University Library



A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Kim, SN;Medelyan, O;Kan, MY;Baldwin, T

Title:
Automatic keyphrase extraction from scientific articles

Date:
2013-09-01

Citation:
Kim, S. N., Medelyan, O., Kan, M. Y. & Baldwin, T. (2013). Automatic keyphrase extraction from scientific articles. Language Resources and Evaluation, 47 (3), pp.723-742. https://doi.org/10.1007/s10579-012-9210-3.

Persistent Link:
https://hdl.handle.net/11343/283246

Automatic Keyphrase Extraction from Scientific Articles

Su Nam Kim,♠ Olena Medelyan,♡ Min-Yen Kan♢ and Timothy Baldwin♠
♠ *Dept of Computing and Information Systems, The University of Melbourne, Australia*
♡ *Pingar, Auckland, New Zealand*
♢ *School of Computing, National University of Singapore, Singapore*

sunamkim@gmail.com,
alyona.medelyan@pingar.com,
kanmy@comp.nus.edu.sg,
tb@ldwin.net

Month Date, Year

Abstract. This paper describes the organization and results of the automatic keyphrase extraction task held at the workshop on Semantic Evaluation 2010 (SemEval-2010). The keyphrase extraction task was specifically geared towards scientific articles. Systems were automatically evaluated by matching their extracted keyphrases against those assigned by the authors as well as the readers to the same documents. We outline the task, present the overall ranking of the submitted systems, and discuss the improvements to the state-of-the-art in keyphrase extraction.

Keywords: keyphrase extraction, scientific document processing, SemEval-2010, shared task

1. Introduction

Keyphrases[1] are words that capture the main topics of a document. Extracting high-quality keyphrases can benefit various natural language processing (NLP) applications: in text summarization, keyphrases are useful as a form of semantic metadata indicating the significance of sentences and paragraphs, in which they appear (Barzilay and Elhadad, 1997; Lawrie et al., 2001; D'Avanzo and Magnini, 2005); in both text categorization and document clustering, keyphrases offer a means of term dimensionality reduction, and have been shown to improve system efficiency and accuracy (Zhang et al., 2004; Hammouda et al., 2005; Hulth and Megyesi, 2006; Wang et al., 2008; Kim et al., 2009); and for search engines, keyphrases can supplement full-text indexing and assist users in formulating queries (Gutwin et al., 1999; Gong and Liu, 2008).

Recently, a resurgence of interest in automatic keyphrase extraction has led to the development of several new systems and techniques for the task, as outlined in Section 2. However, a common base for evaluation has been missing, which has made it hard to perform comparative evaluation of different systems. In light of these developments, we felt that the time was

[1] We use "keyphrase" and "keywords" interchangeably to refer to both single words and multiword expressions.

© 2012 *Kluwer Academic Publishers. Printed in the Netherlands.*

ripe to conduct a shared task on keyphrase extraction, to provide a standard evaluation framework for the task to benchmark current and future systems against.

For our SemEval-2010 Task 5 on keyphrase extraction, we compiled a set of 244 scientific articles with keyphrase annotations from authors and readers. The task was to develop systems which automatically produce keyphrases for each paper. Each team was allowed to submit up to three system runs, to benchmark the contributions of different parameter settings and approaches. The output for each run took the form of a ranked list of 15 keyphrases from each document, ranked by their probability of being keyphrases.

In the remainder of the paper, we first detail related work (Section 2) then describe the task setup, including how data collection was managed and the evaluation methodology (Sections 3 and 4). We present the results of the shared task, and discuss the immediate findings of the competition in Section 5. In Sections 6 and 7, we present a short description of submitted systems and the human performance by comparing reader-assigned keyphrases to those assigned by the authors, giving an approximation of the upper-bound performance for this task. Finally, we conclude our work in Section 8.

2. Related Work

Previous work on automatic keyphrase extraction has broken down the task into four components: (1) candidate identification, (2) feature engineering, (3) developing learning models, and (4) evaluating the extracted keyphrases.

Given a document, *candidate identification* is the task of detecting all keyphrase candidates, in the form of nouns or noun phrases mentioned in the document. The majority methods are based on n-grams (Frank et al., 1999; Hulth, 2003; Tomokiyo and Hurst, 2003; Paukkeri et al., 2008) or POS sequences (Turney, 1999; Barker and Corrnacchia, 2000; Nguyen and Kan, 2007; Kim and Kan, 2009), or both. Some approaches employ heuristics aimed at reducing the number of false-positive candidates while maintaining the true positives. A comprehensive analysis of the accuracy and coverage of candidate extraction methods was carried out by Hulth (2004). She compared three methods: n-grams (excluding those that begin or end with a stop word), POS sequences (pre-defined) and NP-chunks, excluding initial determiners (*a*, *an* and *the*). No single method dominates, and the best results were achieved by voting across the three methods.

The second step of *feature engineering* involves the development of features with which to characterize individual keyphrase candidates, and has been extensively researched in the literature. The majority of proposed features combine frequency statistics within a single document and across an entire collection, semantic similarity among keyphrases (i.e. keyphrase co-

hesion), popularity of keyphrases among manually assigned sets, lexical and morphological analysis, and heuristics such as locality and the length of phrases. The most popular and best-performing single feature is $TF \times IDF$, which is often used as a baseline feature (Frank et al., 1999; Witten et al., 1999; Nguyen and Kan, 2007; Liu et al., 2009a). $TF \times IDF$ highlights those candidate phrases which are particularly frequent in a given document, but less frequent in the overall document collection. Keyphrase cohesion is another widely-used feature. Since keyphrases are intended to capture the topic of a document, they are likely to have higher semantic similarity among themselves than non-keyphrases. Turney (2003) measured keyphrase cohesion within the top-N keyphrase candidates versus the remaining candidates using web frequencies. Others have used term co-occurrence of candidates (Matsuo and Ishizuka, 2004; Mihalcea and Tarau, 2004; Ercan, 2006; Liu et al., 2009b; Liu et al., 2009a) while Ercan (2006) and Medelyan and Witten (2006) used taxonomic relations such as hypernymy and hyponymy. Ercan (2006) additionally built lexical chains based on term senses. As a heuristic feature, the locality of terms is often used. Frank et al. (1999) and Witten et al. (1999) introduced the relative position of the first occurrence of the term, while Nguyen and Kan (2007) and Kim and Kan (2009) analyzed the location and frequency of candidates in terms of document sections, leveraging structure in their dataset (i.e. scientific articles).

Keyphrase extraction is generally construed as a ranking problem — i.e. candidates are ranked based on their feature values, and the top-N ranked candidates are returned as keyphrases. As such, the third step is *developing learning models* with which to rank the candidates. The majority of learning approaches are supervised, with commonly-employed learners being maximum entropy models (Nguyen and Kan, 2007; Kim and Kan, 2009), naïve Bayes (Frank et al., 1999; Turney, 1999; Ercan, 2006), decision trees (Turney, 1999) and support vector machines (Krapivin et al., 2010). Others have proposed simpler probabilistic models using measures such as pointwise mutual information and KL-divergence (Barker and Corrnacchia, 2000; Tomokiyo and Hurst, 2003; Matsuo and Ishizuka, 2004). More recently, unsupervised methods have gained popularity, using graphs and semantic networks to rank candidates (Mihalcea and Tarau, 2004; Litvak and Last, 2008; Liu et al., 2009a; Li et al., 2010).

The final step is *evaluating the extracted keyphrases*. Automatic keyphrase extraction systems have commonly been assessed using the proportion of top-N candidates that exactly match the gold-standard keyphrases (Frank et al., 1999; Witten et al., 1999; Turney, 1999). This number is then used to compute the precision, recall and F-score for a keyphrase set. However, the exact matching of keyphrases is problematic because it ignores near matches that are largely semantically identical, such as synonyms, different grammatical forms, or sub/super-strings of keyphrases, e.g. *linguistic graduate*

program versus *graduate program*. To remedy this, in some cases, inexact matches (sometimes termed "near misses" or "near matches") have also been considered. Some have suggested treating semantically-similar keyphrases as correct based on similarities computed over a large corpus (Jarmasz and Barriere, 2004; Mihalcea and Faruque, 2004), or using semantic relations defined in a thesaurus (Medelyan and Witten, 2006). Zesch and Gurevych (2009) computed near matches using an n-gram based approach relative to the gold standard. To differentiate between plausible near matches and completely erroneous keyphrases, evaluation metrics have been proposed that take into account semantic similarity (Jarmasz and Barriere, 2004; Medelyan and Witten, 2006; Paukkeri et al., 2008) and character n-grams (Zesch and Gurevych, 2009; Kim et al., 2010). However, these metrics have yet to gain traction in the research community.

3. Keyphrase Extraction Datasets

3.1. Existing Datasets

There are several publicly available datasets for evaluating keyphrase extraction, which we detail below.

Hulth (2003) compiled 2,000 journal article abstracts from Inspec, published between the years 1998 and 2002. The dataset contains keyphrases (i.e. controlled and uncontrolled terms) assigned by professional indexers, to 1,000 documents for training, 500 for validation and 500 for testing.

Nguyen and Kan (2007) collected a dataset containing 120 computer science articles, ranging in length from 4 to 12 pages. The articles contain author-assigned keyphrases as well as reader-assigned keyphrases contributed by undergraduate CS students. Krapivin et al. (2009) obtained 2,304 articles from the same source from 2003 to 2005, with author-assigned keyphrases. They marked up the document text with sub-document extents for fields such as title, abstract and references.

In the general newswire domain, Wan and Xiao (2008) developed a dataset of 308 documents taken from DUC 2001, with up to 10 manually-assigned keyphrases per document.

Several databases, including the ACM Digital Library, IEEE Xplore, Inspec and PubMed, provide articles with author-assigned keyphrases and, occasionally, reader-assigned keyphrases. Schutz (2008) collected a set of 1,323 medical articles from PubMed with author-assigned keyphrases.

Medelyan et al. (2009) automatically generated a dataset using tags assigned by users of the collaborative citation platform CiteULike. This dataset additionally records how many people have assigned the same keyword to the

same publication. In total, 180 full-text publications were annotated by over 300 users.[2]

Despite the availability of these datasets, a standardized benchmark dataset with a well-defined training and test split, and standardized evaluation scripts, is needed to maximize comparability of results. This was our primary motivator in running the SemEval-2010 task.

We have consolidated all of datasets listed above as well as the new dataset and evaluation scripts used for SemEval-2010 into a single repository for public download.[3] We hope that the dataset forms a reference dataset to aid more comparative evaluation for future keyphrase endeavors.

3.2. COLLECTING THE SEMEVAL-2010 DATASET

To collect the dataset for this task, we downloaded data from the ACM Digital Library (conference and workshop papers) and partitioned it into trial, training and test subsets. The input papers ranged from 6 to 8 pages, including tables and figures. To ensure a variety of different topics is represented in the corpus, we purposefully selected papers from four different research areas. In particular, the selected articles belong to the following four 1998 ACM classifications: C2.4 (Distributed Systems), H3.3 (Information Search and Retrieval), I2.11 (Distributed Artificial Intelligence – Multiagent Systems) and J4 (Social and Behavioral Sciences – Economics). All three datasets (trial, training and test) had an equal distribution of documents from among the categories (see Table I). This domain-specific information was made available to task participants, to see whether customized solutions would work better within specific sub-areas.

Participants were provided with 40, 144, and 100 articles, respectively, in the trial, training and test data, distributed evenly across the four research areas in each case. Note that the trial data was a subset of the training data that participants were allowed to use in the task. Since the original format for the articles was PDF, we converted them into (UTF-8 encoded) plain text using pdftotext, and systematically restored full words that were originally hyphenated and broken across lines. This policy potentially resulted in valid hyphenated forms having their hyphen removed.

All of the collected papers contained author-assigned keyphrases as part of the original PDF file, which were removed from the text dump of the paper. We additionally collected reader-assigned keyphrases for each paper. We first performed a pilot annotation task with a group of students to check the stability of the annotations, finalize the guidelines, and discover and resolve potential issues that may occur during the actual annotation. To collect the actual reader-assigned keyphrases, we then hired 50 student annotators from

[2] http://bit.ly/maui-datasets

[3] http://github.com/snkim/AutomaticKeyphraseExtraction

Table I. Number of documents per topic in the trial, training and test datasets, across the four ACM document classifications of **C**2.4, **H**3.3, **I**2.11 and **J**4

Dataset	Total	Document Topic			
		C	H	I	J
Trial	40	10	10	10	10
Training	144	34	39	35	36
Test	100	25	25	25	25

the Computer Science department of the National University of Singapore. We assigned 5 papers to each annotator, estimating that assigning keyphrases to each paper would take about 10-15 minutes. Annotators were explicitly told to extract keyphrases that actually appeared in the text of each paper, rather than to create semantically-equivalent phrases. They were also told that they could extract phrases from any part of the document inclusive of headers and captions. Despite these directives, 15% of the reader-assigned keyphrases do not appear in the actual text of the paper, although this is still less than the corresponding figure for author-assigned keyphrases, at 19%.[4] In other words, the maximum recall that the participating systems can achieve on these documents is 85% and 81% for the reader- and author-assigned keyphrases, respectively.

As some keyphrases may occur in multiple but semantically-equivalent forms, we expanded the set of keyphrases to include alternative versions of genitive keyphrases: B *of* $A = A\ B$ (e.g. *policy of school* = *school policy*), and A*'s* $B = A\ B$ (e.g. *school's policy* = *school policy*). We chose to implement only this limited form of keyphrase equivalence in our evaluation, as these two alternations both account for a large portion of the keyphrase variation, and were relatively easy to explain to participants and for them to reimplement. Note, however, that the genitive alternation does change the semantics of the candidate phrase in limited cases (e.g. *matter of fact* versus *?fact matter*). To deal with this, we hand-vetted all keyphrases generated through these alternations, and did not include alternative forms that were judged to be semantically distinct.

Table I shows the distribution of the trial, training and test documents over the four different research areas, while Table II shows the distribution of author- and reader-assigned keyphrases. Interestingly, among the 387 author-

Table II. Number of author- and reader-assigned keyphrases in the different portions of the dataset

Component	Author	Reader	Combined
Trial	149	526	621
Training	559	1824	2223
Test	387	1217	1482

assigned keywords, 125 keywords match exactly with reader-assigned keywords, while many more near matches occur.

4. Evaluation Method and Baseline

For the evaluation we adopt the traditional means of matching auto-generated keyphrases against those assigned by experts (the gold-standard). Prior to computing the matches, all keyphrases are stemmed using the English Porter stemmer.[5] We assume that auto-generated keyphrases are supplied in ranked order starting from the most relevant keyphrase. The top-5, top-10 and top-15 keyphrases are then compared against the gold-standard for the evaluation.

As an example, let us compare a set of 15 top-ranking keyphrases generated by one of the competitors and stemmed using the Porter stemmer:

grid comput, grid, grid servic discoveri, web servic, servic discoveri, grid servic, uddi, distribut hash tabl, discoveri of grid, uddi registri, rout, proxi registri, web servic discoveri, qos, discoveri

with the equivalent gold-standard set of 19 keyphrases (a combined set assigned by both authors and readers):

grid servic discoveri, uddi, distribut web-servic discoveri architectur, dht base uddi registri hierarchi, deploy issu, bamboo dht code, case-insensit search, queri, longest avail prefix, qo-base servic discoveri, autonom control, uddi registri, scalabl issu, soft state, dht, web servic, grid comput, md, discoveri

The system has correctly identified 6 keyphrases, which results in a precision of 40% (6/15) and recall of 31.6% (6/19). Given the results for each individual

[4] These values were computed using the test documents only.

[5] Using the Perl implementation available at `http://tartarus.org/~martin/PorterStemmer/`; we informed participants that this was the stemmer we would be using for the task, to avoid possible stemming variations between implementations.

Table III. Keyphrase extraction performance for baseline unsupervised (TF×IDF) and supervised (ME) systems, in terms of precision (P), recall (R) and F-score (F), given as percentages

Method	Keyphrases	Top-5 candidates			Top-10 candidates			Top-15 candidates		
		P	R	F	P	R	F	P	R	F
TF×IDF	Reader	17.8	7.4	10.4	13.9	11.5	12.6	11.6	14.5	12.9
	Author	10.0	12.9	11.3	7.9	20.4	11.4	6.5	25.3	10.4
	Combined	22.0	7.5	11.2	17.7	12.1	14.4	14.9	15.3	15.1
ME	Reader	16.8	7.0	9.9	13.3	11.1	12.1	11.4	14.2	12.7
	Author	10.4	13.4	11.7	7.9	20.4	11.4	6.3	24.3	10.0
	Combined	21.4	7.3	10.9	17.3	11.8	14.0	14.5	14.9	14.7

document, we then calculate the micro-averaged precision, recall and F-score ($\beta = 1$) for each cut off (5, 10 and 15).[6] Please note that the maximum recall that could be achieved over the combined keyphrase set was approximately 75%, because not all keyphrases actually appear in the document.

Participants were required to extract keyphrases from among the phrases used in a given document. Since it is theoretically possible to access the original PDF articles and extract the author-assigned keyphrases, we evaluate systems over the independently generated reader-assigned keyphrases, as well as the combined set of keyphrases (author- and reader-assigned).

We computed a *TF×IDF* n-gram based baseline using both supervised and unsupervised approaches. First, we generated 1-, 2- and 3-grams as keyphrase candidates for both the test and training data. For training documents, we identified keyphrases using the set of manually-assigned keyphrases for that document. Then, we used a maximum entropy (ME) learner to learn a supervised baseline model based on the keyphrase candidates, *TF×IDF* scores and gold-standard annotations for the training documents.[7] For the unsupervised learning system, we simply use *TF×IDF* scores (higher to lower) as the basis of our keyphrase candidate ranking. Therefore in total, there are two baselines: one supervised and one unsupervised. The performance of the baselines is presented in Table III, broken down across reader-assigned keyphrases (Reader), author-assigned keyphrases (Author), and combined author- and reader-assigned keyphrases (Combined).

[6] An alternative approach could have been to use a more fine-grained evaluation measure which takes into account the relative ranking of different keyphrases at a given cutoff, such as nDCG (Jarvelin and Kekalainen, 2002).

[7] We also experimented with a naive Bayes learner, but found the results to be identical to the ME learner due to the simplicity of the feature set.

Table IV. Performance of the submitted systems over the combined author- and reader-assigned keywords, ranked by Top-15 F-score

System	Rank	Top-5 candidates			Top-10 candidates			Top-15 candidates		
		P	R	F	P	R	F	P	R	F
HUMB	1	39.0	13.3	19.8	32.0	21.8	26.0	27.2	27.8	27.5
WINGNUS	2	40.2	13.7	20.5	30.5	20.8	24.7	24.9	25.5	25.2
KP-Miner	3	36.0	12.3	18.3	28.6	19.5	23.2	24.9	25.5	25.2
SZTERGAK	4	34.2	11.7	17.4	28.5	19.4	23.1	24.8	25.4	25.1
ICL	5	34.4	11.7	17.5	29.2	19.9	23.7	24.6	25.2	24.9
SEERLAB	6	39.0	13.3	19.8	29.7	20.3	24.1	24.1	24.6	24.3
KX_FBK	7	34.2	11.7	17.4	27.0	18.4	21.9	23.6	24.2	23.9
DERIUNLP	8	27.4	9.4	13.9	23.0	15.7	18.7	22.0	22.5	22.3
Maui	9	35.0	11.9	17.8	25.2	17.2	20.4	20.3	20.8	20.6
DFKI	10	29.2	10.0	14.9	23.3	15.9	18.9	20.3	20.7	20.5
BUAP	11	13.6	4.6	6.9	17.6	12.0	14.3	19.0	19.4	19.2
SJTULTLAB	12	30.2	10.3	15.4	22.7	15.5	18.4	18.4	18.8	18.6
UNICE	13	27.4	9.4	13.9	22.4	15.3	18.2	18.3	18.8	18.5
UNPMC	14	18.0	6.1	9.2	19.0	13.0	15.4	18.1	18.6	18.3
JU_CSE	15	28.4	9.7	14.5	21.5	14.7	17.4	17.8	18.2	18.0
Likey	16	29.2	10.0	14.9	21.1	14.4	17.1	16.3	16.7	16.5
UvT	17	24.8	8.5	12.6	18.6	12.7	15.1	14.6	14.9	14.8
POLYU	18	15.6	5.3	7.9	14.6	10.0	11.8	13.9	14.2	14.0
UKP	19	9.4	3.2	4.8	5.9	4.0	4.8	5.3	5.4	5.3

5. Competition Results

The trial data was downloaded by 73 different teams, of which 36 teams subsequently downloaded the training and test data. 21 teams participated officially in the final competition, of which two teams withdrew their systems from the published set of results.

Table IV shows the performance of the final 19 teams. 5 teams submitted one run, 6 teams submitted two runs, and 8 teams submitted the maximum number of three runs. We rank the best-performing run for each team by micro-averaged F-score over the top-15 candidates. We also show system performance over reader-assigned keywords in Table V, and over author-assigned keywords in Table VI. In all these tables, P, R and F denote precision, recall and F-score, respectively. The systems are ranked in the descending order of their F-score over the top-15 candidates.

The best results over the reader-assigned and combined keyphrase sets are 23.5% and 27.5%, respectively, achieved by the *HUMB* team. Most systems outperformed the baselines. Systems generally scored better against the com-

Table V. Performance of the submitted systems over the reader-assigned keywords, ranked by Top-15 F-score

System	Rank	Top-5 candidates			Top-10 candidates			Top-15 candidates		
		P	R	F	P	R	F	P	R	F
HUMB	1	30.4	12.6	17.8	24.8	20.6	22.5	21.2	26.4	23.5
KX_FBK	2	29.2	12.1	17.1	23.2	19.3	21.1	20.3	25.3	22.6
SZTERGAK	3	28.2	11.7	16.6	23.2	19.3	21.1	19.9	24.8	22.1
WINGNUS	4	30.6	12.7	18.0	23.6	19.6	21.4	19.8	24.7	22.0
ICL	5	27.2	11.3	16.0	22.4	18.6	20.3	19.5	24.3	21.6
SEERLAB	6	31.0	12.9	18.2	24.1	20.0	21.9	19.3	24.1	21.5
KP-Miner	7	28.2	11.7	16.5	22.0	18.3	20.0	19.3	24.1	21.5
DERIUNLP	8	22.2	9.2	13.0	18.9	15.7	17.2	17.5	21.8	19.5
DFKI	9	24.4	10.1	14.3	19.8	16.5	18.0	17.4	21.7	19.3
UNICE	10	25.0	10.4	14.7	20.1	16.7	18.2	16.0	19.9	17.8
SJTULTLAB	11	26.6	11.1	15.6	19.4	16.1	17.6	15.6	19.4	17.3
BUAP	12	10.4	4.3	6.1	13.9	11.5	12.6	14.9	18.6	16.6
Maui	13	25.0	10.4	14.7	18.1	15.0	16.4	14.9	18.5	16.1
UNPMC	14	13.8	5.7	8.1	15.1	12.5	13.7	14.5	18.0	16.1
JU_CSE	15	23.4	9.7	13.7	18.1	15.0	16.4	14.4	17.9	16.0
Likey	16	24.6	10.2	14.4	17.9	14.9	16.2	13.8	17.2	15.3
POLYU	17	13.6	5.7	8.0	12.6	10.5	11.4	12.0	14.9	13.3
UvT	18	20.4	8.5	12.0	15.6	13.0	14.2	11.9	14.9	13.2
UKP	19	8.2	3.4	4.8	5.3	4.4	4.8	4.7	5.8	5.2

bined set, as the availability of a larger gold-standard answer set means that more correct cases could be found among the top-5, 10 and 15 keyphrases, which lead to a better balance between precision and recall scores, resulting in a higher F-score.

In Tables VII and VIII, we present system rankings across the four ACM document classifications, ranked in order of top-15 F-score. The numbers in parentheses are the actual F-scores for each team. Note that in the case of a tie in F-score, we sub-ranked the teams in descending order of F-score over the full dataset.

6. A Summary of the Submitted Systems

The following is an overview of the systems which participated in the task, ranked according to their position in the overall system ranking. They are additionally labelled as being supervised or unsupervised, based on whether they made use of the keyphrase-labelled training data. Systems which did not have an accompanying description paper are omitted.

Table VI. Performance of the submitted systems over the author-assigned keywords, ranked by Top-15 F-score

System	Rank	Top-5 candidates			Top-10 candidates			Top-15 candidates		
		P	R	F	P	R	F	P	R	F
HUMB	1	21.2	27.4	23.9	15.4	39.8	22.2	12.1	47.0	19.3
KP-Miner	2	19.0	24.6	21.4	13.4	34.6	19.3	10.7	41.6	17.1
ICL	3	17.0	22.0	19.2	13.5	34.9	19.5	10.5	40.6	16.6
Maui	4	20.4	26.4	23.0	13.7	35.4	19.8	10.2	39.5	16.2
SEERLAB	5	18.8	24.3	21.2	13.1	33.9	18.9	10.1	39.0	16.0
SZTERGAK	6	14.6	18.9	16.5	12.2	31.5	17.6	9.9	38.5	15.8
WINGNUS	7	18.6	24.0	21.0	12.6	32.6	18.2	9.3	36.2	14.8
DERIUNLP	8	12.6	16.3	14.2	9.7	25.1	14.0	9.3	35.9	14.7
KX_FBK	9	13.6	17.6	15.3	10.0	25.8	14.4	8.5	32.8	13.5
BUAP	10	5.6	7.2	6.3	8.1	20.9	11.7	8.3	32.0	13.2
JU_CSE	11	12.0	15.5	13.5	8.5	22.0	12.3	7.5	29.0	11.9
UNPMC	12	7.0	9.0	7.9	7.7	19.9	11.1	7.1	27.4	11.2
DFKI	13	12.8	16.5	14.4	8.5	22.0	12.3	6.6	25.6	10.5
SJTULTLAB	14	9.6	12.4	10.8	7.8	20.2	11.3	6.2	24.0	9.9
Likey	15	11.6	15.0	13.1	7.9	20.4	11.4	5.9	22.7	9.3
UvT	16	11.4	14.7	12.9	7.6	19.6	11.0	5.8	22.5	9.2
UNICE	17	8.8	11.4	9.9	6.4	16.5	9.2	5.5	21.5	8.8
POLYU	18	3.8	4.9	4.3	4.1	10.6	5.9	4.1	16.0	6.6
UKP	19	1.6	2.1	1.8	0.9	2.3	1.3	0.8	3.1	1.3

HUMB (Supervised): Candidates are generated based on n-grams (n = 1 to 5), after removing terms with stop words and mathematical symbols. Ranking is implemented using a bagged decision tree over several features, including document structure (e.g. section and position), content (e.g. score of 2-to-5-grams using Generalized Dice Coefficient and *TF*×*IDF*), lexical/semantic scores from large term-bases (e.g. the GRISP terminological database and Wikipedia). To further improve the candidate ranking, candidates are re-ranked using a probabilistic model trained over author-assigned keyphrases in an independent collection (Lopez and Romary, 2010).

WINGNUS (Supervised): Heuristics are used to select candidates, based on occurrence in particular areas of the document, such as the title, abstract and introduction. The algorithm first identifies the key sections and headers, then extracts candidates based on POS tag sequences only in the selected areas. To rank the candidates, the system employs 19 features based on syntactic and frequency statistics such as length, *TF*×*IDF* and

Table VII. System ranking (and F-score) for each ACM classification: combined keywords

Rank	Group C	Group H	Group I	Group J
1	*HUMB* (28.3)	*HUMB* (30.2)	*HUMB* (24.2)	*HUMB* (27.4)
2	*ICL* (27.2)	*WINGNUS* (28.9)	*SEERLAB* (24.2)	*WINGNUS* (25.4)
3	*KP-Miner* (25.5)	*SEERLAB* (27.8)	*KP-Miner* (22.8)	*ICL* (25.4)
4	*SZTERGAK* (25.3)	*KP-Miner* (27.6)	*KX_FBK* (22.8)	*SZTERGAK* (25.17)
5	*WINGNUS* (24.2)	*SZTERGAK* (27.6)	*WINGNUS* (22.3)	*KP-Miner* (24.9)
6	*KX_FBK* (24.2)	*ICL* (25.5)	*SZTERGAK* (22.25)	*KX_FBK* (24.6)
7	*DERIUNLP* (23.6)	*KX_FBK* (23.9)	*ICL* (21.4)	*UNICE* (23.5)
8	*SEERLAB* (22.0)	*Maui* (23.9)	*DERIUNLP* (20.1)	*SEERLAB* (23.3)
9	*DFKI* (21.7)	*DERIUNLP* (23.6)	*DFKI* (19.3)	*DFKI* (22.2)
10	*Maui* (19.3)	*UNPMC* (22.6)	*BUAP* (18.5)	*Maui* (21.3)
11	*BUAP* (18.5)	*SJTULTLAB* (22.1)	*SJTULTLAB* (17.9)	*DERIUNLP* (20.3)
12	*JU_CSE* (18.2)	*UNICE* (21.8)	*JU_CSE* (17.9)	*BUAP* (19.7)
13	*Likey* (18.2)	*DFKI* (20.5)	*Maui* (17.6)	*JU_CSE* (18.6)
14	*SJTULTLAB* (17.7)	*BUAP* (20.2)	*UNPMC* (17.6)	*UNPMC* (17.8)
15	*UvT* (15.8)	*UvT* (20.2)	*UNICE* (14.7)	*Likey* (17.2)
16	*UNPMC* (15.2)	*Likey* (19.4)	*Likey* (11.3)	*SJTULTLAB* (16.7)
17	*UNICE* (14.3)	*JU_CSE* (17.3)	*POLYU* (13.6)	*POLYU* (14.3)
18	*POLYU* (12.5)	*POLYU* (15.8)	*UvT* (10.3)	*UvT* (12.6)
19	*UKP* (4.4)	*UKP* (5.0)	*UKP* (5.4)	*UKP* (6.8)

occurrence in the selected areas of the document (Nguyen and Luong, 2010).

KP-Miner (Unsupervised): Heuristic rules are used to extract candidates, which are then filtered to remove terms with stop words and punctuation. Further, the candidates are filtered by frequency and their position of first appearance. Finally, candidates are ranked by integrating five factors: term weight in the document D_i, term frequency in the document D_i, term IDF, a boosting factor, and term position (El-Beltagy and Rafea, 2010).

SZTERGAK (Supervised): First, irrelevant sentences are removed from the document based on their relative position in the document. Candidates are then extracted based on n-grams (up to size n=4), restricted by predefined POS patterns. To rank the candidates, the system employs a large number of features computed by analyzing the term (e.g. word length, POS pattern), the document (e.g. acronymity, collocation score for multiword terms), the corpus (e.g. section-based $TF \times IDF$, and phrasehood in the complete dataset) and external knowledge resources (e.g. Wikipedia entries/redirection) (Bernend and Farkas, 2010).

Table VIII. System ranking (and F-score) for each ACM classification: reader-assigned keywords

Rank	Group C	Group H	Group I	Group J
1	*ICL* (23.3)	*HUMB* (25.0)	*HUMB* (21.7)	*HUMB* (24.7)
2	*KX_FBK* (23.3)	*WINGNUS* (23.5)	*KX_FBK* (21.4)	*WINGNUS* (24.4)
3	*HUMB* (22.7)	*SEERLAB* (23.2)	*SEERLAB* (21.1)	*SZTERGAK* (24.4)
4	*SZTERGAK* (22.7)	*KP-Miner* (22.4)	*WINGNUS* (19.9)	*KX_FBK* (24.4)
5	*DERIUNLP* (21.5)	*SZTERGAK* (21.8)	*KP-Miner* (19.6)	*UNICE* (23.8)
6	*KP-Miner* (21.2)	*KX_FBK* (21.2)	*SZTERGAK* (19.6)	*ICL* (23.5)
7	*WINGNUS* (20.0)	*ICL* (20.1)	*ICL* (19.6)	*KP-Miner* (22.6)
8	*SEERLAB* (19.4)	*DERIUNLP* (20.1)	*DFKI* (18.5)	*SEERLAB* (22.0)
9	*DFKI* (19.4)	*DFKI* (19.5)	*SJTULTLAB* (17.6)	*DFKI* (21.7)
10	*JU_CSE* (17.0)	*SJTULTLAB* (19.5)	*DERIUNLP* (17.3)	*BUAP* (19.6)
11	*Likey* (16.4)	*UNICE* (19.2)	*JU_CSE* (16.7)	*DERIUNLP* (19.0)
12	*SJTULTLAB* (15.8)	*Maui* (18.1)	*BUAP* (16.4)	*Maui* (17.8)
13	*BUAP* (15.5)	*UNPMC* (18.1)	*UNPMC* (16.1)	*JU_CSE* (17.9)
14	*Maui* (15.2)	*Likey* (16.9)	*Maui* (14.9)	*Likey* (17.5)
15	*UNICE* (14.0)	*UvT* (16.4)	*UNICE* (14.0)	*UNPMC* (16.6)
16	*UvT* (14.0)	*POLYU* (15.5)	*POLYU* (11.9)	*SJTULTLAB* (16.3)
17	*UNPMC* (13.4)	*BUAP* (14.9)	*Likey* (10.4)	*POLYU* (13.3)
18	*POLYU* (12.5)	*JU_CSE* (12.6)	*UvT* (9.5)	*UvT* (13.0)
19	*UKP* (4.5)	*UKP* (4.3)	*UKP* (5.4)	*UKP* (6.9)

SEERLAB (Supervised): Document sections are first identified, and n-gram candidates of differing length extracted based on their occurrence in an external scholarly corpus and their frequency in different parts of the document. Finally, the system produces its final ranking of candidates using multiple decision trees with 11 features, primarily based on term frequencies, such as term frequency in section headings and document frequency, as well as heuristics such as the word length and whether the candidate is used as an acronym in the document (Treeratpituk et al., 2010).

KX_FBK (Supervised): n-gram candidates are computed similarly to *SZTERGAK*, in addition to simple statistics such as the local document frequency, and global corpus frequency. The system then ranks candidates using five features: IDF, keyphrase length, position of first occurrence, "shorter concept subsumption" and "longer concept boosting" (whereby a candidate which contains a second candidate substring receives the score of the substring) (Pianta and Tonelli, 2010).

DERIUNLP (Unsupervised): Based on the assumption that keyphrases often occur with "skill types" (important domain words that are general enough to be used in different subfields and that reflect theoretical or practical expertise e.g. *analysis, algorithm, methodology* in scientific articles), 81 skill type words were manually extracted from the corpus. Next, POS patterns that appear in phrases containing these skill type words were used to identify candidate keyphrases. To rank the candidates, the system introduces a probabilistic model based on *TF*×*IDF*, keyphrase length and term frequency in the collection (Bordea and Buitelaar, 2010).

Maui (Supervised): Maui is an open-source system developed by one of the task organizers prior to and independently of the competition (Medelyan et al., 2009). Maui's candidates are n-grams, and the keyphrase ranking is generated using bagged decision trees over features such as *TF*×*IDF*, location, phrase length, and how often a candidate was chosen as a keyphrase in the training set. The features are enhanced with statistics from Wikipedia.

DFKI (Supervised): Candidates are generated using "closed-class forms" (i.e. function words such as conjunctions and prepositions, and suffixes such as plural and tense markers) and four types of nominal groups, all within the first 2000 characters of a document. Candidate selection takes the form of an ordinal regression problem using SVM^{rank}, based on eight features including web counts, the use of special characters, and Wikipedia statistics (Eichler and Neumann, 2010).

BUAP (Unsupervised): The documents are first pre-processed to remove stop words, punctuation and abbreviations, and then the words are lemmatized and stemmed. Candidates are then selected using heuristic rules to prefer longer sequences which occur above a frequency threshold, based on the local document and the collection. Finally, the candidates are ranked using PageRank (Ortiz et al., 2010).

SJTULTLAB (Supervised): OpenNLP[8] is used to extract noun phrase chunks as candidates, which are then filtered using three heuristic rules: phrase length, frequency, and POS patterns. The candidates are then ranked using the top-30 keyphrases extracted by running KEA (Witten et al., 1999), a separate keyphrase extraction system (Wang and Li, 2010).

UNICE (Supervised): Abbreviations are first identified using ExtractAbbrev (Schwartz and Hearst, 2003), then OpenNLP is used for sentence tokenization and

[8] http://opennlp.sourceforge.net/projects.html

POS tagging. Candidates are selected based on POS patterns, and represented in a sentence–term matrix. Clustering algorithms are employed to reduce the dimensionality of the matrix, and Latent Dirichlet Allocation (LDA) is applied to identify the topics of each cluster. Finally, candidates are scored using a probabilistic metric based on the topical relatedness of candidates (Pasquier, 2010).

UNPMC (Supervised): Candidates are selected based on n-grams ($n \leq 3$) which do not contain stop words. For each candidate, the frequency within pre-defined sections of the paper (i.e. title, abstract, introduction and conclusion) is computed, as well as the number of sections it appears in. The authors empirically determine the weight of these features and then use them to rank the candidates (Park et al., 2010).

Likey (Unsupervised): First, section headings, references, figures, tables, equations, citations and punctuation are removed from the text, and all numbers are replaced with the <NUM> tag. Then, candidates are selected as those words and phrases that appear in a reference corpus based on Europarl (European Parliament plenary speeches). Finally, the system ranks candidates using document and reference corpus frequencies (Paukkeri and Honkela, 2010).

UvT (Unsupervised): First, URLs and inline references are removed from each document, and section boundaries are detected. Then, candidates are extracted using eight POS patterns. These candidates are further normalized based on lexical and morphological variation (e.g. morphological affixes and hyphenated phrases). Finally, the C-value (Frantzi et al., 2000) probabilistic measure is used to rank candidates (Zervanou, 2010).

POLYU (Unsupervised): Simplex candidates are selected based on POS tag, and scored by frequency in the title, abstract and body of the document. The top-scoring words are "core words", which are expanded into keyphrases, by appending neighboring words, based on predefined POS patterns (Ouyang et al., 2010).

7. Discussion of Results

The top-performing systems return F-scores in the upper twenties. Superficially, this number is low, and it is instructive to examine how much room there is for improvement. Keyphrase extraction is a subjective task, and an F-score of 100% is infeasible. On the author-assigned keyphrases in our test

collection, the highest a system could theoretically achieve was 81% recall[9] and 100% precision, which gives a maximum F-score of 89%. However, such a high value would only be possible if the number of keyphrases extracted per document could vary; in our task, we fixed the thresholds at 5, 10 or 15 keyphrases.

Another way of computing the upper-bound performance would be to look into how well people perform the same task. We analyzed the performance of our readers, taking the author-assigned keyphrases as the gold standard. The authors assigned an average of 4 keyphrases to each paper, whereas the readers assigned 12 on average. These 12 keyphrases cover 77.8% of the authors' keyphrases, which corresponds to a precision of 21.5%. The F-score achieved by the readers on the author-assigned keyphrases is 33.6%, whereas the F-score of the best-performing system on the same data is 19.3% (for top-15, not top-12 keyphrases, see Table VI).

Reviewing the techniques employed by the 15 submitted systems revealed interesting trends in the different stages of keyphrase extraction: candidate identification, feature engineering and candidate ranking. In the candidate identification step, most systems used either n-grams or POS-based regular expressions, or both. Additionally, there is a clear tendency to apply pre-processing prior to the candidate identification step. For example, dealing with abbreviations seems to be an important step for improving candidate coverage, specifically aimed at scientific papers. Also, filtering candidates by frequency and location in different sections of the document was broadly employed among the participating systems. The majority of systems which used section information found the boundaries with heuristic approaches over the provided text dump, while *HUMB* and *WINGNUS* performed section boundary detection over the original PDF files.

In ranking the candidates, the systems applied a variety of features: lexical, structural and statistical. It is particularly interesting that many systems used external information, such as Wikipedia and external corpora. On the other hand, none of systems made use of the 4 ACM document classifications that the test and training documents were grouped into. Table IX describes the features used by each system, as described in the system description paper.

To rank the candidates, supervised systems used learners such as maximum entropy, naïve Bayes and bagged decision trees, all of which are popular approaches for keyphrase extraction. Another approach used for ranking was a learn-to-rank classifier based on SVM^{rank}. Unsupervised systems tended to propose a novel probabilistic model to score candidates, mostly based on simple multiplication of feature values, but also including PageRank and topic modeling. It is difficult to gauge the relative superiority of different ma-

[9] The remaining 19% of keyphrases do not actually appear in the documents and thus cannot be extracted.

Table IX. The participating systems, ordered by overall rank, with the different feature types used by each system (broken down into Token Scoring, Lexical/Syntactic, Sem(antic), External and Format).

Feature type	**Token Scoring**						**Lexical/Syntactic**						**Sem**		**External**			**Format**	
	TF, IDF, TF×IDF and variants	Structural Information	First and/or last occurrence	Term Length	Maximal frequent sequences	Suffix	POS sequences	Acronyms	Lexical associations via statistics tests	Special characters, fonts	Parsing	Average token count	"Keyphraseness"	Shorter/Longer concept subsumption	Wikipedia document occurrence, links/redirects, IDF	Web counts	Reference corpus (DBLP, GRISP, Europarl)	Text-based section analysis	PDF parsing-based section analysis
HUMB	x	x	x						x				x				x		x
WINGNUS	x	x	x	x						x									x
KP-Miner	x		x																
SZTERGAK	x		x	x		x	x	x	x		x		x		x			x	
SEERLAB	x	x		x				x									x	x	
KX_FBK	x		x	x										x					
DERIUNLP	x																		
Maui	x		x	x									x						
DFKI	x	x	x						x	x		x			x	x			
BUAP					x														
SJTULTLAB	x						x												
UNICE							x	x											
UNPMC		x		x									x					x	
Likey	x																		
UvT									x										
POLYU	x				x														

chine learning approaches over the task, as they were combined with different candidate selection techniques and feature sets. However, the standardized evaluation on the common training and test data does uncover some trends: namely that document structure and IR-style term weighting approaches appear to be effective across the board. There is no doubt, however, that there is definitely still room for improvement on the task, and we look forward to seeing the dataset used in future experimentation on keyphrase extraction.

For any future shared task on keyphrase extraction, we recommend against fixing a system threshold on the number of keyphrases to be extracted per document. Finally, as we use a strict exact matching metric for evaluation, the presented evaluation figures are likely underestimations of actual system performance, as many semantically-equivalent keyphrases are not counted as correct. For future runs of this challenge, we believe a more semantically-motivated evaluation should be employed to give a more accurate impression of keyphrase acceptability.

8. Conclusion

We describe Task 5 of the Workshop on Semantic Evaluation 2010 (SemEval-2010), focusing on keyphrase extraction. We provided an overview of the keyphrase extraction process and related work in this area. We outlined the design of the datasets used in the shared task and the evaluation metrics, before presenting the official results for the task and summarizing the immediate findings. We also analyzed the upper-bound performance for this task, and demonstrated that there is still room for improvement on the task. We look forward to future advances in automatic keyphrase extraction based on this and other datasets.

Acknowledgements

This work was supported by National Research Foundation grant "Interactive Media Search" (grant # R-252-000-325-279) for Min-Yen Kan, and ARC Discovery grant no. DP110101934 for Timothy Baldwin.

References

Barker, K. and N. Corrnacchia: 2000, 'Using noun phrase heads to extract document keyphrases'. In: *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*. Montreal, Canada, pp. 40–52.

Barzilay, R. and M. Elhadad: 1997, 'Using lexical chains for text summarization'. In: *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain, pp. 10–17.

Bernend, G. and R. Farkas: 2010, 'SZTERGAK : Feature Engineering for Keyphrase Extraction'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 186–189.

Bordea, G. and P. Buitelaar: 2010, 'DERIUNLP: A Context Based Approach to Automatic Keyphrase Extraction'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 146–149.

D'Avanzo, E. and B. Magnini: 2005, 'A Keyphrase-Based Approach to Summarization: the LAKE System'. In: *Proceedings of the 2005 Document Understanding Workshop (DUC 2005)*. Vancouver, Canada, pp. 6–8.

Eichler, K. and G. Neumann: 2010, 'DFKI KeyWE: Ranking Keyphrases Extracted from Scientific Articles'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 150–153.

El-Beltagy, S. R. and A. Rafea: 2010, 'KP-Miner: Participation in SemEval-2'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 190–193.

Ercan, G.: 2006, 'Automated Text Summarization and Keyphrase Extraction'. Master's thesis, Bilkent University.

Frank, E., G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning: 1999, 'Domain Specific Keyphrase Extraction'. In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*. Stockholm, Sweden, pp. 668–673.

Frantzi, K., S. Ananiadou, and H. Mima: 2000, 'Automatic recognition of multi-word terms'. *International Journal of Digital Libraries* **3**(2), 117–132.

Gong, Z. and Q. Liu: 2008, 'Improving keyword based web image search with visual feature distribution and term expansion'. *Knowledge and Information Systems* **21**(1), 113–132.

Gutwin, C., G. Paynter, I. Witten, C. Nevill-Manning, and E. Frank: 1999, 'Improving browsing in digital libraries with keyphrase indexes'. *Journal of Decision Support Systems* **27**, 81–104.

Hammouda, K. M., D. N. Matute, and M. S. Kamel: 2005, 'CorePhrase: Keyphrase Extraction for Document Clustering'. In: *Proceedings of the 4th International Conference on Machine Learning and Data Mining (MLDM 2005)*. Leipzig, Germany, pp. 265–274.

Hulth, A.: 2003, 'Improved automatic keyword extraction given more linguistic knowledge'. In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan, pp. 216–223.

Hulth, A.: 2004, 'Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction'. Ph.D. thesis, Stockholm University.

Hulth, A. and B. B. Megyesi: 2006, 'A study on automatically extracted keywords in text categorization'. In: *Proceedings of 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia, pp. 537–544.

Jarmasz, M. and C. Barriere: 2004, 'Keyphrase Extraction: Enhancing Lists'. In: *Proceedings of the 2nd Conference on Computational Linguistics in the North-East*. Montreal, Canada. `http://arxiv.org/abs/1204.0255`.

Jarvelin, K. and J. Kekalainen: 2002, 'Cumulated Gain-based Evaluation of IR techniques'. *ACM Transactions on Information Systems* **20**(4).

Kim, S. N., T. Baldwin, and M.-Y. Kan: 2009, 'The Use of Topic Representative Words in Text Categorization'. In: *Proceedings of the Fourteenth Australasian Document Computing Symposium (ADCS 2009)*. Sydney, Australia, pp. 75–81.

Kim, S. N., T. Baldwin, and M.-Y. Kan: 2010, 'Evaluating N-gram based Evaluation Metrics for Automatic Keyphrase Extraction'. In: *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*. Beijing, China, pp. 572–580.

Kim, S. N. and M.-Y. Kan: 2009, 'Re-examining Automatic Keyphrase Extraction Approach in Scientific Articles'. In: *Proceedings of the ACL/IJCNLP 2009 Workshop on Multiword Expressions*. Singapore, pp. 7–16.

Krapivin, M., A. Autayeu, and M. Marchese: 2009, 'Large Dataset for Keyphrases Extraction'. Technical Report DISI-09-055, DISI, University of Trento, Italy.

Krapivin, M., A. Autayeu, M. Marchese, E. Blanzieri, and N. Segata: 2010, 'Improving Machine Learning Approaches for Keyphrases Extraction from Scientific Documents with

Natural Language Knowledge'. In: *Proceedings of the Joint JCDL/ICADL International Digital Libraries Conference*. Gold Coast, Australia, pp. 102–111.

Lawrie, D., W. B. Croft, and A. Rosenberg: 2001, 'Finding Topic Words for Hierarchical Summarization'. In: *Proceedings of SIGIR 2001*. New Orleans, USA, pp. 349–357.

Li, D., S. Li, and W. Li: 2010, 'A Semi-supervised key phrase extraction approach: learning from title phrases through a document semantic network'. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, pp. 296–300.

Litvak, M. and M. Last: 2008, 'Graph-based keyword extraction for single-document summarization'. In: *Proceedings of the 2nd Workshop on Multi-source Multilingual Information Extraction and Summarization*. Manchester, UK, pp. 17–24.

Liu, F., D. Pennell, F. Liu, and Y. Liu: 2009a, 'Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts'. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, USA, pp. 620–628.

Liu, Z., P. Li, Y. Zheng, and S. Maosong: 2009b, 'Clustering to Find Exemplar Terms for Keyphrase Extraction'. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore, pp. 257–266.

Lopez, P. and L. Romary: 2010, 'HUMB: Automatic Key Term Extraction from Scientific Articles in GROBID'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 248–251.

Matsuo, Y. and M. Ishizuka: 2004, 'Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information'. *International Journal on Artificial Intelligence Tools* **13**(1), 157–169.

Medelyan, O., E. Frank, and I. H. Witten: 2009, 'Human-competitive tagging using automatic keyphrase extraction'. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore, pp. 1318–1327.

Medelyan, O. and I. Witten: 2006, 'Thesaurus based automatic keyphrase indexing'. In: *Proceedings of the 6th ACM/IEED-CS joint conference on Digital libraries*. pp. 296–297.

Mihalcea, R. and E. Faruque: 2004, 'SenseLearner: Minimally supervised word sense disambiguation for all words in open text'. In: *Proceedings of the ACL/SIGLEX Senseval-3 Workshop*. Barcelona, Spain, pp. 155–158.

Mihalcea, R. and P. Tarau: 2004, 'TextRank: Bringing Order into Texts'. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain.

Nguyen, T. D. and M.-Y. Kan: 2007, 'Key phrase Extraction in Scientific Publications'. In: *Proceeding of International Conference on Asian Digital Libraries*. Hanoi, Vietnam, pp. 317–326.

Nguyen, T. D. and M.-T. Luong: 2010, 'WINGNUS: Keyphrase Extraction Utilizing Document Logical Structure'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 166–169.

Ortiz, R., D. Pinto, M. Tovar, and H. Jiménez-Salazar: 2010, 'BUAP: An Unsupervised Approach to Automatic Keyphrase Extraction from Scientific Articles'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 174–177.

Ouyang, Y., W. Li, and R. Zhang: 2010, '273. Task 5. Keyphrase Extraction Based on Core Word Identification and Word Expansion'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 142–145.

Park, J., J. G. Lee, and B. Daille: 2010, 'UNPMC: Naive Approach to Extract Keyphrases from Scientific Articles'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 178–181.

Pasquier, C.: 2010, 'Single Document Keyphrase Extraction Using Sentence Clustering and Latent Dirichlet Allocation'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 154–157.

Paukkeri, M.-S. and T. Honkela: 2010, 'Likey: Unsupervised Language-Independent Keyphrase Extraction'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 162–165.

Paukkeri, M.-S., I. T. Nieminen, M. Polla, and T. Honkela: 2008, 'A Language-Independent Approach to Keyphrase Extraction and Evaluation'. In: *Proceedings of the 22nd International Conference on Computational Linguistics*. Manchester, UK, pp. 83–86.

Pianta, E. and S. Tonelli: 2010, 'KX: A Flexible System for Keyphrase eXtraction'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 170–173.

Schutz, A. T.: 2008, 'Keyphrase Extraction from Single Documents in the Open Domain Exploiting Linguistic and Statistical Methods'. Master's thesis, National University of Ireland.

Schwartz, A. S. and M. A. Hearst: 2003, 'A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text'. In: *Proceedings of the Pacific Symposium on Biocomputing*, Vol. 8. pp. 451–462.

Tomokiyo, T. and M. Hurst: 2003, 'A Language Model Approach to Keyphrase Extraction'. In: *Proceedings of ACL Workshop on Multiword Expressions*. Sapporo, Japan, pp. 33–40.

Treeratpituk, P., P. Teregowda, J. Huang, and C. L. Giles: 2010, 'SEERLAB: A System for Extracting Keyphrases from Scholarly Documents'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 182–185.

Turney, P.: 1999, 'Learning to Extract Keyphrases from Text'. National Research Council, Institute for Information Technology, Technical Report ERB-1057. (NRC #41622).

Turney, P.: 2003, 'Coherent keyphrase extraction via Web mining'. In: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Acapulco, Mexico, pp. 434–439.

Wan, X. and J. Xiao: 2008, 'CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction'. In: *Proceedings of 22nd International Conference on Computational Linguistics*. Manchester, UK, pp. 969–976.

Wang, C., M. Zhang, L. Ru, and S. Ma: 2008, 'An automatic online news topic keyphrase extraction system'. In: *Proceedings of 2008 IEEE/WIC/ACM International Conference on Web Intelligence*. Sydney, Australia, pp. 214–219.

Wang, L. and F. Li: 2010, 'SJTULTLAB: Chunk Based Method for Keyphrase Extraction'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 158–161.

Witten, I., G. Paynter, E. Frank, C. Gutwin, and G. Nevill-Manning: 1999, 'KEA: Practical Automatic Key phrase Extraction'. In: *Proceedings of the Fourth ACM Conference on Digital libraries*. Berkeley, USA, pp. 254–255.

Zervanou, K.: 2010, 'UvT: The UvT Term Extraction System in the Keyphrase Extraction Task'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 194–197.

Zesch, T. and I. Gurevych: 2009, 'Approximate Matching for Evaluating Keyphrase Extraction'. In: *Proceedings of RANLP 2009 (Recent Advances in Natural Language Processing)*. Borovets, Bulgaria, pp. 484–489.

Zhang, Y., N. Zincir-Heywood, and E. Milios: 2004, 'Term based Clustering and Summarization of Web Page Collections'. In: *Proceedings of the 17th Conference of the Canadian Society for Computational Studies of Intelligence*. London, Canada, pp. 60–74.