reliance on external knowledge bases or fact-checking.

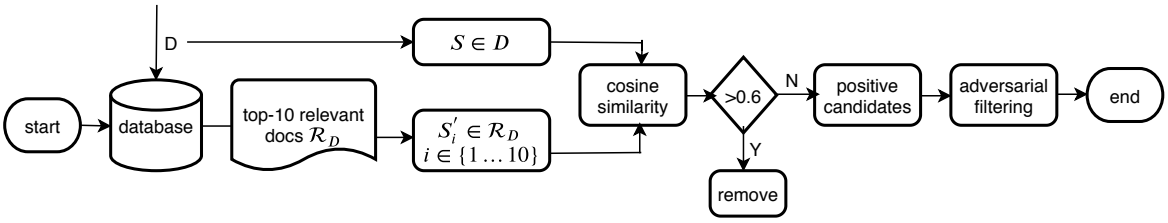


Figure 6.2 Illustration of generating candidate incoherent documents.

6.3.2 Data Sources

We construct a dataset from two sources — Wikipedia and CNN — which differ in style and genre, satisfying the first desideratum. Similar to WikiQA Yang et al. (2015) and HotpotQA Yang et al. (2018), we represent a Wikipedia document by its summary section (i.e., the opening paragraph), constraining the length to be between 3 and 8 sentences. For CNN, we adopt the dataset of Hermann et al. (2015) and Nallapati et al. (2016), which consists of over 100,000 news articles. To obtain documents with sentence length similar to those from Wikipedia, we randomly select the first 3–8 sentences from each article.

To defend against dataset hacks[1] which could expose the labels of the test data (desideratum 2), the Wikipedia test set is randomly sampled from 37 historical dumps of Wikipedia, where the selected article has a cosine similarity less than the historical average of 0.72 with its online version.[2] For the training set, we remove this requirement and randomly select articles from different Wikipedia dumps, i.e., the articles in the training set might be the same as their current online version. For CNN, we impose no such limitations.

6.3.3 Generating Candidate Positive Samples

We consider the original documents to be coherent. We construct incoherent documents from half of our sampled documents as follows (satisfying desiderata 3–5):

[1] Deliberate or otherwise, e.g. via pre-training on the same version of Wikipedia our dataset was constructed over.

[2] This threshold was determined by calculating the average TF-IDF-weighted similarity of the summary section for documents in all 37 dumps with their current online versions.

1. Given a document D, use bigram hashing and TF-IDF matching Chen et al. (2017) to retrieve the top-10 most similar documents from a collection of documents from the same domain, where D is the query text. Let the set of retrieved documents be $\mathscr{R}_D$.

2. Randomly choose a non-opening sentence S from document D, to be replaced by a sentence candidate generated later. We do not replace the opening sentence as it is needed to establish document context.

3. For each document $D' \in \mathscr{R}_D$, randomly select one non-opening sentence $S' \in D'$ as an intruder sentence candidate.

4. Calculate the TF-IDF-weighted cosine similarity between sentence S and each candidate S'. Remove any candidates with similarity scores ≥ 0.6, to attempt to generate a KB-free incoherence.

5. Replace sentence S with each low-similarity candidate S', and use a fine-tuned XLNET-LARGE model (Yang et al., 2019c) to check whether it is easy for XLNET-LARGE to detect (see Section 6.5). For documents with both easy and difficult sentence candidates, we randomly sample from the difficult sentence candidates; otherwise, we randomly choose from all the sentence candidates.

The decision to filter out sentence candidates with similarity ≥ 0.6 was based on the observation that more similar sentences often led to the need for world knowledge to identify the intruder sentence (violating the fifth desideratum). For example, given *It is the **second** novel in the first of three trilogies about Bernard Samson, ...*, a candidate intruder sentence candidate with high similarity is *It is the **first** novel in the first of three trilogies about Bernard Samson ....*

We also trialled other ways of generating incoherent samples, such as using sentence S from document D as the query text to retrieve documents, and adopting a 2-hop process to retrieve relevant documents. We found that these methods resulted in documents which can be identified by the pre-trained models easily.

Source	#docs	avg. #sents	avg. #tokens
Wikipedia	106,352 (46%)	5±1	126±24
CNN News	72,670 (49%)	5±1	134±32

Table 6.1 Dataset statistics for INSteD. Numbers in parentheses are percentages of incoherent documents.

6.4 Dataset Analysis

6.4.1 Statistics of the Dataset

The process described in Section 6.3 resulted in 106,352 Wikipedia documents and 72,670 CNN documents, at an average sentence length of 5 in both cases (see Table 6.1). The percentages of positive samples (46% and 49%, respectively) are slightly less than 50% due to our data generation constraints (detailed in Section 6.3.3), which can lead to no candidate intruder sentence S' being generated for original sentence S. We set aside 8% of Wikipedia (which we manually tag, as detailed in Section 6.4.5) and 20% of CNN for testing.

6.4.2 Types of Incoherence

To better understand the different types of issues resulting from our automatic method, we sampled 100 (synthesised) incoherent documents from Wikipedia and manually classified the causes of incoherence according to three overlapping categories (ranked in terms of expected ease of detection): (1) information structure inconsistency (a break in information flow); (2) logical inconsistency (a logically inconsistent world state is generated, such as someone attending school before they were born); and (3) factual inconsistency (where the intruder sentence is factually incorrect). See Table 6.2 for a breakdown across the categories, noting that a single document can be incoherent across multiple categories. Information structure inconsistency is the most common form of incoherence, followed by factual inconsistency. The 35% of documents with factual inconsistencies break down into 8% (overall) that have other types of incoherence, and 27% that only have a factual

Incoherence	Example	%
Information structure inconsistency	*He is currently the senior pastor at Sovereign Grace Church of Louisville.* ***The Church is led by Senior Pastor Ray Johnston, Senior Pastor Curt Harlow and Senior Pastor Andrew McCourt, and Senior Pastor Lincoln Brewster.*** *Under Mahaney's leadership, Sovereign Grace Church of Louisville is a member of Sovereign Grace Churches.*	58
Logical inconsistency	*Michael David, born September 22, 1954, is an American-born American painter.* ***From 1947–1949 he attended the Otis Art Institute, from 1947 to 1950 he also attended the Art Center College of Design in Los Angeles, and in 1950 the Chouinard Art Institute.***	26
Factual inconsistency	*The Newport Tower has 37 floors.* ***It is located on the beachfront on the east side of Collins Avenue between 68th and 69th Streets.*** *The building was developed by Melvin Simon & Associates in 1990.*	35

Table 6.2 Types of document incoherence in Wikipedia. Text in bold indicates the intruder sentence.

inconsistency. This is an issue for the fifth desideratum for our dataset (see Section 6.3.1), motivating the need for manual checking of the dataset to determine how readily the intruder sentence can be detected.[3]

6.4.3 Evaluation Metrics

We base evaluation of intruder sentence detection at both the document and sentence levels:

1. **document level**: *Does the document contain an intruder sentence?* This is measured based on classification accuracy (Acc), noting that the dataset is relatively balanced at the document level (see Table 6.1). A prediction is "correct" if at least one sentence/none of the sentences is predicted to be an intruder.

2. **sentence level**: *Is a given (non-opening) sentence an intruder sentence?* This is measured based on F_1, noting that most (roughly 88%) sentences are non-intruder sentences.

[3]We keep these documents in the dataset, as it is beyond the scope of this work to filter these documents out.

6.4.4 Testing for Dataset Artefacts

To test for artefacts, we use XLNET-LARGE Yang et al. (2019c) to predict whether each non-opening sentence is an intruder sentence, *in complete isolation of its containing document* (i.e., as a standalone sentence classification task). We compare the performance of XLNET-LARGE with a majority-class baseline ("MAJORITY") which predicts all sentences to be non-intruder sentences (i.e., from the original document), where XLNET-LARGE is fine-tuned over the Wikipedia/CNN training set, and tested over the corresponding test set.

For Wikipedia, XLNET-LARGE obtains an Acc of 55.4% (vs. 55.1% for MAJORITY) and F_1 of 3.4% (vs. 0.0% for MAJORITY). For CNN, the results are 50.8% and 1.2%, respectively (vs. 51.0% and 0.0% resp. for MAJORITY). These results suggest that the dataset does not contain obvious artefacts, at least for XLNET-LARGE. We also experiment with a TF-IDF weighted bag-of-words logistic regression model, achieving slightly worse results than XLNET-LARGE (Acc = 55.1%, F_1 = 0.05% for Wikipedia, and Acc = 50.6%, F_1 = 0.3% for CNN).[4]

6.4.5 Human Verification

We performed crowdsourcing via Amazon Mechanical Turk over the Wikipedia test data to examine how humans perform over this task. Each Human Intelligence Task (HIT) contained 5 documents and was assigned to 5 workers. For each document, the task was to identify a single sentence which "creates an incoherence or break in the content flow", or in the case of no such sentence, "None of the above", indicating a coherent document. In the task instructions, workers were informed that there is at most one intruder sentence per document, and were not able to select the opening sentence. Among the 5 documents for each HIT, there was one incoherent document from the training set, which was pre-identified as being easily detectable by an author of the paper, and acts as a quality control item. We include documents where at least 3 humans assign the same label as our test

[4]For RoBERTa-Large (Section 6.6.1), there were also no obvious artefacts observed in the standalone sentence setting: Acc = 55.7% and F_1 = 5.3% over Wikipedia, and Acc = 51.3% and F_1 = 4.3% over CNN.

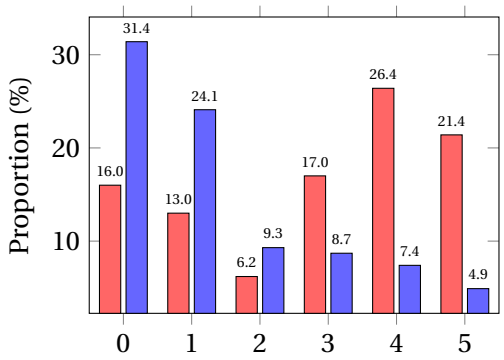


Figure 6.3 Distribution of instances where different numbers of humans produce correct answers. Note that the red bars indicate distributions over all documents and the blue bars indicate distributions over incoherent documents.

dataset (90.3% of the Wikipedia test dataset), where all the results are reported over these documents, if not specified.[5] Payment was calibrated to be above minimum wage in the region the research was carried out.

Figure 6.3 shows the distribution of instances where different numbers of workers produced the correct answer (the red bar). For example, for 6.2% of instances, 2/5 workers annotated correctly. The blue bars indicate the proportion of incoherent documents where the intruder sentence was correctly detected by the given number of annotators (e.g., for 9.3% of incoherent documents, only 2/5 workers were able to identify the intruder sentence correctly). Humans tend to agree with each other over coherent documents, as indicated by the increasing percentages for red bars but decreasing percentages for blue bars across the x-axis. Intruder sentences in incoherent documents, however, are harder to detect. One possible explanation is that the identification of intruder sentences requires fact-checking, which workers were instructed not to do (and base their judgement only on the information in the provided document); another reason is that intruder sentences disrupt local incoherence with neighbouring sentences, creating confusion as to which is the intruder sentence (with many of the sentence-level mis-annotations being off-by-one errors).

[5]Different people may have different thresholds in considering a document to be incoherent, but this is beyond the scope of our work.

6.5 Models

We model intruder sentence detection as a binary classification task: each non-opening sentence in a document is concatenated with the document, and a model is asked to predict whether the sentence is an intruder sentence to the document.

Our focus is on the task, dataset, and how existing models perform at document coherence prediction rather than modelling novelty, and we thus experiment with pre-existing pre-trained models. The models are as follows, each of which is fed into an MLP layer with a softmax output.

BoW: Average the word embeddings for the combined document (sentence + sequence of sentences in the document), based on pre-trained 300D GloVe embeddings trained on a 840B-token corpus Pennington et al. (2014).

Bi-LSTM: Feed the sequence of words in the combined document into a single-layer 512D Bi-LSTM with average-pooling; word embeddings are initialised as with BoW.

InferSent: Generate representations for the sentence and document with InferSent Conneau et al. (2017), and concatenate the two; InferSent is based on a Bi-LSTM with a max-pooling layer, trained on SNLI Bowman et al. (2015).

Skip-Thought: Generate representations for the sentence and document with Skip-Thought Kiros et al. (2015), and concatenate the two; Skip-Thought is an encoder–decoder model where the encoder extracts generic sentence embeddings and the decoder reconstructs surrounding sentences of the encoded sentence.

BERT: Generate representations for the concatenated sentence and document with BERT Devlin et al. (2019), which was pre-trained on the tasks of masked language modelling and next sentence prediction over Wikipedia and BooksCorpus Zhu et al. (2015); we experiment with both BERT-Large and BERT-Base (the cased versions).

RoBERTa: Generate representations for the concatenated sentence and document with RoBERTa Liu et al. (2019), which was pre-trained on the tasks of masked language modelling (dynamically masking) and full sentences (continuous sentences from the same document) over Cc-news, OpenWebTextCorpus, and STORIES Trinh and Le (2018), in addition to the same data BERT was pre-trained on; we experiment with both RoBERTa-Large and RoBERTa-Base.

ALBERT: Generate representations for the concatenated sentence and document with ALBERT Lan et al. (2020), which was pre-trained over the same dataset as BERT but replaces the next sentence prediction objective with a sentence-order prediction objective, to model document coherence; we experiment with both ALBERT-Large and ALBERT-xxLarge.

XLNET: Generate representations for the concatenated sentence and document with XLNET Yang et al. (2019c), which was pre-trained using a permutation language modelling objective over datasets including Wikipedia, BooksCorpus, Giga5 Parker et al. (2011), ClueWeb 2012-B Callan et al. (2009), and Common Crawl; we experiment with both XLNET-LARGE and XLNET-BASE (the cased versions). Although XLNET-LARGE is used in removing data artefacts when selecting the intruder sentences, our experiments suggest that the comparative results across models (with or without artefact filtering) are robust.

6.6 Experiments

6.6.1 Preliminary Results

In our first experiments, we train the various models across both Wikipedia and CNN, and evaluate them in-domain and cross-domain. We are particularly interested in the cross-domain setting, to test the true ability of the model to detect document incoherence, as distinct from overfitting to domain-specific idiosyncrasies. It is also worth mentioning that BERT, RoBERTa, ALBERT, and XLNET are pre-trained on multi-sentence

	→Wikipedia				→CNN			
	Wiki →Wiki		CNN →Wiki		CNN →CNN		Wiki →CNN	
	Acc (%)	F_1 (%)	Acc (%)	F_1 (%)	Acc (%)	F_1 (%)	Acc (%)	F_1 (%)
MAJORITY	57.3	0.0	57.3	0.0	50.6	0.0	50.6	0.0
BOW	57.3	0.0	57.3	0.0	50.6	0.0	50.6	0.0
BI-LSTM	56.2	12.7	57.3	0.0	51.7	25.1	50.2	3.0
INFERSENT	57.3	0.0	57.3	0.0	50.6	0.0	50.6	0.0
SKIP-THOUGHT	57.3	0.0	57.3	0.0	50.6	0.0	50.6	0.0
BERT-BASE	65.3	35.7	61.2	21.1	80.8	71.6	57.0	23.5
BERT-LARGE	67.0	39.6	64.0	29.1	82.4	74.8	61.5	35.9
XLNET-BASE	67.8	45.0	62.2	22.4	91.2	86.6	64.0	43.3
XLNET-LARGE	72.9	55.4	62.8	22.2	**96.9**	95.0	80.7	73.8
RoBERTa-Base	69.5	47.0	63.2	26.1	92.5	88.8	77.6	68.1
RoBERTa-Large	76.1	59.8	63.7	24.6	96.0	94.5	88.3	83.5
ALBERT-Large	70.7	49.6	63.8	24.9	93.4	90.8	72.6	61.5
ALBERT-xxLarge	**81.7**	**71.5**	**66.6**	**33.2**	**96.9**	**95.9**	**89.1**	**86.7**
Human	66.6	35.9	66.6	35.9	N/A		N/A	

Table 6.3 Experimental results over Wikipedia and CNN, in both in-domain and cross-domain settings.

Wikipedia data, and have potentially memorised sentence pairs, making in-domain experiments problematic for Wikipedia in particular. Also of concern in applying models to the automatically-generated data is that it is entirely possible that an intruder sentence is undetectable to a human, because no incoherence results from the sentence substitution (bearing in mind that only 58% of documents in Table 6.2 contained information structure inconsistencies).

From Table 6.3, we can see that the simpler models (BOW, BI-LSTM, INFERSENT, and SKIP-THOUGHT) perform only at the level of MAJORITY at the document level, for both Wikipedia and CNN. At the sentence level (F_1), once again the models perform largely at the level of MAJORITY (F_1 = 0.0), other than BI-LSTM in-domain for Wikipedia and CNN. In the final row of the table, we also see that humans are much better at detecting whether documents are incoherent (at the document level) than identifying the position of intruder sentences (at the sentence level), and that in general, human performance is low. This is likely the result of the fact that there are only 58% of documents in Table 6.2 containing information structure inconsistencies. We only conducted crowdsourcing over

Wikipedia due to budget limitations and the fact that the CNN documents are available online, making dataset hacks possible.

Among the pre-trained LMs, ALBERT-xxLarge achieves the best performance over Wikipedia and CNN, at both the document and sentence levels. Looking closer at the Wikipedia results, we find that BERT-LARGE achieves a higher precision than XLNET-LARGE (71.0% vs. 60.3%), while XLNET-LARGE achieves a higher recall (51.3% vs. 27.4%). ALBERT-xxLarge achieves a precision higher than BERT-LARGE (79.7%) and a recall higher than XLNET-LARGE (64.9%), leading to the overall best performance. Over CNN, ALBERT-xxLarge, RoBERTa-Large, and XLNET-LARGE achieve high precision and recall (roughly 93.0% to 97%). The competitive results for ALBERT-xxLarge over Wikipedia and CNN result from the pretraining strategies, especially the sentence-order prediction loss capturing document coherence in isolation, different from next sentence prediction loss which conflates topic prediction and coherence prediction in a lower-difficulty single task. The performance gap for ALBERT, RoBERTa, and XLNET between the base and large models are bigger than that of BERT, suggesting that they benefit from greater model capacity.[6]

Looking to the cross-domain results, again, ALBERT-xxLarge achieves the best performance over both Wikipedia and CNN. The lower results for RoBERTa-Large and XLNET-LARGE over Wikipedia may be due to both RoBERTa and XLNET being pre-trained over newswire documents, and fine-tuning over CNN reducing the capacity of the model to generalise. ALBERT and BERT do not suffer from this as they are not pre-trained over newswire documents. The substantial drop between the in- and cross-domain settings for ALBERT, RoBERTa, XLNET, and BERT indicates that the models have limited capacity to learn a generalised representation of document coherence, in addition to the style differences between Wikipedia and CNN.

[6]We also performed experiments where the models were allowed to predict the first sentence as the intruder sentence. As expected, model performance drops, e.g., F_1 of XLNET-LARGE drops from 55.4% to 47.9%, reflecting both the increased complexity of the task and the lack of (at least) one previous sentence to provide document context.

	Wiki→Wiki	Ubuntu→Wiki
MAJORITY	50.0	50.0
ALBERT-xxLarge	96.8	53.1
Human	98.0	98.0
	Ubuntu→Ubuntu	Wiki→Ubuntu
MAJORITY	50.0	50.0
ALBERT-xxLarge	58.1	58.7
Human	74.0	74.0

Table 6.4 Acc for the dataset of Chen et al. (2019).

6.6.2 Results over the Existing Dataset

We also examine how ALBERT-xxLarge performs over the coarse-grained dataset of Chen et al. (2019), where 50 documents from each domain were annotated by a native English speaker. Performance is measured at the document level only, as the dataset does not include indication of which sentence is the intruder sentence. As shown in Table 6.4, ALBERT-xxLarge achieves an Acc of 96.8% over the Wikipedia subset, demonstrating that our Wikipedia dataset is more challenging (Acc of 81.7%) and also underlining the utility of adversarial filtering in dataset construction. Given the considerably lower results, one could conclude that Ubuntu is a good source for a dataset. However, when one of the authors attempted to perform the task manually, they found the document-level task to be extremely difficult as it relied heavily on expert knowledge of Ubuntu packages, much more so than document coherence understanding.

In the cross-domain setting, there is a substantial drop over the Wikipedia dataset, which can be explained by ALBERT-Large failing to generate a representation of document coherence from the Ubuntu dataset, due to the high dependence on domain knowledge as described above, resulting in near-random results. The cross-domain results for ALBERT-xxLarge over Ubuntu are actually marginally higher than the in-domain results but still close to random, suggesting that the in-domain model isn't able to capture either document coherence or domain knowledge, and underlining the relatively minor role of coherence for the Ubuntu dataset.

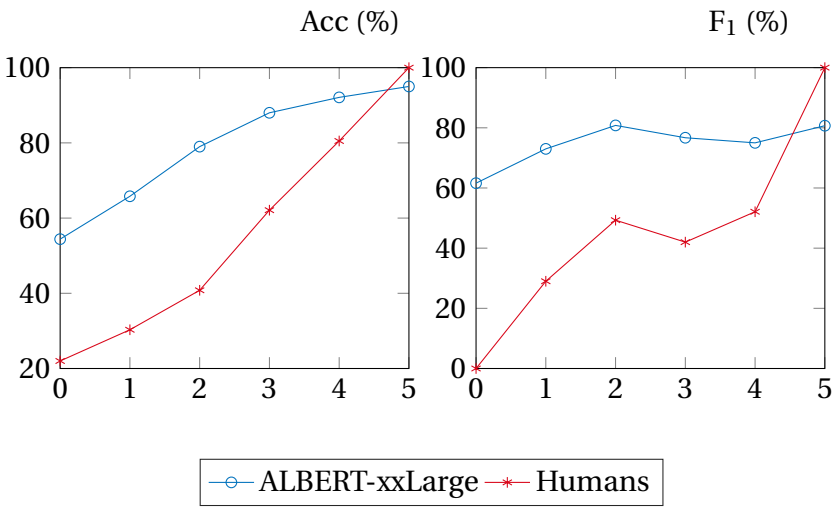


Figure 6.4 ALBERT-xxLarge vs. humans.

6.6.3 Performance on Documents of Different Difficulty Levels

One concern with our preliminary experiments was whether the intruder sentences generate genuine incoherence in the information structure of the documents. We investigate this question by breaking down the results over the best-performing model (ALBERT-xxLarge) based on the level of agreement between the human annotations and the generated gold-standard, for Wikipedia. The results are in Figure 6.4, where the x-axis denotes the number of annotators who agree with the gold-standard: for example, "2" indicates that 2/5 annotators were able to assign the gold-standard labels to the documents.

Our assumption is that the incoherent documents which humans fail to detect are actually not perceptibly incoherent,[7] and that any advantage for the models over humans for documents with low-agreement (with respect to the gold-standard) is actually due to dataset artefacts. At the document level (Acc), there is reasonable correlation between model and human performance (i.e., the model struggles on the same documents as the humans). At the sentence level (F_1), there is less discernible difference in model performance over documents of varying human difficulty.

[7] Although the intruder sentence may lead to factual errors, the annotators were instructed not to do fact checking.

	# coherent docs	# incoherent docs
Correct	1385	177
Wrong	11	404

Table 6.5 Statistics over documents where all 5 humans agree.

	Wiki →Wiki		CNN →Wiki	
	Acc (%)	F_1 (%)	Acc (%)	F_1 (%)
MAJORITY	70.6	0.0	70.6	0.0
BERT-LARGE	76.7	42.0	75.4	36.9
XLNET-LARGE	79.1	57.0	76.6	35.4
RoBERTa-Large	82.0	59.6	77.3	37.4
ALBERT-xxLarge	**85.9**	**68.8**	**78.8**	**42.9**
Human	79.5	45.4	79.5	45.4

Table 6.6 Results over documents annotated consistently by all 5 annotators, where annotations can be the same as or different from gold-standard.

6.6.4 Analysis over Documents with High Human Agreement

To understand the relationship between human-assigned labels and the gold-standard, we further examine documents where all 5 annotators agree, noting that human-assigned labels can potentially be different from the gold-standard here. Table 6.5 shows the statistics of humans over these documents, with regard to the gold-standard. Encouragingly, we can see that humans tend to agree more over coherent documents than incoherent documents. Examining the 11 coherent documents which were annotated as incoherent by all annotators, we find out that there is a break in information flow due to references or urls, even though there is no intruder sentence. For documents with an intruder sentence, where humans disagree with the gold-standard (humans perceive the documents as coherent or the position of the intruder sentence to be other than the actual intruder sentence), we find that 98% of the documents are considered to be coherent. We randomly sampled 100 documents from these documents and examined whether the intruder sentence results in a break in information flow. We find that fact-checking is needed to identify the intruder sentence for 93% of the documents.[8]

[8]Here, the high percentage of incoherent documents with factual inconsistencies does not necessarily point to a high percentage of factual inconsistency in the overall dataset, as humans are more likely to agree with the gold-standard for coherent documents.

Table 6.6 shows the performance over the Wikipedia documents that are annotated consistently by all 5 annotators (from Table 6.5). Consistent with the results from Table 6.3, ALBERT-xxLarge achieves the best performance both in- and cross-domain. To understand the different behaviours of humans and ALBERT-xxLarge, we analyse documents which only humans got correct, only ALBERT-xxLarge got correct, or neither humans nor ALBERT-xxLarge got correct, as follows:

1. Humans only: 7 gold-standard incoherent and 73 coherent documents
2. ALBERT-xxLarge only: 181 gold-standard incoherent (of which we found 97% to require fact-checking[9]) and 9 coherent documents (of which 8 contain urls/references, which confused humans)
3. Neither humans nor models: 223 gold-standard incoherent (of which 98.2% and 77.1% were predicted to be coherent by humans and ALBERT-xxLarge, respectively, and for the remainder, the wrong intruder sentence was identified) and 2 coherent documents (both of which were poorly organised, confusing allcomers)

Looking over the incoherent documents which require fact-checking, no obvious differences are discernible between the documents that ALBERT-xxLarge predicts correctly and those it misses. Our assumption here is that ALBERT-xxLarge is biased by the pretraining dataset, and that many of the cases where it makes the wrong prediction are attributable to mismatches between the text in our dataset and the Wikipedia version used in pretraining the model.

6.6.5 Question Revisited

Q1: Do models truly capture the intrinsic properties of document coherence?

A: It is certainly true that models which incorporate a more explicit notion of document coherence into their pretraining (namely ALBERT, RoBERTa, and XLNET) tend to perform

[9]There are 4 documents that humans identify as incoherent based on the wrong intruder sentence, due to the intruder sentence leading to a misleading factual inconsistency.

better, suggesting a tentative yes, but there were equally instances of strong disagreement with human intuitions and model predictions for the better-performing models and evidence to suggest that the models were performing fact-checking at the same time as coherence modelling.

Q2: What types of document incoherence can/can't these models detect?

A: Over incoherent documents resulting from fact inconsistencies, where humans tend to fail, the better-performing models can often make correct predictions; over incoherent documents with information structure or logical inconsistencies which humans can easily detect, ALBERT-Large, RoBERTa-Large, and XLNET-LARGE achieve an Acc ≥ 87%, showing that they can certainly capture information structure and logical inconsistencies to a high degree. That said, the fcat that they misclassify clearly coherent documents as incoherent suggests that are in part lacking in their ability to capture document coherence. We thus can conclude that they can reliably identify intruder sentences which result in a break in information structure or logical flow, but are imperfect models of document coherence.

6.7 Linguistic Probes

To further examine the models, we constructed a language probe dataset.

6.7.1 Linguistic Probe Dataset Construction

We handcrafted adversarial instances based on a range of linguistic phenomena which generate information structure inconsistencies. In constructing such a dataset, minimal modifications were made to sentences, such that it is easy for a human to perceive the difference. For each phenomenon, we hand-constructed 100 adversarial instances by modifying the intruder sentences in incoherent Wikipedia test documents which were pre-filtered for ease of detection, such that they should be relatively easy for the models to detect.[10] The phenomena are as follows:

[10]Except for *Demonstrative*, where there are only 91 relevant examples, in Wikipedia. There are roughly 8×100 instances in total.

1. gender pronoun flip (*Gender*), converting a pronoun to its opposite gender, e.g. *she→ he*;

2. animacy downgrade (*Animacy↓*), downgrading pronouns and possessive determiners to their inanimate versions, e.g. *she/ he/ her/ him→ it,* and *her/ his→ its*;

3. animacy upgrade (*Animacy↑*), upgrading pronouns and possessive determiners to their third person version, e.g. *it→ she/ he/ her/ him*, and *its→ her/ his*;

4. singular demonstrative flip (*Demonstrative*), converting singular demonstratives to plural ones, e.g. *this→ these* and *that→ those*;

5. conjunction flip (*Conjunction*), converting conjunctions to their opposites, e.g. *but→ and therefore, and→ but, although→ therefore*, and vice versa;

6. past tense flip (*Past to Future*), converting past to future tense, e.g. *was→ will be* and *led→ will lead*;

7. sentence negation (*Negation*), negating the sentence, e.g. *He has [a] ... warrant ...→ He doesn't have [a] ... warrant ...*;

8. number manipulation (*Number*), changing numbers to implausible values, e.g. *He served as Chief Operating Officer ... from 2002 to 2005→ He served as Chief Operating Officer ... from 200 BCE to 201 BCE* and *Line 11 has a length of 51.7 km and a total of 18 stations.→ Line 11 has a length of 51.7 m and a total of 1.8 stations.*.

All the probes generate syntactically correct sentences, and the first four generally lead to sentences which are also semantically felicitous, with the incoherence being at the document level. For example, in *He was never convicted and was out on parole within a few years*, if we replace *he* with *she*, the sentence is felicitous, but if the focus entity in the preceding and subsequent sentences is a male, the information flow will be disrupted.

The last four language probes are crafted to explore the capacity of a model to capture commonsense reasoning, in terms of discourse relationships, tense and polarity awareness, and understanding of numbers. For *Conjunction*, we only focus on explicit

Models	*Gender*		*Animacy*↓		*Animacy*↑		*Past to Future*	
	F_1	ΔF_1	F_1	ΔF_1	F_1	ΔF_1	F_1	ΔF_1
BERT-LARGE	27.9	↑64.1	25.0	↑54.1	34.1	↑44.7	35.5	↑41.4
XLNET-LARGE	56.1	↑41.5	49.6	↑45.3	64.2	↑23.2	64.6	↑16.3
RoBERTa-Large	65.4	↑31.2	51.4	↑38.3	59.5	↑22.3	69.2	↑18.8
ALBERT-xxLarge	74.4	↑24.6	72.2	↑9.5	81.0	↑3.0	79.8	↑3.2
BERT-Large$_{CNN}$	22.6	↑71.8	21.0	↑61.3	29.9	↑50.3	29.5	↑16.5
XLNet-Large$_{CNN}$	12.7	↑83.7	9.4	↑71.7	7.4	↑72.1	22.2	↑26.7
RoBERTa-Large$_{CNN}$	15.2	↑82.7	7.8	↑63.4	9.5	↑73.8	23.4	↑39.7
ALBERT-xxLarge$_{CNN}$	23.0	↑70.3	20.0	↑51.7	28.3	↑34.0	38.0	↑30.8
Models	*Conjunction*		*Demonstrative*		*Negation*		*Number*	
	F_1	ΔF_1	F_1	ΔF_1	F_1	ΔF_1	F_1	ΔF_1
BERT-LARGE	49.6	↑18.2	34.2	↑16.5	33.9	↑33.2	31.1	↑31.4
XLNET-LARGE	66.7	↑3.4	56.6	0.0	57.1	↑10.3	51.0	↑10.7
RoBERTa-Large	73.0	↑1.5	57.9	↑1.0	68.4	↑10.9	54.2	↑21.6
ALBERT-xxLarge	83.5	↓2.4	75.2	↑1.4	79.5	↑1.8	63.9	↑10.4
BERT-Large$_{CNN}$	38.5	↓3.9	35.0	↓4.1	29.9	↑3.9	18.8	↑11.2
XLNet-Large$_{CNN}$	29.5	0.0	15.7	0.0	18.0	↑10.8	14.5	↑12.6
RoBERTa-Large$_{CNN}$	33.9	↑2.6	17.8	↓1.8	21.0	↑12.4	18.3	↑24.4
ALBERT-xxLarge$_{CNN}$	41.6	↑2.0	30.9	↓1.4	28.1	↑19.0	23.0	↑18.2

Table 6.7 Results over language probes in incoherent Wikipedia test documents. BERT-Large$_{CNN}$, XLNet-Large$_{CNN}$, RoBERTa-Large$_{CNN}$, and ALBERT-xxLarge$_{CNN}$ are trained over CNN, while BERT-LARGE, XLNET-LARGE, RoBERTa-Large, and ALBERT-xxLarge are trained over Wikipedia. Here, F_1 is over the original incoherent documents (excluding linguistic probes), and ΔF_1 indicates the absolute performance difference resulting from incorporating linguistic probes.

connectives within a sentence. For *Past to Future*, there can be intra-sentence inconsistency if there are time-specific signals, failing which broader document context is needed to pick up on the tense flip. Similarly for *Negation* and *Number*, the change can lead to inconsistency either intra- or inter-sententially. For example, *He did not appear in more than 400 films between 1914 and 1941 ...* is intra-sententially incoherent.

6.7.2 Experimental Results

Table 6.7 lists the performance of pre-trained LMs at recognising intruder sentences within incoherent documents, with and without the addition of the respective linguistic probes.[11] All models achieve near-perfect results with *Gender* linguistic probes, and are also highly

[11] Results for coherent documents are omitted due to space.

successful at detecting *Animacy* mismatches and *Past to Future* (the top half). For the probes in the bottom half, none of these three models except ALBERT-xxLarge performs particularly well, especially for *Demonstrative*. For each linguistic probe, we observe performance improvement for all models by incorporating linguistic probe signals (except for XLNET-LARGE over *Demonstrative* and ALBERT-xxLarge over *Conjunction*).

In the cross-domain setting, the performance of XLNet-Large$_{\text{CNN}}$ and ALBERT-xxLarge$_{\text{CNN}}$ drops across all linguistic probes, while BERT-Large$_{\text{CNN}}$ (over *Gender*, *Animacy*↓, and *Animacy*↑) and Roberta-Large$_{\text{CNN}}$ (*Gender*, and *Animacy*↑) perform better than in-domain. Roberta-Large$_{\text{CNN}}$ achieves the best performance over *Gender*, *Animacy*↑, and *Conjunction* while ALBERT-xxLarge$_{\text{CNN}}$ achieves the best performance over *Conjunction*, *Past to Future*, and *Negation*. The reason that the models tend to struggle with *Demonstrative* and *Conjunction* is not immediately clear, and will be explored in future work.

6.8 Summary

In this chapter, we have addressed the fourth research question – can a quality assessment system detect whether there are intruder sentences in documents and identify the span of any such intruder sentences, given that they interrupt the coherence of documents. We propose new task of detecting whether there is an intruder sentence in a document, generated by replacing an original sentence with a similar sentence from a second document. To benchmark model performance over this task, we construct a large-scale dataset consisting of documents from English Wikipedia and CNN news articles. Experimental results show that pre-trained LMs which incorporate larger document contexts in pretraining perform remarkably well in-domain, but experience a substantial drop cross-domain. In follow-up analysis based on human annotations, substantial divergences from human intuitions were observed, pointing to limitations in their ability to model document coherence. Further results over a linguistic probe dataset show that pre-trained models fail to identify some linguistic characteristics that affect document coherence, suggesting

room to improve for them to truly capture document coherence, and motivating the construction of a dataset with intruder text at the intra-sentential level.

In the next chapter, we will conclude this thesis and point out future research directions in document quality assessment.

Chapter 7

Conclusions

The goal of this thesis is to improve model performance in predicting the quality of documents. To this end, we first presented a comprehensive literature review on document quality assessment across various domains in Chapter 2, outlining the definition, modelling, and evaluation metrics for each task, summarising hand-crafted features and end-to-end models explored in the literature. In the subsequent chapters, we introduced a number of methods for improving document quality assessment, incorporating quality indicators from hand-crafted features to visual indicators, in addition to uncertainty modelling and proposing a new task of detecting whether there is an intruder sentence in a document.

In this chapter, we first revisit our research questions, proposed methods, and findings of each chapter in Section 7.1. Then, we outline a number of potential future research directions in Section 7.2, before finishing this thesis.

7.1 Research Question Revisited

Question 1: *How can we capture visual features not present in the text to enhance representations learned from text content?*

In Chapter 3, we proposed to use visual renderings of documents to capture visual features that are not present in the document text, such as images and visual layout,

which are fed into INCEPTION to learn visual features to predict the quality of documents. Preliminary results over the Wikipedia and arXiv datasets demonstrate its effectiveness in predicting the quality of documents, validating our observation that visual renderings of documents can capture visual features. To go one step further, we propose a joint model (JOINT) combining visual and textual features learned by the visual and textual models, to predict the quality of documents. Further improvements are observed across different datasets, indicating complementarity between visual and textual features and the general applicability of JOINT. To have a better understanding of what kinds of features the visual and textual models have learned, we model document quality assessment and feature learning in a multi-task learning setting. Analysis shows that the visual model is better at capturing structural features and the textual model is better at learning readability scores, confirming our findings that there is complementary between visual and textual features.

Question 2: *How can we make use of hand-crafted features widely adopted in traditional machine learning approaches in the context of neural networks, to generate a more accurate document quality assessment system?*

In Chapter 4, we focused on combining hand-crafted features with neural networks, with the assumption that information carried by hand-crafted features may not be fully captured by neural models. We proposed two methods to incorporate hand-crafted features into neural networks, with one concatenating hand-crafted features with learned representations and the other utilising hand-crafted features to guide neural model learning by explicitly attending to feature indicators. Experimental results over three Wikipedia datasets verify the utility of the first method, while results over the Wikipedia and online petitions datasets, across a variety of neural architectures, demonstrate the superiority of the second method, confirming the utility of hand-crafted features in the context of neural models.

Question 3: *How can we take the inherent subjectivity of quality assessment into consideration in evaluating the performance of quality assessment systems?*

In Chapter 5, we turned our attention to measuring the uncertainty of document quality assessment systems in making their predictions, to imitate people's disagreement over the inherently subjective task of document quality assessment. We investigate two methods: GPs and RFs, both of which can provide predictive distributions, making it feasible to measure their uncertainty over their predictions. Experimental results show that both methods can yield competitive results in predicting the quality of Wikipedia articles, with the advantage of providing an estimate of uncertainty in their predictions. To imitate a real world scenario where decision-making processes based on model predictions can lead to different costs, such as the time and human effort involved in manual reviewing if a low-quality article is predicted to be a featured article versus no manual reviewing in the opposite case, we propose an asymmetric cost, considering the prediction uncertainty, which is in turn used to evaluate the performance of these two methods. Experiments suggest that GPs are a better option if overestimates are penalised more severely.

Question 4: *Can a quality assessment system detect whether there are intruder sentences in documents and identify the span of any such intruder sentences, given that they interrupt the coherence of documents, thereby lowering their quality?*

In Chapter 6, we proposed the new task of detecting whether there is an intruder sentence in a document, generated by replacing an original sentence with a similar sentence from a second document. To benchmark model performance over this task, we construct a large-scale dataset consisting of documents from English Wikipedia and CNN news articles. Experimental results show that pre-trained language models which incorporate larger document contexts in pretraining perform remarkably well in-domain, but experience a substantial drop cross-domain. In follow-up analysis based on human annotations, substantial divergences from human intuitions were observed, pointing to limitations in their ability to model document coherence. Further results over a linguistic probe dataset show that pre-trained models fail to identify some linguistic characteristics that affect document coherence, suggesting room to improve for them to truly capture

document coherence, and motivating the construction of a dataset with intruder text at the intra-sentential level.

7.2 Future Directions

While there are a number of possible ways to extend research in document quality assessment, we name three promising directions to better predict document quality.

7.2.1 Incorporating Discourse Knowledge

Discourse structure in a document weights text chunks differently and connects text chunks with different relationships, from a linguistic perspective. According to the Rhetorical Structure Theory (RST; Mann and Thompson (1988)), a text is parsed into a hierarchical tree, where leaf nodes are non-overlapping text chunks, referred as the elementary discourse units (EDUs), and non-terminal nodes reflect the relations between EDUs. For example, *Farm lending was enacted to correct this problem by providing a reliable flow of lendable funds.*, from the RST Discourse Treebank corpus (Carlson et al., 2001), can be parsed into EDUs and associated relations, as shown in Figure 7.1. Arrows in the figure capture the nuclearity of relations, with a "satellite" pointing to its "nucleus", where the nucleus is usually more important than the satellite. Based on this theory, we can safely say that *Farm lending was enacted* is more important than *to correct this problem*, which in turn is more prominent than *by providing a reliable low of lendable funds.*

The utility of discourse structure has been demonstrated over a range of NLP tasks, such as document summarisation (Koto et al., 2019), and sentiment analysis (Voll and Taboada, 2007, Bhatia et al., 2015). For example, Koto et al. (2019) show the utility of discourse structures obtained from a rhetorical structure theory, in the forms of either engineered shallow features or latent discourse features, over the tasks of abstractive summarisation and popularity prediction of online petitions. Bhatia et al. (2015) propose to incorporate discourse structure in sentiment analysis, either by reweighting each EDU according to its position in the discourse tree or by propagating sentiment up through

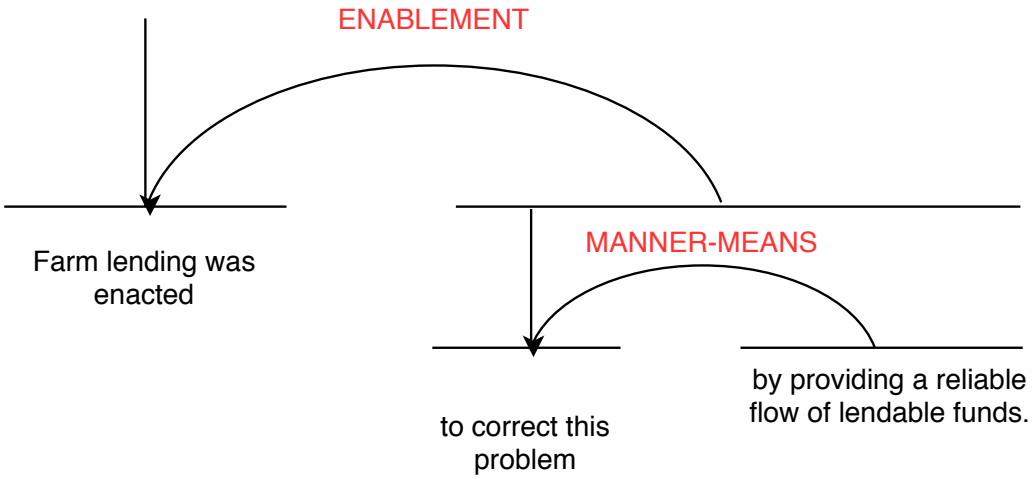


Figure 7.1 Illustration of a simple discourse tree, from Hernault et al. (2010).

the RST parse in a recursive network. Elsewhere, Liu and Lapata (2018) propose to use attention mechanisms to incorporate structural biases in document representation learning. Document quality assessment systems can benefit from incorporating discourse structures, especially for essay scoring. In essay scoring, we measure whether writers master a target language or levels of language proficiency by examining essays written by them. Appropriate discourse structure units in essays can indicate that writers comprehend the target language to a certain degree. Thus, incorporating discourse structure into document quality assessment can be a future line of research.

7.2.2 End-to-End Models in a Generative-Discriminative Fashion

There are two main types of models in tackling NLP tasks, namely the discriminative model and generative model. Specifically, a discriminative model learns the relationship between labels and inputs, where an encoder is used to represent documents. A discriminative model is usually exploited to deal with classification or regression problem, such as sentiment analysis, and document classification. A generative model learns the distribution in a dataset, where an encoder-decoder is needed: the encoder learns source documents representations, and the decoder generates the target documents. A generative model is usually used to tackle sequence-to-sequence tasks, such as machine

translation, and question-answering. A generative-discriminative model performs the discrimination and generation tasks simultaneously, with one task acting as the auxiliary task to the other task or treating both tasks equally.

Research in document quality assessment has mostly focused on discriminative models, ignoring the probability that generative models can potentially improve the performance of models in assessing the quality of documents. The utility of generative-discrinimative models has been demonstrated by previous studies (Deng et al., 2020, Dvornek et al., 2019, Xu et al., 2019b), across a range of tasks, such as answer selection in cQA (Deng et al., 2020), image classification (Xu et al., 2019b), and functional magnetic resonance imaging classification (Dvornek et al., 2019). For example, Deng et al. (2020) propose a generative-discriminative model to tackle the tasks of answer selection and answer summary generation simultaneously. Specifically, the correlated information between question-answer pairs are used to generate question-aware abstractive summaries of original answers, which in turn are used to reduce redundancy and noise in the original answers to improve the performance of answer selection. Dvornek et al. (2019) propose an LSTM based generative-discriminative model to predict classes in the functional magnetic resonance imaging (fMRI) data and generate the fMRI time-series data. With a generative-discriminative model, the performance of document quality assessment systems can be potentially improved, where the generated text can be used as a complementary to the original document or to augment the training data if labelled documents are limited. For example, in assessing the quality of Wikipedia articles, a generative model can be used to generate summaries, which in turn guide the model to attend to salient information in the documents. A similar idea can be adopted for academic paper rating, where the generative model learns to generate paper abstracts. For essay scoring, generated essays can be used to augment the training data when available scored essays are limited, thereby improving models' generalisation ability.

7.2.3 Document Quality Assessment via Multitask Learning

Multitask learning aims to solve multiple tasks in parallel, by learning a shared representation, followed by subsequent task-specific layers. In most approaches embracing multitask learning, there is one main task, accompanied by one or more auxiliary tasks, by assigning different weights to tasks, where tasks can be trained iteratively or jointly. During learning, either the same set of documents or multiple sets of documents, are used to train tasks simultaneously. For example, the Twitter user dataset used in predicting mental health conditions (Benton et al., 2017) can signal the presence of anxiety, and eating disorder at the same time, thus the same set of documents is used to learn the tasks of predicting anxiety and eating disorder. On the other hand, different sets of documents are needed to learn loosely-related tasks, such as image caption generation and machine translation (Luong et al., 2016).

The utility of multitask learning has been demonstrated across various NLP tasks, such as document classification (Liu et al., 2016), syntactic chunking (Søgaard and Goldberg, 2016), and machine translation (Luong et al., 2016). For example, Salehi et al. (2018) propose to incorporate the task of semantic classification of words, as an auxiliary task, into the main task of query segmentation, improving the performance in query segmentation. Søgaard and Goldberg (2016) demonstrate the benefits of introducing an auxiliary task of POS-tagging, supervised at a lower layer, into the tasks of syntactic chunking and combinatory categorical grammar super-tagging. The utility of multitask learning is verified in predicting mental health conditions (Benton et al., 2017). Specifically, the performance in predicting anxiety is improved by utilising tasks such as gender, and suicide prediction. Klerke et al. (2016) show that tagging and gaze prediction help the task of sentence compression. Liu et al. (2016) explore three sharing mechanisms to share information among four document classification tasks, where performance for each task is better than single-task in isolation. Multitask learning can even benefit sequence-to-sequence tasks. For example, Luong et al. (2016) find that coupling syntactic parsing or image caption generation with machine translation improves the performance of machine translation, which in turn enhances the performance in parsing.

Document quality assessment can potentially benefit from multitask learning. For quality assessment of Wikipedia articles, there are not only quality labels but also importance levels in the article talk pages[1]. With the assumption that important articles are more likely to receive intense attention from Wikipedia contributors, thereby a higher chance of improving the quality of articles, learning to predict the importance level of Wikipedia articles can contribute to assessing the quality of Wikipedia articles. For academic paper rating, scores capturing different aspects of papers can be learned simultaneously with paper acceptance prediction. In terms of essay scoring, scores capturing dimensions, such as ideas, organisation, and word choice, can potentially contribute to scoring essays in a holistic view, whenever these scores are available.

7.3 Summary

In summary, this thesis focused on improving the performance of document quality assessment systems, modelling uncertainty over their predictions, and understanding how existing pre-trained transformer models perform in document coherence modelling. Document quality assessment, due to its complexity and subjectivity, requires consideration of information from multiple sources and aspects, capturing quality indicators. While traditional machine learning approaches based upon hand-crafted features and neural models are proposed to predict the quality of documents, the complementary between hand-crafted features and learned representations, and visual features have not been taken into consideration. In this thesis, we introduced a number of methods for improving document quality assessment, incorporating quality indicators from hand-crafted features to visual indicators, in addition to uncertainty modelling. Furthermore, we proposed a novel task of detecting whether there is an intruder sentence in a document, which we suggested is important in assessing document quality. Experimental results have demonstrated the utility of our proposed methods, making it one step closer to document quality assessment and establishing a strong base on which to build future studies.

[1] `https://en.wikipedia.org/wiki/Help:Talk_pages`

References

L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. 2008. Knowledge sharing and yahoo answers: Everyone knows something. In *Proceedings of the 17th International Conference on World Wide Web*, pages 665–674.

W. Adouane, J. Bernardy, and S. Dobnik. 2018a. Improving neural network performance by injecting background knowledge: Detecting code-switching and borrowing in Algerian texts. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 20–28.

W. Adouane, S. Dobnik, J.-P. Bernardy, and N. Semmar. 2018b. A comparison of character neural language model and bootstrapping for language identification in multilingual noisy texts. In *Proceedings of the Second Workshop on Subword/Character LEVEL Models*, pages 22–31.

E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 183–194.

D. Alikaniotis, H. Yannakoudakis, and M. Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 715–725.

M. Alikhani, S. Nag Chowdhury, G. de Melo, and M. Stone. 2019. CITE: A corpus of image-text discourse relations. In *Proceedings of the 2019 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 570–575.

E. Amorim, M. Cançado, and A. Veloso. 2018. Automated essay scoring in the presence of biased ratings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 229–237.

K. Arai and A. N. Handayani. 2013. Predicting quality of answer in collaborative Q/A community. *International Journal of Advanced Research in Artificial Intelligence*, 2(3):21–25.

N. M. Asher and A. Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Y. Attali and J. Burstein. 2004. Automated essay scoring with e-rater® v.2.0. *The Journal of Technology, Learning and Assessment*, 4(3):1–30.

S. Baccianella, A. Esuli, and F. Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2200–2204.

D. Balchev, Y. Kiprov, I. Koychev, and P. Nakov. 2016. PMI-cool at SemEval-2016 task 3: Experiments with PMI and goodness polarity lexicons for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 844–850.

U. Barman, A. Das, J. Wagner, and J. Foster. 2014a. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23.

U. Barman, J. Wagner, G. Chrupala, and J. Foster. 2014b. DCU-UVT: Word-level language classification with code-mixed data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 127–132.

M. Baroni, G. Dinu, and G. Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.

A. Barrón-Cedeño, S. Filice, G. Da San Martino, S. Joty, L. Màrquez, P. Nakov, and A. Moschitti. 2015. Thread-level information for comment classification in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 687–693.

A. Barrón-Cedeño, G. D. S. Martino, S. R. Joty, A. Moschitti, F. Al-Obaidli, S. Romeo, K. Tymoshenko, and A. Uva. 2016. ConvKN at SemEval-2016 task 3: Answer and question selection for question answering on Arabic and English fora. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 896–903.

R. Barzilay and M. Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148.

R. Barzilay and M. Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

D. Beck. 2017. Modelling representation noise in emotion analysis using Gaussian processes. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 140–145.

D. Beck and T. Cohn. 2017. Learning kernels over strings using Gaussian processes. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 67–73.

D. Beck, L. Specia, and T. Cohn. 2016. Exploring prediction uncertainty in machine translation quality estimation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 208–218.

Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872.

Y. Belinkov and J. Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Y. Belinkov, M. Mohtarami, S. Cyphers, and J. R. Glass. 2015. VectorSLU: A continuous word vector approach to answer selection in community question answering systems. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 282–287.

J. Benesty, J. Chen, Y. Huang, and I. Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

A. Benton, M. Mitchell, and D. Hovy. 2017. Multitask learning for mental health conditions with limited social media data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 152–162.

G. G. Betancourt, A. Segnine, C. Trabuco, A. Rezgui, and N. Jullien. 2016. Mining team characteristics to predict Wikipedia article quality. In *Proceedings of the 12th International Symposium on Open Collaboration*, pages 15:1–15:9.

P. Bhatia, Y. Ji, and J. Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218.

J. Bian, Y. Liu, E. Agichtein, and H. Zha. 2008. Finding the right facts in the crowd: Factoid question answering over social media. In *Proceedings of the 17th International Conference on World Wide Web*, pages 467–476.

J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha. 2009. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th International Conference on World Wide Web*, pages 51–60.

S. Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

T. Blevins, O. Levy, and L. Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19.

M. J. Blooma, A. Y. Chua, and D. H. Goh. 2008. A predictive framework for retrieving the best answer. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 1107–1111.

M. J. Blooma, A. Y. Chua, and D. H. Goh. 2011. What makes a high-quality user-generated answer? *IEEE Internet Computing*, 15(1):66–71.

M. J. Blooma, D. H. Goh, and A. Y. Chua. 2012. Predictors of high-quality answers. *Online Information Review*, 36(3):383–400.

J. E. Blumenstock. 2008. Size matters: Word count as a measure of quality on Wikipedia. In *Proceedings of the 17th International Conference on World Wide Web*, pages 1095–1096.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

L. Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

T. Briscoe. 2011. Introduction to linguistics for natural language processing. *Computer Laboratory, University of Cambridge*, 4.

T. Briscoe, B. Medlock, and Ø. Andersen. 2010. Automated assessment of ESOL free text examinations. Technical report, University of Cambridge, Computer Laboratory.

E. Bruni, N. Tran, and M. Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47.

G. Burel, Y. He, and H. Alani. 2012. Automatic identification of best answers in online enquiry communities. In *Proceedings of the 9th Extended Semantic Web Conference*, pages 514–529.

G. Burel, P. Mulholland, and H. Alani. 2016. Structural normalisation methods for improving best answer identification in question answering communities. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 673–678.

J. Burstein and D. Higgins. 2005. Advanced capabilities for evaluating student writing: Detecting off-topic essays without topic-specific training. In *Proceedings of the International Conference on Artificial Intelligence in Education*, pages 112–119.

J. Burstein, D. Marcu, and K. Knight. 2003. Finding the WRITE stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39.

F. Calefato, F. Lanubile, and N. Novielli. 2016. Moving to Stack Overflow: Best-answer prediction in legacy developer forums. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 13:1–13:10.

J. Callan, M. Hoy, C. Yoo, and L. Zhao. 2009. Clueweb09 data set.

M. Campr and K. Ježek. 2015. Comparing semantic models for evaluating automatic document summarization. In *Proceedings of the 18th International Conference on Text, Speech, and Dialogue*, pages 252–260.

J. Q. Candela, C. E. Rasmussen, F. H. Sinz, O. Bousquet, and B. Schölkopf. 2005. Evaluating semantic evaluations: How RTE measures up. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, volume 3944, pages 1–27.

L. Carlson, D. Marcu, and M. E. Okurovsky. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–10.

D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

K. Chai, P. Hayati, V. Potdar, C. Wu, and A. Talevski. 2010. Assessing post usage for measuring the quality of forum posts. In *Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies*, pages 233–238.

J. S. Chall and E. Dale. 1995. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.

K. Chalupka, C. K. I. Williams, and I. Murray. 2013. A framework for evaluating approximation methods for Gaussian process regression. *Journal of Machine Learning Research*, 14(1):333–350.

K. Chandu, E. Nyberg, and A. W. Black. 2019. Storyboarding of recipes: Grounded contextual generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6040–6046.

D. Charlet and G. Damnati. 2017. SimBow at SemEval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 315–319.

D. Chen, S. Ma, P. Yang, and X. Sun. 2018. Identifying high-quality Chinese news comments based on multi-target text matching model. *arXiv preprint arXiv:1808.07191*.

H.-H. Chen. 2012. How to use readability formulas to access and select English reading materials. *Journal of Educational Media and Library Sciences*, 50(2):229–254.

M. Chen, Z. Chu, and K. Gimpel. 2019. Evaluation benchmarks and learning criteria for discourse-aware sentence representations. In *Proceedings of the 2019 Conference*

on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 649–662.

Q. Chen, Q. Hu, J. X. Huang, L. He, and W. An. 2017. Enhancing recurrent neural networks with positional attention for question answering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 993–996.

M. Cheng, J. Yi, H. Zhang, P.-Y. Chen, and C.-J. Hsieh. 2018. Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*.

G. Chittaranjan, Y. Vyas, K. Bali, and M. Choudhury. 2014. Word-level language identification using CRF: Code-switching shared task report of MSR India system. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 73–79.

K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Y. Cho and L. K. Saul. 2009. Kernel methods for deep learning. In *Proceedings of the Twenty-third Conference on Neural Information Processing Systems*, pages 342–350.

F. Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807.

P. F. Christoffersen and F. X. Diebold. 1997. Optimal prediction under asymmetric loss. *Econometric Theory*, 13(6):808–817.

A. Y. K. Chua and S. Banerjee. 2015. Measuring the effectiveness of answers in Yahoo! Answers. *Online Information Review*, 39(1):104–118.

O. D. Clercq and V. Hoste. 2016. All mixed up? Finding the optimal feature set for general readability prediction and its application to English and Dutch. *Computational Linguistics*, 42(3):457–490.

O. D. Clercq, V. Hoste, B. Desmet, P. van Oosten, M. D. Cock, and L. Macken. 2014. Using the crowd for readability prediction. *Natural Language Engineering*, 20(3):293–325.

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

J. Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

T. Cohn and L. Specia. 2013. Modelling annotator bias with multi-task Gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32–42.

M. Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283–284.

A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

M. Cozma, A. M. Butnaru, and R. T. Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 503–509.

S. Crossley, L. K. Allen, E. L. Snow, and D. S. McNamara. 2015. Pssst... textual features... there is more to automatic essay scoring than just you! In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 203–207.

R. Cummins and M. Rei. 2018. Neural multi-task learning in automated assessment. *arXiv preprint arXiv:1801.06830*.

E. Dale and J. S. Chall. 1948. A formula for predicting readability: Instructions. *Educational Research Bulletin*, 27(1):37–54.

D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado. 2009. Automatic quality assessment of content created collaboratively by web communities: A case study of Wikipedia. In *Proceedings of the 18th Joint Conference on Digital Libraries*, pages 295–304.

D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado. 2011. Automatic assessment of document quality in web collaborative digital libraries. *Journal of Data and Information Quality*, 2(3):14:1–14:30.

D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado. 2013. Exploiting user feedback to learn to rank answers in q&a forums: A case study with Stack Overflow. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 543–552.

D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado. 2017. A general multiview framework for assessing the quality of collaboratively created content on web 2.0. *Journal of the Association for Information Science and Technology*, 68(2):286–308.

D. H. Dalip, H. Lima, M. A. Gonçalves, M. Cristo, and P. Calado. 2014. Quality assessment of collaborative content with minimal information. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 201–210.

Q.-V. Dang and C.-L. Ignat. 2016a. Measuring quality of collaboratively edited documents: The case of Wikipedia. In *Proceedings of the 2nd IEEE International Conference on Collaboration and Internet Computing*, pages 266–275.

Q.-V. Dang and C.-L. Ignat. 2016b. Quality assessment of Wikipedia articles: A deep learning approach. *SIGWEB Newsletter*, 5(6):5:1–5:6.

Q.-V. Dang and C.-L. Ignat. 2016c. Quality assessment of Wikipedia articles without feature engineering. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, pages 27–30.

Q.-V. Dang and C.-L. Ignat. 2017. An end-to-end learning solution for assessing the quality of Wikipedia articles. In *Proceedings of the 13th International Symposium on Open Collaboration*, pages 4:1–4:10.

A. Das and B. Gambäck. 2014. Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387.

Y. Deng, W. Lam, Y. Xie, D. Chen, Y. Li, M. Yang, and Y. Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*.

Y. Deng, Y. Xie, Y. Li, M. Yang, N. Du, W. Fan, K. Lei, and Y. Shen. 2019. Multi-task learning with multi-view attention for answer selection and knowledge base question answering. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, pages 6318–6325.

J. Deriu and M. Cieliebak. 2017. SwissAlps at SemEval-2017 task 3: Attention-based convolutional neural network for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 334–338.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, pages 138–145.

P. Dondio and S. Barrett. 2007. Computational trust in web content quality: A comparative evaluation on the Wikipedia project. *Informatica (Slovenia)*, 31(2):151–160.

F. Dong, Y. Zhang, and J. Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162.

N. C. Dvornek, X. Li, J. Zhuang, and J. S. Duncan. 2019. Jointly discriminative and generative recurrent neural networks for learning from fMRI. In *International Workshop on Machine Learning in Medical Imaging*, pages 382–390.

J. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

M. Elsner, J. Austerweil, and E. Charniak. 2007. A unified local and global model for discourse coherence. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 436–443.

M. Elsner and E. Charniak. 2008. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT*, pages 41–44.

M. Elsner and E. Charniak. 2011. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129.

A. Fan, M. Lewis, and Y. Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660.

H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, and M. Ester. 2016. Community-based question answering via heterogeneous social network learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 122–128.

N. Farra, S. Somasundaran, and J. Burstein. 2015. Scoring persuasive essays using opinions and their targets. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–74.

S. Filice, D. Croce, A. Moschitti, and R. Basili. 2016. KeLP at SemEval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1116–1123.

S. Filice, G. D. S. Martino, and A. Moschitti. 2017. KeLP at SemEval-2017 task 3: Learning pairwise patterns in community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 326–333.

P. W. Foltz, D. Laham, and T. K. Landauer. 1999. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1(2):939–944.

M. Franco-Salvador, S. Kar, T. Solorio, and P. Rosso. 2016. UH-PRHLT at SemEval-2016 task 3: Combining lexical and semantic-based features for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 814–821.

Y. Gal, M. van der Wilk, and C. E. Rasmussen. 2014. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Proceedings of the Twenty-eighth Conference on Neural Information Processing Systems*, pages 3257–3265.

B. A. Galitsky. 2013. Machine learning of syntactic parse trees for search and classification of text. *Engineering Applications of Artificial Intelligence*, 26(3):1072–1091.

S. Gella, K. Bali, and M. Choudhury. 2014. "ye word kis lang ka hai bhai?" testing the limits of word level language identification. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 368–377.

T. Ghosal, D. Dey, A. Dutta, A. Ekbal, S. Saha, and P. Bhattacharyya. 2019a. A multiview clustering approach to identify out-of-scope submissions in peer review. In *Proceedings of the 18th Joint Conference on Digital Libraries*, pages 392–393.

T. Ghosal, A. Raj, A. Ekbal, S. Saha, and P. Bhattacharyya. 2019b. A deep multimodal investigation to determine the appropriateness of scholarly submissions. In *Proceedings of the 18th Joint Conference on Digital Libraries*, pages 227–236.

T. Ghosal, R. Sonam, S. Saha, A. Ekbal, and P. Bhattacharyya. 2018a. Investigating domain features for scope detection and classification of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 7–12.

T. Ghosal, R. Verma, A. Ekbal, and P. Bhattacharyya. 2019c. DeepSentiPeer: Harnessing sentiment in review texts to recommend peer review decisions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1120–1130.

T. Ghosal, R. Verma, A. Ekbal, and P. Bhattacharyya. 2019d. A sentiment augmented deep architecture to predict peer review outcomes. In *Proceedings of the 18th Joint Conference on Digital Libraries*, pages 414–415.

T. Ghosal, R. Verma, A. Ekbal, S. Saha, and P. Bhattacharyya. 2018b. Investigating impact features in editorial pre-screening of research papers. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pages 333–334.

D. Ghosh, A. Khanam, Y. Han, and S. Muresan. 2016. Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 549–554.

G. Gkotsis, K. Stepanyan, C. Pedrinaci, J. Domingue, and M. Liakata. 2014. It's all in the content: State of the art best answer prediction based on discretisation of shallow linguistic features. In *Proceedings of the 2014 ACM Conference on Web Science*, pages 202–210.

Y. Graham, T. Baldwin, A. Moffat, and J. Zobel. 2017. Can machine translation systems be evaluated by the crowd alone? *Natural Language Engineering*, 23(1):3–30.

H. P. Grice. 2002. Logic and conversation. *Foundations of Cognitive Psychology*, pages 719–732.

C. Guinaudeau and M. Strube. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 93–103.

K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, and M. Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205.

R. Gunning. 1969. The Fog index after twenty years. *International Journal of Business Communication*, 6(2):3–13.

F. Guzmán, L. Màrquez, and P. Nakov. 2016a. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 460–466.

F. Guzmán, P. Nakov, and L. Màrquez. 2016b. MTE-NN at SemEval-2016 task 3: Can machine translation evaluation help community question answering? In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 887–895.

A. Halfaker and D. Taraborelli. 2015. Artificial intelligence service ORES gives Wikipedians X-ray specs to see through bad edits. *Wikimedia Blog*.

L. Han. 2016. Machine translation evaluation resources and methods: A survey. *arXiv preprint arXiv:1605.04515*.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

J. Hensman, N. Fusi, and N. D. Lawrence. 2013. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, page 282–290.

K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the Twenty-ninth Conference on Neural Information Processing Systems*, pages 1693–1701.

J. M. Hernández-Lobato and R. P. Adams. 2015. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1861–1869.

H. Hernault, H. Prendinger, M. Ishizuka, et al. 2010. Hilda: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).

J. Hewitt and P. Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743.

J. Hewitt and C. D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

F. Heylighen and J.-M. Dewaele. 2002. Variation in the contextuality of language: An empirical measure. *Foundations of Science*, 7(3):293–340.

F. Hieber and S. Riezler. 2011. Improved answer ranking in social question-answering portals. In *Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents*, pages 19–26.

D. Higgins, J. Burstein, and Y. Attali. 2006. Identifying off-topic student essays without topic-specific training data. *Natural Language Engineering*, 12(2):145–159.

D. Higgins, J. Burstein, D. Marcu, and C. Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 185–192.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

A. Holtzman, J. Buys, M. Forbes, A. Bosselut, D. Golub, and Y. Choi. 2018. Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*.

Y. Hou, C. Tan, X. Wang, Y. Zhang, J. Xu, and Q. Chen. 2015. HITSZ-ICRC: Exploiting classification approach for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 196–202.

D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. H. Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130.

W.-N. Hsu, Y. Zhang, and J. Glass. 2016. Recurrent neural network encoder with attention for community question answering. *arXiv preprint arXiv:1603.07044*.

H. Hu, B. Liu, B. Wang, M. Liu, and X. Wang. 2013. Multimodal DBN for predicting high-quality answers in cQA portals. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 843–847.

J. Hu, S. Y. Chen, and R. Levy. 2020a. A closer look at the performance of neural language models on reflexive anaphor licensing. In *Proceedings of the Society for Computation in Linguistics (SCiL 2020)*, pages 382–392.

J. Hu, Y. Cheng, Z. Gan, J. Liu, J. Gao, and G. Neubig. 2020b. What makes a good story? Designing composite rewards for visual storytelling. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, pages 7969–7976.

M. Hu, E. Lim, A. Sun, H. W. Lauw, and B. Vuong. 2007. Measuring article quality in Wikipedia: Models and evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pages 243–252.

J. Huang. 2018. Deep paper gestalt. *arXiv preprint arXiv:1812.08775*.

J. Huang, M. Zhou, and D. Yang. 2007. Extracting chatbot knowledge from online discussion forums. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 423–428.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. 2011. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, pages 507–523.

C. J. Hutto and E. Gilbert. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM*, pages 216–225.

K. Järvelin and J. Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transaction of In Information Systems*, 20(4):422–446.

S. Javanmardi and C. V. Lopes. 2010. Statistical measure of quality in Wikipedia. In *Proceedings of the First Workshop on Social Media Analytics*, pages 132–138.

J. Jeon, W. B. Croft, J. H. Lee, and S. Park. 2006. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval, pages 228–235.

Y. Ji and N. A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005.

C. Jin, B. He, K. Hui, and L. Sun. 2018. TDNN: A two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1097.

M. Johnson and E. Charniak. 2004. A TAG-based noisy-channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 33–39.

S. R. Joty, A. Barrón-Cedeño, G. D. S. Martino, S. Filice, L. Màrquez, A. Moschitti, and P. Nakov. 2015. Global thread-level inference for comment classification in community question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 573–578.

S. R. Joty, L. Màrquez, and P. Nakov. 2016. Joint learning with global inference for comment classification in community question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 703–713.

S. R. Joty, L. Màrquez, and P. Nakov. 2018. Joint multitask learning for community question answering using task-specific embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4196–4207.

P. Jurczyk and E. Agichtein. 2007. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pages 919–922.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665.

D. Kang, W. Ammar, B. Dalvi, M. van Zuylen, S. Kohlmeier, E. H. Hovy, and R. Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661.

Y. Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

J. P. Kincaid, R. P. Fishburne Jr., R. L. Rogers, and B. S. Chissom. 1975. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease formula) for Navy enlisted personnel. Technical report, DTIC Document.

D. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

W. Kintsch and T. A. Van Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review*, 85(5):363–394.

R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. 2015. Skip-thought vectors. In *Proceedings of the Twenty-ninth Conference on Neural Information Processing Systems*, pages 3294–3302.

B. B. Klebanov and M. Flor. 2013. Word association profiles and their use for automated scoring of essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1148–1158.

J. M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

S. Klerke, Y. Goldberg, and A. Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1528–1533.

T. Kocmi and O. Bojar. 2017. LanideNN: Multilingual language identification on character window. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 927–936.

F. Koto, J. H. Lau, and T. Baldwin. 2019. Improved document modelling with a neural discourse parser. In *Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association*, pages 67–76.

A. Kuncoro, C. Dyer, J. Hale, D. Yogatama, S. Clark, and P. Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

B. de La Robertie, Y. Pitarch, and O. Teste. 2015. Measuring article quality in Wikipedia using the collaboration network. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 464–471.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, San Francisco, USA.

A. Lai and J. Tetreault. 2018. Discourse coherence in the wild: A dataset, evaluation and methods. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 214–223.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations*. OpenReview.net.

T. Landauer, D. Laham, and P. Foltz. 2003. Automatic essay assessment. *Assessment in Education: Principles, Policy & Practice*, 10(3):295–308.

T. K. Landauer, P. W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284.

M. Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 545–552.

J. H. Lau and T. Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86.

J. H. Lau, A. Clark, and S. Lappin. 2015. Unsupervised prediction of acceptability judgements. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1618–1628.

A. Lavie and M. J. Denkowski. 2009. The Meteor metric for automatic evaluation of Machine Translation. *Machine Translation*, 23(2-3):105–115.

L. T. Le, C. Shah, and E. Choi. 2016. Evaluating the quality of educational answers in community question-answering. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, pages 129–138.

Q. V. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196.

Y. Leng, L. Yu, and J. Xiong. 2019. DeepReviewer: Collaborative grammar and innovation neural network for automatic paper review. In *Proceedings of the 21st ACM International Conference on Multimodal Interaction*, pages 395–403.

W. Lewoniewski, N. Khairova, K. Węcel, N. Stratiienko, and W. Abramowicz. 2017. Using morphological and semantic features for the quality assessment of Russian Wikipedia.

In *Proceedings of the 23nd International Conference on Information and Software Technologies*, pages 550–560.

D. Li, Y. Yu, Q. Chen, and X. Li. 2019. BERTSel: Answer selection with pre-trained models. *arXiv preprint arXiv:1905.07588*.

J. Li and D. Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209.

X. Li, M. Chen, J. Nie, Z. Liu, Z. Feng, and Y. Cai. 2018. Coherence-based automated essay scoring using self-attention. In *Proceedings of the 17th China National Conference on Computational Linguistics, CCL 2018, and the 6th International Symposium on Natural Language Processing Based on Naturally Annotated Big Data, NLP-NABD 2018*, pages 386–397.

X. Li, J. Tang, T. Wang, Z. Luo, and M. De Rijke. 2015. Automatically assessing Wikipedia article quality by exploiting article–editor networks. In *Proceedings of the 37th European Conference on Information Retrieval*, pages 574–580.

B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4208–4215.

E. Lim, B. Vuong, H. W. Lauw, and A. Sun. 2006. Measuring qualities of articles contributed by online communities. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 81–87.

C. Lin, W. Ammar, L. S. Levin, and C. Dyer. 2014. The CMU submission for the shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 80–86.

N. Lipka and B. Stein. 2010. Identifying featured articles in Wikipedia: Writing style matters. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1147–1148.

P. Liu, X. Qiu, and X. Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2873–2879.

Y. Liu and E. Agichtein. 2008. You've got answers: Towards personalized models for predicting success in community question answering. In *Proceedings of ACL-08: HLT*, pages 97–100.

Y. Liu, J. Bian, and E. Agichtein. 2008. Predicting information seeker satisfaction in community question answering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 483–490.

Y. Liu and M. Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

L. Logeswaran, H. Lee, and D. R. Radev. 2018. Sentence ordering and coherence modeling using recurrent neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5285–5292.

B. Longo. 1994. The role of metadiscourse in persuasion. *Technical Communication*, 41(2):348–352.

D. Lonsdale and D. Strong-Krause. 2003. Automated rating of ESL essays. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 61–67.

P. J. Lou, P. Anderson, and M. Johnson. 2018. Disfluency detection using auto-correlational neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4610–4619.

P. J. Lou and M. Johnson. 2017. Disfluency detection using a noisy channel model and a deep neural language model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 547–553.

A. Louis. 2013. Predicting text quality: Metrics for content, organization and reader interest.

A. Louis and D. Higgins. 2010. Off-topic essay detection using short prompt texts. In *Proceedings of the NAACL HLT 2010: Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 92–95.

A. Louis and A. Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing*, pages 1157–1168.

A. Louis and A. Nenkova. 2014. Verbose, laconic or just right: A simple computational model of content appropriateness under length constraints. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 636–644.

M. Lui and T. Baldwin. 2009. You are what you post: User-level features in threaded discourse. In *Proceedings of the 14th Australasian Document Computing Symposium (ADCS 2009)*, pages 98–105.

M. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. 2016. Multi-task sequence to sequence learning. In *Proceedings of the 4th International Conference on Learning Representations*.

L. van der Maaten and G. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Rearch*, 9:2579–2605.

K. Maleewong. 2016. Predicting quality-assured consensual answers in community-based question answering systems. In *Proceedings of the 12th International Conference on Computing and Information Technology*, pages 117–127.

W. C. Mann and S. A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

E. Marrese-Taylor, P. Loyola, and Y. Matsuo. 2019. An edit-centric approach for Wikipedia article quality assessment. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 381–386.

R. Marvin and T. Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. 2017. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

D. Mave, S. Maharjan, and T. Solorio. 2018. Language identification and analysis of code-switched social media text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 51–61.

T. McCoy, E. Pavlick, and T. Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

D. L. McGuinness, H. Zeng, P. P. da Silva, L. Ding, D. Narayanan, and M. Bhaowal. 2006. Investigations into trust for collaborative information repositories: A Wikipedia case study. In *Proceedings of the WWW'06 Workshop on Models of Trust for the Web*, pages 3–131.

K. McKeown, H. Daume III, S. Chaturvedi, J. Paparrizos, K. Thadani, P. Barrio, O. Biran, S. Bothe, M. Collins, K. R. Fleischmann, et al. 2016. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology*, 67(11):2684–2696.

G. H. McLaughlin. 1969. SMOG grading: A new readability formula. *Journal of Reading*, 12(8):639–646.

D. S. McNamara, S. A. Crossley, R. D. Roscoe, L. K. Allen, and J. Dai. 2015. A hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23:35–59.

M. Mesgar and M. Strube. 2016. Lexical coherence graph modeling using word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1414–1423.

M. Mesgar and M. Strube. 2018. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4328–4339.

T. Mihaylov and P. Nakov. 2016. SemanticZ at SemEval-2016 task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 879–886.

T. Mihaylova, P. Gencheva, M. Boyanov, I. Yovcheva, T. Mihaylov, M. Hardalov, Y. Kiprov, D. Balchev, I. Koychev, P. Nakov, I. Nikolova, and G. Angelova. 2016. SUper team at SemEval-2016 task 3: Building a feature-rich system for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 836–843.

G. A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

E. Miltsakaki and K. Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10(1):25–55.

F. S. Mim, N. Inoue, P. Reisert, H. Ouchi, and K. Inui. 2019. Unsupervised learning of discourse-aware text representation for essay scoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 378–385.

R. Mohamed, M. Ragab, H. Abdelnasser, N. M. El-Makky, and M. Torki. 2015. Al-bayan: A knowledge-based system for Arabic answer selection. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 226–230.

M. Mohtarami, Y. Belinkov, W. Hsu, Y. Zhang, T. Lei, K. Bar, S. Cyphers, and J. R. Glass. 2016. SLS at SemEval-2016 task 3: Neural-based approaches for ranking in community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 828–835.

G. Molina, F. AlGhamdi, M. Ghoneim, A. Hawwari, N. Rey-Villamizar, M. T. Diab, and T. Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49.

P. Molino, L. M. Aiello, and P. Lops. 2016. Social question answering: Textual, user, and network features for best answer prediction. *ACM Transactions on Information Systems*, 35(1):4:1–4:40.

J. Morris and G. Hirst. 1991a. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

J. Morris and G. Hirst. 1991b. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

A. Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, pages 318–329.

B. Murray. 1998. The latest techno tool: Essay-grading computers. *American Psychological Association*, 8(29).

R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, and B. Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

S. M. Nasehi, J. Sillito, F. Maurer, and C. Burns. 2012. What makes a good code example?: A study of programming Q&A in StackOverflow. In *Proceedings of the 28th IEEE International Conference on Software Maintenance*, pages 25–34.

D. T. Nguyen and S. R. Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330.

M. Nicosia, S. Filice, A. Barrón-Cedeno, I. Saleh, H. Mubarak, W. Gao, P. Nakov, G. Da San Martino, A. Moschitti, K. Darwish, et al. 2015. QCRI: Answer selection for community question answering - experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 203–209.

N. Ong, D. J. Litman, and A. Brusilovsky. 2014. Ontology-based argument mining and automatic essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 24–28.

E. B. Page. 1968. The use of the computer in analyzing student essays. *International Review of Education*, 14(2):210–225.

E. B. Page. 1994. Computer grading of student prose, using modern concepts and software. *The Journal of Experimental Education*, 62(2):127–142.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The PageRank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172.

N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long. 2018. Technical report on the Cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

C. C. Park and G. Kim. 2015. Expressing an image stream with a sequence of natural sentences. In *Advances in neural information processing systems*, pages 73–81.

R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda. 2011. English Gigaword. *Linguistic Data Consortium*.

R. J. Passonneau and B. Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326.

S. Paun, B. Carpenter, J. Chamberlain, D. Hovy, U. Kruschwitz, and M. Poesio. 2018. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

J. Pennebaker, C. Chung, M. Ireland, A. Gonzales, and R. Booth. 2007. The development and psychometric properties of LIWC2007. *Austin, TX, LIWC. Net*.

J. Pennington, R. Socher, and C. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

I. Persing, A. Davis, and V. Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239.

I. Persing and V. Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269.

I. Persing and V. Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543.

M. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

P. Phandi, K. M. A. Chai, and H. T. Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439.

M. Piergallini, R. A. Shirvani, G. S. Gautam, and M. F. Chouikha. 2016. Word-level language identification and predicting codeswitching points in Swahili-English language data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 21–29.

E. Pitler, A. Louis, and A. Nenkova. 2010. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 544–554.

E. Pitler and A. Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195.

L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza. 2014. Understanding and classifying the quality of technical forum questions. In *Proceedings of the 14th International Conference on Quality Software*, pages 343–352.

S. Prabhumoye, R. Salakhutdinov, and A. W. Black. 2020. Topological sort for sentence ordering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2783–2792.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The Penn discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 2961–2968.

J. W. G. Putra and T. Tokunaga. 2017. Evaluating text coherence based on semantic similarity graph. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 76–85.

F. Qiao, L. Xu, and X. Han. 2018. Modularized and attention-based recurrent convolutional neural network for automatic academic paper aspect scoring. In *Proceedings of the 15th International Conference on Web Information Systems and Applications*, pages 68–76.

D. R. Radev, S. Teufel, H. Saggion, W. Lam, J. Blitzer, H. Qi, A. Celebi, D. Liu, and E. Drabek. 2003. Evaluation challenges in large-scale document summarization. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 375–382.

Y. Ran, B. He, and J. Xu. 2018. A study on performance sensitivity to data sparsity for automated essay scoring. In *Proceedings of the 11th International Conference on Knowledge Science, Engineering and Management*, pages 104–116.

C. E. Rasmussen and C. K. Williams. 2006. *Gaussian Process for Machine Learning*. MIT Press.

M. Recasens, C. Danescu-Niculescu-Mizil, and D. Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659.

E. M. Roochi, K. Tao, B. Haslhofer, and G. Houben. 2013. Identification of useful user comments in social media: A case study on flickr commons. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 1–10.

L. M. Rudner, V. Garcia, and C. Welch. 2006. An evaluation of IntelliMetric™ essay scoring system. *The Journal of Technology, Learning and Assessment*, 4(4):1–21.

L. M. Rudner and T. Liang. 2002. Automated essay scoring using Bayes' theorem. *The Journal of Technology, Learning and Assessment*, 1(2):3–21.

M. Sahlgren. 2005. An introduction to random indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*.

B. Salehi, F. Liu, T. Baldwin, and W. Wong. 2018. Multitask learning for query segmentation in job search. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 179–182.

S. Samanta and S. Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

Y. Samih, S. Maharjan, M. Attia, L. Kallmeyer, and T. Solorio. 2016. Multilingual code-switching identification via LSTM recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59.

Y. Samih and W. Maier. 2016. Detecting code-switching in Moroccan Arabic social media. *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media*.

M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4323–4330.

M. Schreiner, B. Rehder, T. Landauer, and D. Laham. 1997. How latent semantic analysis LSA represents essay semantic content: Technical issues and analysis. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, page 1041.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pages 618–626.

C. Shah. 2015. Building a parsimonious model for identifying best answers using interaction history in community Q&A. *Journal of the Association for Information Science and Technology*, 52(1):1–10.

C. Shah and J. Pomerantz. 2010. Evaluating and predicting answer quality in community QA. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 411–418.

K. Simonyan and A. Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*.

D. D. Sleator and D. Temperley. 1995. Parsing English with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, pages 277–292.

E. A. Smith and R. J. Senter. 1967. Automated readability index. *Aerospace Medical Research Laboratories (U.S.)*, pages 1–14.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

A. Søgaard and Y. Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235.

T. Solorio, E. Blair, S. Maharjan, S. Bethard, M. T. Diab, M. Ghoneim, A. Hawwari, F. AlGhamdi, J. Hirschberg, A. Chang, and P. Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

S. Somasundaran, J. Burstein, and M. Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 950–961.

C. Spearman. 1961. The proof and measurement of association between two things. *American Journal of Psychology*, 15(1):72–101.

L. Specia, D. Raj, and M. Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50.

K. Stein and C. Hess. 2007. Does it matter who contributes: A study on featured articles in the German Wikipedia. In *Proceedings of the Eighteenth Conference on Hypertext and Hypermedia*, pages 171–174.

P. J. Stone, R. F. Bales, J. Z. Namenwirth, and D. M. Ogilvie. 1962. The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. *Systems Research and Behavioral Science*, 7(4):484–498.

W. Strunk. 2007. *The Elements of Style*. Penguin Books.

B. Stvilia, L. Gasser, M. B. Twidale, and L. C. Smith. 2007. A framework for information quality assessment. *Journal of the Association for Information Science and Technology*, 58(12):1720–1733.

B. Stvilia, M. B. Twidale, L. C. Smith, and L. Gasser. 2005. Assessing information quality of a community-based Encyclopedia. In *Proceedings of the 20th International Conference on Information Quality*, pages 442–454.

S. Subramanian, T. Baldwin, and T. Cohn. 2018. Content-based popularity prediction of online petitions using a deep regression model. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 182–188.

S. P. Suggu, K. N. Goutham, M. K. Chinnakotla, and M. Shrivastava. 2016a. Deep feature fusion network for answer quality prediction in community question answering. *arXiv preprint arXiv:1606.07103*.

S. P. Suggu, K. N. Goutham, M. K. Chinnakotla, and M. Shrivastava. 2016b. Hand in glove: Deep feature fusion network architectures for answer quality prediction in community question answering. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1429–1440.

M. A. Suryanto, E. Lim, A. Sun, and R. H. L. Chiang. 2009. Quality-aware collaborative question answering: Methods and evaluation. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 142–151.

Y. Suzuki. 2015. Quality assessment of Wikipedia articles using h-index. *Journal of Information Processing*, 23(1):22–30.

Y. Suzuki and M. Yoshikawa. 2012. Mutual evaluation of editors and texts for assessing quality of Wikipedia articles. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, pages 18:1–18:10.

Y. Suzuki and M. Yoshikawa. 2013. Assessing quality score of Wikipedia article using mutual evaluation of editors and texts. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 1727–1732.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

K. Taghipour and H. T. Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891.

M. Tan, C. N. dos Santos, B. Xiang, and B. Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 464–473.

D. Tang, B. Qin, and T. Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

G. Tang, M. Müller, A. Rios, and R. Sennrich. 2018. Why self-attention? A targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272.

Y. R. Tausczik and J. W. Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54.

Y. Tay, M. C. Phan, L. A. Tuan, and S. C. Hui. 2018. SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5948–5955.

Q. Tian and B. Li. 2016. Weakly hierarchical lasso based learning to rank in best answer prediction. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 307–314.

Q. Tian, P. Zhang, and B. Li. 2013. Towards predicting the best answers in community-based question-answering services. In *Proceedings of the 7th International Conference on Weblogs and Social Media*, pages 725–728.

M. K. Titsias. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 567–574.

H. Toba, Z.-Y. Ming, M. Adriani, and T.-S. Chua. 2014. Discovering high quality answers in community question answering archives using a hierarchy of classifiers. *Information Sciences*, 261:101–115.

K. Tran, A. Bisazza, and C. Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736.

Q. H. Tran, V. Tran, T. Vu, M. Nguyen, and S. B. Pham. 2015. JAIST: combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 215–219.

T. H. Trinh and Q. V. Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

H. Varian. 1975. A Bayesian approach to real estate assessment. *Studies in Bayesian Econometrics and Statistics in Honor of Leonard J. Savage*, pages 195–208.

P. Vincent-Lamarre and V. Larivière. 2019. Textual analysis of artificial intelligence manuscripts reveals features associated with peer review outcome. *arXiv*, pages arXiv–1911.

N. P. A. Vo, S. Magnolini, and O. Popescu. 2015. FBK-HLT: An application of semantic textual similarity for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 231–235.

K. D. Voll and M. Taboada. 2007. Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence*, pages 337–346.

H. Wachsmuth, K. A. Khatib, and B. Stein. 2016. Using argument mining to assess the argumentation quality of essays. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691.

N. M. Wanas, M. El-Saban, H. Ashour, and W. Ammar. 2008. Automatic scoring of online discussion posts. In *Proceedings of the 2nd ACM Workshop on Information Credibility on the Web*, pages 19–26.

G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin. 2018a. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331.

K. Wang and X. Wan. 2018. Sentiment analysis of peer review texts for scholarly papers. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 175–184.

P. Wang and X. Li. 2020. Assessing the quality of information on Wikipedia: A deep-learning approach. *Journal of the Association for Information Science and Technology*, 71(1):16–28.

P. Wang, X. Li, and R. Wu. 2019. A deep learning-based quality assessment model of collaboratively edited documents: A case study of Wikipedia. *Journal of Information Science*.

S. Wang and M. Iwaihara. 2011. Quality evaluation of Wikipedia articles through edit history and editor groups. In *Proceedings of the 13th Asia-Pacific Web Conference on Web Technologies and Applications*, pages 188–199.

X.-J. Wang, X. Tu, D. Feng, and L. Zhang. 2009. Ranking community answers by modeling question-answer relationships via analogical reasoning. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 179–186.

Y. Wang, Z. Wei, Y. Zhou, and X. Huang. 2018b. Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 791–797.

M. Warncke-Wang, V. R. Ayukaev, B. Hecht, and L. Terveen. 2015. The success and failure of quality improvement projects in peer production communities. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 743–756.

M. Warncke-Wang, D. Cosley, and J. Riedl. 2013. Tell me more: An actionable quality model for Wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration*, pages 8:1–8:10.

A. Warstadt, A. Singh, and S. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

B. Webber. 2009. Genre distinctions for discourse in the Penn TreeBank. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 674–682.

M. Weimer and I. Gurevych. 2007. Predicting the perceived quality of web forum posts. In *Proceedings of the 2007 Conference on Recent Advances in Natural Language Processing (RANLP-07)*, pages 643–648.

M. Weimer, I. Gurevych, and M. Mühlhäuser. 2007. Automatically assessing the post quality in online discussions on software. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 125–128.

Wikipedia Dump. Accessed Dec., 2018. Wikipedia dump. https://archive.org/download/enwiki-20171020.

Wikipedia Statistics. Accessed April, 2020. Statistics. `https://en.wikipedia.org/wiki/Wikipedia:Statistics`.

E. Wilcox, R. Levy, T. Morita, and R. Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221.

D. Wilson and D. Sperber. 2004. Relevance theory: A theory. In *Proceedings of the Third Tokyo Conference on Psycholinguistics*, pages 45–70.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354.

T. Wöhner and R. Peters. 2009. Assessing the quality of Wikipedia articles with lifecycle based metrics. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, pages 16:1–16:10.

B. Woods, D. Adamson, S. Miel, and E. Mayfield. 2017. Formative essay feedback using predictive scoring models. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2071–2080.

G. Wu and M. Lan. 2016. ECNU at SemEval-2016 task 3: Exploring traditional method and deep learning method for question retrieval and answer ranking in community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 872–878.

M. Wu, F. Liu, and T. Cohn. 2018a. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856.

W. Wu, X. Sun, and H. Wang. 2018b. Question condensing networks for answer selection in community question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1746–1755.

Y. Xiang, Q. Chen, X. Wang, and Y. Qin. 2017. Answer selection in community question answering via attentive neural networks. *IEEE Signal Processing Letters*, 24(4):505–509.

P. Xu, H. Saghir, J. S. Kang, T. Long, A. J. Bose, Y. Cao, and J. C. K. Cheung. 2019a. A cross-domain transferable neural coherence model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 678–687.

Y. Xu, M. Gong, J. Chen, T. Liu, K. Zhang, and K. Batmanghelich. 2019b. Generative-discriminative complementary learning. *arXiv preprint arXiv:1904.01612*.

Y. Xu and T. Luo. 2011. Measuring article quality in Wikipedia: Lexical clue model. In *Proceedings of the 3rd Symposium on Web Society*, pages 141–146.

R. Yan, J. Tang, X. Liu, D. Shan, and X. Li. 2011. Citation count prediction: learning to estimate future citations for literature. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1247–1252.

M. Yang, W. Tu, Q. Qu, W. Zhou, Q. Liu, and J. Zhu. 2019a. Advanced community question answering by leveraging external knowledge and multi-task learning. *Knowledge-Based Systems*, 171:106–119.

P. Yang, X. Sun, W. Li, and S. Ma. 2018. Automatic academic paper rating based on modularized hierarchical convolutional neural network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 496–502.

P. Yang, J. Chen, C.-J. Hsieh, J.-L. Wang, and M. I. Jordan. 2020. Greedy attack and Gumbel attack: Generating adversarial examples for discrete data. *Journal of Machine Learning Research*, 21(43):1–36.

X. Yang, M. Khabsa, M. Wang, W. Wang, A. H. Awadallah, D. Kifer, and C. L. Giles. 2019b. Adversarial training for community question answer selection based on multi-scale matching. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, pages 395–402.

Y. Yang, W.-t. Yih, and C. Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018.

Y. Yang and A. Nenkova. 2017. Combining lexical and syntactic features for detecting content-dense texts in news. *Journal of Artificial Intelligence Research*, 60:179–219.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. 2019c. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the Thirty-third Conference on Neural Information Processing Systems*, pages 5754–5764.

Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

H. Yannakoudakis. 2013. Automated assessment of English-learner writing. Technical report, University of Cambridge, Computer Laboratory.

H. Yannakoudakis, T. Briscoe, and B. Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.

Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu. 2013. Want a good answer? Ask a good question first! *arXiv preprint arXiv:1311.6876*.

Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu. 2015. Detecting high-quality posts in community question answering sites. *Information Sciences*, 302:70–82.

Z. Yirmibesoglu and G. Eryigit. 2018. Detecting code-switching between Turkish-English language pair. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 110–115.

D. Yogatama, M. Heilman, B. O'Connor, C. Dyer, B. R. Routledge, and N. A. Smith. 2011. Predicting a scientific community's response to an article. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 594–604.

I. Zamanov, M. Kraeva, N. Hateva, I. Yovcheva, I. Nikolova, and G. Angelova. 2015. Voltron: A hybrid system for answer validation based on lexical and distance features. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 242–246.

R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104.

R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

H. Zeng, M. A. Alhossaini, L. Ding, R. Fikes, and D. L. McGuinness. 2006. Computing trust from revision history. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*, pages 8:1–8:8.

J. Zhang, X. Wang, and H. Xu. 2008. Survey of automatic summarization evaluation methods. *Journal of Chinese Information Processing*, 22(3):81–88.

S. Zhang, Z. Hu, C. Zhang, and K. Yu. 2018. History-based article quality assessment on Wikipedia. In *Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing*, pages 1–8.

X. Zhang, J. Zhao, and Y. LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the Twenty-ninth Conference on Neural Information Processing Systems*, pages 649–657.

X. Zhang, S. Li, L. Sha, and H. Wang. 2017. Attentive interactive neural networks for answer selection in community question answering. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3525–3531.

S. Zhao, Y. Zhang, X. Xiong, A. Botelho, and N. T. Heffernan. 2017. A memory-augmented neural model for automated grading. In *Proceedings of the Fourth ACM Conference on Learning @ Scale*, pages 189–192.

X. Zhou, B. Hu, Q. Chen, B. Tang, and X. Wang. 2015a. Answer sequence learning with neural networks for answer selection in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 713–718.

X. Zhou, B. Hu, Q. Chen, and X. Wang. 2018. Recurrent convolutional neural network for answer selection in community question answering. *Neurocomputing*, 274:8–18.

X. Zhou, B. Hu, J. Lin, Y. Xiang, and X. Wang. 2015b. ICRC-HIT: A deep learning based comment sequence labeling system for answer selection challenge. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 210–214.

X. Zhu and S. Gauch. 2000. Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 288–295.

Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pages 19–27.