Approximate Matching for Evaluating Keyphrase Extraction

Torsten Zesch and Iryna Gurevych
Ubiquitous Knowledge Processing Lab
Computer Science Department
Technische Universität Darmstadt, D-64289 Darmstadt, Germany
http://www.ukp.tu-darmstadt.de

Abstract

We propose a new evaluation strategy for keyphrase extraction based on approximate keyphrase matching. It corresponds well with human judgments and is better suited to assess the performance of keyphrase extraction approaches. Additionally, we propose a generalized framework for comprehensive analysis of keyphrase extraction that subsumes most existing approaches, which allows for fair testing conditions. For the first time, we compare the results of state-of-the-art unsupervised and supervised keyphrase extraction approaches on three evaluation datasets and show that the relative performance of the approaches heavily depends on the evaluation metric as well as on the properties of the evaluation dataset.

Keywords

keyphrase extraction; approximate matching

1 Introduction

Keyphrases are small sets of expressions representing a document's content. **Keyphrase extraction** is the task of automatically extracting such keyphrases from a document. The extracted phrases have to be present in the document itself, in contrast to keyphrase assignment (a multi-class text classification problem) where a fixed set of keyphrases is used which are not necessarily contained in the document. Keyphrase extraction has important applications in NLP including summarization [4, 11], clustering [9], as well as indexing and browsing [8], highlighting [22] and searching [2].

Despite the importance of the task, the evaluation of keyphrase extraction has not received much research attention in the past. In this paper, we address three core problems with the evaluation of keyphrase extraction: (i) the evaluation metric, (ii) the evaluation datasets, and (iii) the evaluation framework.

The performance of most keyphrase extraction algorithms is evaluated by comparing whether the extracted keyphrases exactly match the human assigned gold standard keyphrases. However, this is known to underestimate performance [22]. Allowing only exact matchings cannot account for variations in the extracted keyphrases that might be perfectly acceptable when presented to humans. For example, longer noun phrases like "congress party spokesman" are usually more specific and thus more informative to the reader than shorter noun phrases like "congress party". However, due to reading and writing economy, specific terms are usually not often repeated in a document [1]. Thus, longer noun phrases are unlikely to be annotated by human annotators, preventing exact matching. To compensate for these shortcomings, we propose a new approximate matching strategy that also accounts for non-exact matches and is able to give a better picture of the actual quality of a keyphrase extraction algorithm. We evaluate the validity of the new matching strategy in a human annotation study in Section 3.

The lack of standard datasets is the second problem tackled in this paper. Comparing results from different papers is difficult as no standard datasets are used, and few papers have compared their results on more than one dataset with different competing systems. Thus it cannot be judged conclusively which approaches improve results on which kind of dataset. We collected three publicly available datasets with different properties, which allows comparison of the applicability of keyphrase extraction algorithms to those datasets.

Some datasets contain annotated keyphrases that actually cannot be found in the document. This has serious implications on the comparability of results, as including them in the evaluation might significantly lower the reachable performance on the dataset. A way to solve this problem is to use a unified framework for the evaluation of keyphrase extraction. This also prevents influence from varying pre- and postprocessing. Thus, we propose a generalized framework for keyphrase extraction, which allows for fair testing conditions and a comprehensive analysis of results.

2 Related Work

In this section, we give an overview of (i) existing approaches to keyphrase extraction, (ii) the different ways to evaluate keyphrase extraction, and (iii) the datasets that have been used for evaluation.

Keyphrase Extraction Approaches Existing methods for keyphrase extraction can be categorized into supervised and unsupervised approaches.[1]

[1] Note that unsupervised approaches might use tools like NP chunkers relying on supervised approaches. However, as such

Closely related to keyphrase extraction are glossary extraction [17] and back-of-the-book indexing [3].

Unsupervised approaches usually select quite general sets of candidates (e.g. all tokens in a document), and use a subsequent ranking step to limit the selection to the most important candidates. For example, Barker and Cornacchia [1] restrict candidates to noun phrases, and rank them using heuristics based on length, term frequency, and head noun frequency. Bracewell et al. [2] also restrict candidates to noun phrases, and cluster them if they share a term. The clusters are ranked according to the noun phrase and token frequencies in the document. Finally, the centroids of the top-n ranked clusters are selected as keyphrases. Mihalcea and Tarau [15] propose a graph-based approach called *TextRank*, where the graph nodes are tokens and the edges reflect co-occurrence relations between tokens in the document. The nodes are ranked using PageRank, and longer keyphrases can be reconstructed in a post-processing step merging adjacent keywords. The method was found to yield competitive results with state-of-the-art supervised systems [15]. Wan and Xiao [24] expand TextRank by augmenting the graph with highly similar documents, which improves results compared with standard TextRank and a tf.idf baseline.

Another branch of unsupervised approaches is based on statistical analysis. Tomokiyo and Hurst [21] use pointwise KL-divergence between language models derived from the documents and a reference corpus. Paukkeri et al. [18] use a similar method based on likelihood ratios. Matsuo and Ishizuka [12] present a statistical keyphrase extraction approach that does not make use of a reference corpus, but is based on co-occurrences of terms in a single document.

Supervised approaches use a corpus of training data to learn a keyphrase extraction model that is able to classify candidates as keyphrases. A well known supervised system is *Kea* [6] that uses all n-grams of a certain length as candidates, and ranks them using the probability of being a keyphrase. *Kea* is based on a Naïve Bayes classifier using *tf.idf* and *position* as its main features. *Extractor* [22] is another supervised system that uses stems and stemmed n-grams as candidates. Its features are tuned using a genetic algorithm. *Kea* and *Extractor* are known to achieve roughly the same level of performance [23]. Hulth [10] uses a combination of lexical and syntactic features adding more linguistic knowledge which outperforms *Kea*. Medelyan and Witten [13] present the improved *Kea++* that selects candidates with reference to a controlled vocabulary from a thesaurus or Wikipedia [14]. Turney [23] augments *Kea* with a feature set based on statistical word association to ensure that the returned keyphrase set is coherent. However, this assumption might not hold if a document covers different topics. Nguyen and Kan [16] augment *Kea* with features tailored towards scientific publications such as section information and certain morphological phenomena often found in scientific papers.

Evaluation Methods The prevalent approaches for evaluating keyphrase extraction algorithms are: (i) *manual evaluation based on human judges* [1, 12, 22], (ii) *application-based evaluation* [2, 11], and (iii) *automated evaluation against human assigned keyphrases* [6, 10, 15, 16, 23].

In **manual evaluation**, human judges can easily decide whether the returned keyphrases are good representatives of a document's content or not. Thus, manual evaluation is not restricted to exact matches between gold standard keyphrases and keyphrases returned by a method. However, manual evaluation of extracted keyphrases is very costly and time-consuming. In particular, it is not suited for any kind of parameter tuning, as the output of each new system configuration involves manual re-evaluation.

An **application-based evaluation** utilizes keyphrases as part of a usually complex application, and the performance is measured in terms of the overall performance of the application. However, this entails influence of parameters besides the keyphrase extraction algorithm to be tested. For example, Bracewell et al. [2] use the information retrieval task of keyword search to determine the effectiveness of keywords at uniquely describing the document from which they were extracted. However, this method might extract keyphrase sets that are good indicators for relevant documents, but that are not acceptable when presented to humans. Litvak and Last [11] use a summary-based evaluation, where a term is used as a gold standard keyphrase if it appears in the document and in the summary.

Automated evaluation against human assigned keyphrases relies on automated matching of human annotated gold standard keyphrases with the keyphrases extracted by a certain approach. The human assigned keyphrases are either derived keyphrases assigned by authors [6, 22], or are annotated by indexers [10, 16, 24]. As this approach avoids the problems of manual evaluation (costly, time-consuming, difficult algorithm tuning), and of application-based evaluation (influence of complex applications, keyphrases unacceptable to humans), we are going to use it for evaluation in this paper.

Datasets We now describe three publicly available datasets with manually annotated gold standard keyphrases. They differ in length and domain (see Table 1), and can thus be used to assess different properties of keyphrase extraction algorithms.

The **Inspec dataset** [10] contains 2000 abstracts of journals in the Inspec database from the years 1998 to 2002. There are two sets of keyphrases assigned by professional indexers: controlled terms (restricted to the Inspec index terms, and useful for keyphrase assignment) and uncontrolled terms. Some uncontrolled terms (23.8%) are not directly found in the documents and therefore ignored in our evaluation. However, this dataset has the highest number of human assigned keyphrases per document, while the documents are rather short with an average length of ≈ 140 tokens. The Pearson correlation between the length of the document and the number of human assigned keyphrases is quite high ($r = 0.56$), indicating that indexers often

tools are usually already available for most languages, we consider an approach to be unsupervised if it does not make use of any training data with annotated *keyphrases*.

Name	Reference	Domain	Indexing	# Docs	∅ # Tokens	∅ # Keyphrases	r
Inspec	Hulth (2004)	Scientific	Single Indexer	2000	138.6	9.64	0.56
DUC	Wan and Xiao (2008)	News	Multiple Indexers	301	902.8	8.08	0.18
SP	Nguyen and Kan (2007)	Scientific	Multiple Indexers	134	8491.6	8.31	0.08

Table 1: *Keyphrase evaluation datasets. r is the Pearson correlation between the document length and the number of assigned keyphrases.*

exhaustively annotated keyphrases in the documents. Thus, it should be relatively easy to extract keyphrases from the documents, and we expect the performance on this dataset to be higher than on the other datasets.

The **DUC dataset** [24] consists of 308 documents from DUC2001 that were manually annotated with at most 10 keyphrases per document by two indexers. Annotation conflicts between the indexers were solved by discussion. Two documents in the DUC2001 data obtained from NIST were empty, and 5 documents had no annotated keyphrases. Thus, the final dataset used in this paper contains 301 documents.

The **SP dataset** [16] originally contains 211 scientific publications downloaded from the internet and automatically converted to plain text. Keyphrases were manually annotated by multiple indexers, but conflicts were not resolved. We removed documents for which no keyphrase annotation was available, and those with multiple conflicting annotations. The final dataset contains 134 documents.

3 Automated Evaluation

We now give an overview of the automated evaluation as introduced in the previous section. It relies on matching a set of human annotated gold standard keyphrases K_{gold} with a ranked list of keyphrases K_{ext} extracted by a certain approach. We define a matching m between a gold standard keyphrase $k_{gold} \in K_{gold}$ and an extracted keyphrase $k_{ext} \in K_{ext}$ to be a tuple $m = (k_{gold}, k_{ext})$. The matching can either be true or false, depending on whether k_{gold} and k_{ext} are equivalent according to the matching strategy. Previous works used exact matching (Exact) that requires k_{gold} and k_{ext} to have exactly the same string representation, i.e. $\textsc{Exact}(k_{gold}, k_{ext}) = \text{true} \Leftrightarrow k_{gold} = k_{ext}$.

To evaluate the overall performance of a keyphrase extraction system, we do not need to look at single matchings m, but at the full list of matchings M. Previous studies used Precision (P), Recall (R), and F-measure (F_1) at a certain fixed cutoff value, e.g. after the first 10 retrieved keyphrase matchings. However, if documents have varying numbers of keyphrases assigned (which is the case for all datasets presented in Section 2), a cutoff will distort results. For example, if we always extract 10 keyphrases, but a document only has 8 gold keyphrases assigned, then 2 extracted keyphrases will always be wrong. Thus, we propose to use the **R-precision (R-p)** measure from information retrieval [19] to evaluate keyphrase extraction systems. In information retrieval, R-p is defined as the precision when the number fo retrieved documents equals the number of relevant documents in the document collection. Hence, for keyphrase extraction we define R-p as the precision when the number of retrieved keyphrase matchings equals the number of gold standard keyphrases assigned to the document. An R-precision of 1.0 is equivalent to perfect keyphrase ranking and perfect recall.

These properties make R-p a favorable metric for keyphrase extraction, as it puts a focus on the precision on the first ranks, which is necessary for most practical systems that assign or present only a handful of keyphrases. R-p also measures whether the keyphrases on the first ranks cover the whole set of topics in the document. For example, a keyphrase extraction approach that extracts a lot of variants (e.g. “scheduling”, “real-time scheduling”, “embedded real-time scheduling”) on the first ranks will have a lower precision than an approach that covers more topics. As an additional benefit, R-p is a single number metric allowing for more compact presentation of results and easier comparison.

We formally define R-p as the precision when $|M| = |K_{gold}|$. Precision is computed as $\frac{M_c}{M}$, where M_c is the list of correct matchings in M.

3.1 Approximate Matching Strategy

The exact matching strategy Exact is only partially indicative of the performance of a keyphrase extraction method, as it is known to underestimate performance as perceived by human judges [22]. Additionally, it may not be a good indicator of the overall quality of the extracted set of keyphrases, as there are many cases in which exact matching fails, e.g. lexical semantic variations (*automobile sales*, *car sales*), overlapping phrases (*scheduling*, *real-time scheduling*), or morphological variants like plurals (*performance metric*, *performance metrics*).[2] Thus, we propose a new approximate matching strategy $\textsc{Approx}(k_{gold}, k_{ext})$ that accounts for morphological variants (Morph) and the two cases of overlapping phrases: either the extracted keyphrase includes the gold standard keyphrase (Includes) or the extracted keyphrase is a part of the gold standard keyphrase (PartOf). Exact matchings are of course still valid in addition to approximate matchings.

For overlapping phrases, we do not allow character level variations, but only token level variations, i.e. the Includes category contains matchings where the extracted keyphrase contains all the tokens in the gold keyphrase plus some additional tokens. In the case of the morphological variants Morph, we limit approximate matching to the detection of plurals. We leave the inclusion of other morphological variations and lexical semantic variants to future work.

[2] In the remainder of this paper, we present examples of matchings as (*gold keyphrase*, *extracted keyphrase*).

		Judges accepting matchings	
	#	4	≥ 3
INCLUDES	274	.58	.80
PARTOF	239	.31	.44
MORPH	53	.96	.96
MORPH+INCLUDES	327	.65	.83

Table 2: *Ratio of approximate keyphrase matchings acceptable to human judges (4 = all judges; ≥ 3 = at least 3 out of 4 judges).*

3.2 Approximate Matching Evaluation

For testing whether the approximate matching strategy is acceptable to humans, we randomly selected a maximum of 300 non-exact matchings from each of the three datasets (yielding a maximum of 900 randomly selected matchings). We included matchings from each of the 3 approximate matching categories (INCLUDES, PARTOF, and MORPH) using different candidate selection methods and length restrictions to account for all kinds of keyphrase variants. The total number of selected approximate matchings is 566, as some matchings were included in multiple sets of the random matchings and morphological approximate matching MORPH did not always account for 100 approximate matchings per dataset.

We had four judges annotate whether it would be acceptable to replace the gold standard keyphrase with the extracted keyphrase using the approximate matching strategy. As no context was given when judging about a matching, annotators were instructed to annotate a pair as invalid if in doubt. Thus, the annotation has a pessimistic bias and rather underestimates human agreement with the approximate matching. The results of the study are presented in Table 2.

In the MORPH category of morphological variants, agreement between judges was very high: 96% of all MORPH matchings were acceptable to all 4 judges. The only problematic case were two abbreviations (*fms*, *fmss*) and (*soa*, *soas*) where the judges could not decide about the validity without looking at the context. Agreement between all 4 judges is considerably lower for INCLUDES and PARTOF. However, given the inherent subjectivity of the task, we treat an agreement of 3 out of 4 judges as valid for accepting a match. In the INCLUDES category agreement reaches 80%, while for the PARTOF category it is only 44%.

The major source of error in the INCLUDES category was wrong pre-processing. For example, the matching (*security level*, *give security level*) was unanimously rejected by all judges, as the extracted keyphrase contains a chunking error.

A major source of error in the PARTOF category were cases when the extracted keyphrase is too general compared to the gold keyphrase, e.g. (*topic importance*, *topic*). A potential refinement of the PARTOF heuristic would be to match only extracted keyphrases whose head noun matches the head of the gold keyphrase. However, only 52% of such cases (66 out of 128) were accepted by at least 3 judges. Furthermore, in 35% of the cases (39 out of 111) a matching with a non-matching head like (*tubercolosis cases*, *tubercolosis*) was accepted by at least 3 judges. This means, that neither is a matching head required for a keyphrase to be acceptable to human judges, nor is a matching head sufficient for an acceptable match. As we aim for an approximate matching with high precision, we decided not to use the PARTOF category due to these problems, but combined MORPH and INCLUDES to an approximate matching strategy[3] with a human agreement of 83%.

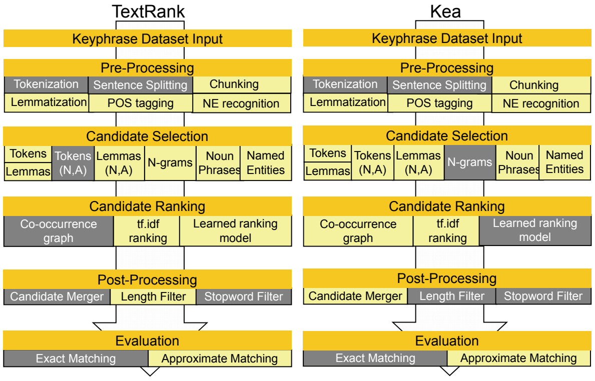

Fig. 1: *State-of-the-art keyphrase extraction systems represented in our framework.*

4 Extraction Framework

Most automatic keyphrase extraction methods have two stages: first they select a list of keyphrase candidates that is then ranked according to some measure of keyphrase importance. To allow for a fair comparison, the same pre- and postprocessing is necessary, as well as exactly the same evaluation strategy. We propose a generalized framework for the comprehensive analysis of keyphrase extraction as shown in Figure 1. It was designed to be as language-independent as possible using either no language dependent information at all, or components that are already available for most languages (like tokenizers or chunkers). The preprocessing pipeline is based on the DKPro UIMA component repository [7].

Pre-Processing and Candidate Selection For preprocessing, we tokenize the documents, and split them into sentences. We integrated the TreeTagger for lemmatization, POS-tagging, and NP chunking [20], as well as the Stanford NER tool [5] for named entity recognition. From this pool of preprocessed data, we select as candidates *Tokens*, *Lemmas*, *N-grams*, *Noun Phrases*, and *Named Entities*. Following [15], we additionally use the restricted set of tokens *Tokens (N,A)* and lemmas *Lemmas (N,A)*.

Candidate Ranking The unsupervised graph-based methods (e.g. *TextRank*) build a co-occurence graph using the candidates. The final candidate ranking is determined by computing the centrality scores of the graph nodes using PageRank. For tf.idf ranking,

[3] It is formally defined as: $\text{APPROX}(k_{gold}, k_{ext}) = \text{EXACT} \vee \text{MORPH} \vee \text{INCLUDES}$.

		Inspec		DUC		SP	
	Candidates	R-p$_{ex}$	R-p$_{ap}$	R-p$_{ex}$	R-p$_{ap}$	R-p$_{ex}$	R-p$_{ap}$
KEA	N-grams	.16	.19	.11	.14	**.21**	**.25**
TextRank	Token N,A	**.31**	**.36**	.21	.23	.04	.10
tf.idf	Tokens	.11	.22	.05	.12	.06	.18
	Tokens (N,A)	.27	**.32**	**.12**	.15	.12	**.22**
	Lemmas	.15	.27	.06	.14	.07	.21
	Lemmas (N,A)	**.28**	**.32**	**.12**	**.16**	**.13**	**.22**
	N-grams	.10	.16	.03	.06	.06	.15
	Noun Phrases	.27	.32	**.12**	.14	.10	.21
	Named Entities	.01	.01	.11	.13	.06	.08
co-occ	Tokens	.06	.22	.00	.07	.00	.05
	Tokens (N,A)	**.31**	**.36**	.21	.23	.04	.10
	Lemmas	.07	.22	.00	.06	.00	.06
	Lemmas (N,A)	.29	.35	**.22**	**.24**	.08	.15
	N-grams	.07	.22	.03	.10	.01	.09
	Noun Phrases	.28	.34	.12	.14	**.12**	**.18**
	Named Entities	.01	.01	.09	.09	.04	.05

Table 3: *Keyphrase extraction results in terms of R-precision using exact matching (R-p$_{ex}$) and approximate matching (R-p$_{ap}$).*

the tf.idf scores are computed using token frequencies. If candidates contain more than one token, the overall tf.idf score for the candidate is the maximum tf.idf score among all the contained tokens. The supervised keyphrase extraction systems use the extraction model obtained from the training data to classify the candidates into keyphrases and rank them according to their importance in the document.

Postprocessing and Evaluation We merge candidates that are adjacent in the source document to reconstruct longer keyphrases from short candidates like *Tokens* or *Lemmas*. However, to ensure a fair comparison, we apply merging to all configurations of our keyphrase extraction framework, because also approaches with higher quality candidates like *Noun Phrases* can benefit from merging.

We use an additional post-filtering step to remove candidates or keyphrases that do not conform to length restrictions. When analyzing the length of the gold standard keyphrases in the training set, we found that - depending on the dataset - 97.7 to 99.2% of all keyphrases in the training data contain 1 to 4 tokens. Thus, we limited the length of returned keyphrases to 1 to 4 tokens.

We remove trailing stopwords from candidates, but keep stopwords that appear inside a keyphrase.[4] We also remove keyphrases that exactly match a stopword. Finally, the post-processed list of ranked keyphrases is used to compute the R-precision scores for each of the keyphrase extraction systems.

5 Experiments and Results

For our comprehensive analysis, we selected *Kea* [6] as the most widely used supervised system, and *TextRank* [15] as a state-of-the-art unsupervised system. The only external component used is the *Kea* ranking model. *TextRank* was fully modelled in our framework. We applied exactly the same pre- and post-processing to all experimental configurations.

We set aside two thirds of the documents in each dataset for training, while the rest of the data is used for evaluation.[5] We compare *Kea* and *TextRank* with all possible combinations of the candidate selection strategies and the ranking methods (tf.idf ranking as well co-occurrence graph based ranking abbreviated as "co-occ"). For comparison of the exact matching and the approximate matching strategy, we computed R-precision for exact matching (R-p$_{ex}$) and approximate matching (R-p$_{ap}$). Table 3 gives an overview of the obtained results.[6]

Theoretically, *Kea* as a supervised system is expected to yield the best performance. Tf.idf ranking based methods (that do not use any training data, but use information drawn from the whole document collection) are supposed to perform worse than supervised systems, but better than co-occurrence graph based methods like *TextRank* that only use information from a single document. However, under the controlled conditions of our keyphrase extraction framework, the unsupervised *TextRank* outperforms *Kea* by a wide margin on the Inspec and on the DUC dataset. Both datasets contain only rather small documents ($\approx$ 100–1000 tokens), making it relatively easy to select the right keyphrases.

On the SP dataset containing the longer documents, *Kea* outperforms all co-occurrence or tf.idf based system configurations by a wide margin when using exact matching. However, the approximate matching strategy reveals that the performance gap between *Kea* and the best configuration using tf.idf ranking with *Lemma (N,A)* candidates is not as large as exact matching indicates (dropping from .08 to .03).

The wide range of candidates tested within our framework allows to draw other interesting conclu-

[4] For example, we keep "United States of America" as the stopword appears inside a keyphrase, while "the weak economy" is pruned to "weak economy" as the stopword occurs at the boundary of the candidate.

[5] Note that all keyphrase extraction methods used in that paper except *Kea* did not make use of that training data. However, as we wanted to ensure a fair comparison, we tested all keyphrase extraction systems on the same evaluation data.

[6] Note that in our framework, the *TextRank* system is equivalent to using *Token (N,A)* as the candidate selection strategy and using co-occurrence graph based ranking. We duplicated this row of results as 'TextRank' for convenience.

sions: The candidate selection strategies *Tokens*, *Lemmas*, and *N-grams* generally lead to poor performance due to the overgeneration of candidates. In most cases, *Lemma (N,A)* candidates perform slightly better than *Tokens (N,A)* candidates, but the small difference does not justify the additional effort of lemmatization. The *TextRank* result on the SP dataset can almost be doubled (from .10 to .18 R-p_{ap}) by using noun phrases instead of Tokens (N,A) as candidates. This indicates that using higher quality candidates can have a positive impact on keyphrase extraction performance on longer documents.

6 Conclusions

We presented a new evaluation strategy for keyphrase extraction based on approximate keyphrase matching that accounts for the shortcomings of exact matching. In an annotation study, we showed that approximate matching (based on morphological variants and extracted keyphrases which include the gold standard keyphrases) corresponds well with human judgments. We showed that the approximate matching strategy is better suited to assess the performance of keyphrase extraction approaches.

We proposed a generalized framework for the comprehensive analysis and evaluation of keyphrase extraction systems, and compared the results of state-of-the-art unsupervised and supervised keyphrase extraction approaches on three evaluation datasets. We showed that the relative performance of the approaches heavily depends on the matching strategy as well as on the properties of the evaluation dataset especially the length of documents. We found that for small and medium sized documents (≈ 100–1000 tokens), the unsupervised approach using co-occurrence graph based ranking outperforms the supervised system by a wide margin. On larger documents, the supervised system outperforms the tf.idf and co-occurrence graph based approaches, but using approximate matching reveals that the improvement over the unsupervised tf.idf ranking based approaches is small. We also find that the performance of co-occurrence graph based methods on large documents can be increased by 80% when using higher quality noun phrase candidates instead of tokens restricted to nouns and adjectives.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the German Research Foundation under grant 798/1-3. We thank Anette Hulth and Min-Yen Kan for providing evaluation datasets. We thank Melanie Hartmann, Matthieu Hermet, and Johannes Hoffart for participating in the annotation study.

References

[1] K. Barker and N. Cornacchia. Using Noun Phrase Heads to Extract Document Keyphrases. In *Canadian Conference on AI*, pages 40–52. Springer, 2000.

[2] D. B. Bracewell, F. Ren, and S. Kuriowa. Multilingual Single Document Keyword Extraction for Information Retrieval. In *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, pages 517–522, 2005.

[3] A. Csomai and R. Mihalcea. Investigations in Unsupervised Back-of-the-Book Indexing. In *Proceedings of the Florida Artificial Intelligence Research Society*, pages 211–216, 2007.

[4] E. D'Avanzo and B. Magnini. A Keyphrase-Based Approach to Summarization: the LAKE System at DUC-2005. In *Proceedings of DUC Workshop at HLT/EMNLP'05*, Vancouver, B.C., Canada, October 6-8 2005.

[5] J. R. Finkel, T. Grenager, and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, 2005.

[6] E. Frank, G. W. Paynter, I. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-Specic Keyphrase Extraction. In *Proceedings of the 16th International Joint Conference on Artiticial Intelligence*, pages 668–673, 1999.

[7] I. Gurevych, M. Mühlhäuser, C. Müller, J. Steimle, M. Weimer, and T. Zesch. Darmstadt Knowledge Processing Repository Based on UIMA. In *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the GSCL*, 2007.

[8] C. Gutwin, G. Paynter, I. Witten, C. Nevill-Manning, and E. Frank. Improving Browsing in Digital Libraries with Keyphrase Indexes. *Decision Support Systems*, 27(1-2):81–104, 1999.

[9] K. M. Hammouda, D. N. Matute, and M. S. Kamel. CorePhrase: Keyphrase Extraction for Document Clustering. *Machine Learning and Data Mining in Pattern Recognition*, 2005:265–274, 2005.

[10] A. Hulth. Enhancing Linguistically Oriented Automatic Keyword Extraction. In *Proceedings of HLT/NAACL: Short Papers*, pages 17–20, 2004.

[11] M. Litvak and M. Last. Graph-Based Keyword Extraction for Single-Document Summarization. In *Proceedings of COLING*, pages 17–24, 2008.

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

[13] O. Medelyan and I. H. Witten. Thesaurus based automatic keyphrase indexing. In *In Proceedings of the Joint Conference on Digital Libraries (JCDL) 2006*, pages 296–297, 2006.

[14] O. Medelyan, I. H. Witten, and D. Milne. Topic Indexing with Wikipedia. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence*, pages 19–24, 2008.

[15] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 404–411, 2004.

[16] T. D. Nguyen and M.-Y. Kan. Keyphrase Extraction in Scientific Publications. In *Proceedings of International Conference on Asian Digital Libraries*, pages 317–326, 2007.

[17] Y. Park, R. J. Byrd, and B. K. Boguraev. Automatic Glossary Extraction: Beyond Terminology Identification. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, 2002.

[18] M.-S. Paukkeri, I. T. Nieminen, M. Pöllä, and T. Honkela. A Language-Independent Approach to Keyphrase Extraction and Evaluation. In *Coling 2008 Posters*, pages 83–86, 2008.

[19] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

[20] H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1995.

[21] T. Tomokiyo and M. Hurst. A Language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 33–40, 2003.

[22] P. D. Turney. Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2:303–336, 2000.

[23] P. D. Turney. Coherent Keyphrase Extraction via Web Mining. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 434–439, 2003.

[24] X. Wan and J. Xiao. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of AAAI*, pages 855–860, 2008.