TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction

Adrien Bougouin and **Florian Boudin** and **Béatrice Daille**
Université de Nantes, LINA, France
{adrien.bougouin,florian.boudin,beatrice.daille}@univ-nantes.fr

Abstract

Keyphrase extraction is the task of identifying single or multi-word expressions that represent the main topics of a document. In this paper we present TopicRank, a graph-based keyphrase extraction method that relies on a topical representation of the document. Candidate keyphrases are clustered into topics and used as vertices in a complete graph. A graph-based ranking model is applied to assign a significance score to each topic. Keyphrases are then generated by selecting a candidate from each of the top-ranked topics. We conducted experiments on four evaluation datasets of different languages and domains. Results show that TopicRank significantly outperforms state-of-the-art methods on three datasets.

1 Introduction

Keyphrases are single or multi-word expressions that represent the main topics of a document. Keyphrases are useful in many tasks such as information retrieval (Medelyan and Witten, 2008), document summarization (Litvak and Last, 2008) or document clustering (Han et al., 2007). Although scientific articles usually provide them, most of the documents have no associated keyphrases. Therefore, the problem of automatically assigning keyphrases to documents is an active field of research.

Automatic keyphrase extraction methods are divided into two categories: supervised and unsupervised methods. Supervised methods recast keyphrase extraction as a binary classification task (Witten et al., 1999), whereas unsupervised methods apply different kinds of techniques such as language modeling (Tomokiyo and Hurst, 2003), clustering (Liu et al., 2009) or graph-based ranking (Mihalcea and Tarau, 2004).

In this paper, we present a new unsupervised method called TopicRank. This new method is an improvement of the TextRank method applied to keyphrase extraction (Mihalcea and Tarau, 2004). In the TextRank method, a document is represented by a graph where words are vertices and edges represent co-occurrence relations. A graph-based ranking model derived from PageRank (Brin and Page, 1998) is then used to assign a significance score to each word. Here, we propose to represent a document as a complete graph where vertices are not words but topics. We define a topic as a cluster of similar single and multiword expressions.

Our approach has several advantages over TextRank. Intuitively, ranking topics instead of words is a more straightforward way to identify the set of keyphrases that covers the main topics of a document. To do so, we simply select a keyphrase candidate from each of the top-ranked clusters. Clustering keyphrase candidates into topics also eliminates redundancy while reinforcing edges. This is very important because the ranking performance strongly depends on the conciseness of the graph, as well as its ability to precisely represent semantic relations within a document. Hence, another advantage of our approach is the use of a complete graph that better captures the semantic relations between topics.

To evaluate TopicRank, we follow Hasan and Ng (2010) who stated that multiple datasets must be used to evaluate and fully understand the strengths and weaknesses of a method. We use four evaluation datasets of different languages, document sizes and domains, and compare the keyphrases extracted by TopicRank against three baselines (TF-IDF and two graph-based methods). TopicRank outperforms the baselines on three of the datasets. As for the fourth one, an additional experiment shows that an improvement could be achieved with a more effective selection strategy.

International Joint Conference on Natural Language Processing, pages 543–551,
Nagoya, Japan, 14-18 October 2013.

The rest of this paper is organized as follows. Section 2 presents the existing methods for the keyphrase extraction task, Section 3 details our proposed approach, Section 4 describes the evaluation process and Section 5 shows the analyzed results. Finally, Section 6 concludes this work and suggests directions for future work.

2 Related Work

The task of automatic keyphrase extraction has been well studied and many supervised and unsupervised approaches have been proposed. For supervised methods, keyphrase extraction is often treated as a binary classification task (Witten et al., 1999). Unsupervised approaches proposed so far have involved a number of techniques, including language modeling (Tomokiyo and Hurst, 2003), clustering (Liu et al., 2009) and graph-based ranking (Mihalcea and Tarau, 2004). While supervised approaches have generally proven to be more successful, the need for training data and the bias towards the domain on which they are trained remain two critical issues.

In this paper, we concentrate on graph-based ranking methods for keyphrase extraction. Starting with TextRank (Mihalcea and Tarau, 2004), these methods are becoming the most widely used unsupervised approaches for keyphrase extraction. In TextRank, a document is represented as a graph in which vertices are words connected if they co-occur in a given window of words. The significance of each vertex is computed using a random walk algorithm derived from PageRank (Brin and Page, 1998). Words corresponding to the top ranked vertices are then selected and assembled to generate keyphrases.

Wan and Xiao (2008) propose SingleRank, a simple modification of TextRank that weights the edges with the number of co-occurrences and no longer extracts keyphrases by assembling ranked words. Keyphrases are noun phrases extracted from the document and ranked according to the sum of the significance of the words they contain. Although it improves the results, this scoring method has no proper justification and tends to assign high scores to long but non important phrases. For example, “nash equilibrium”, from the file *J-14.txt* of our evaluation dataset named SemEval, is a keyphrase composed of the two most significant words in the document, according to SingleRank. Therefore, SingleRank succeeds to extract it, but candidates such as “unique nash equilibrium” or “exact nash equilibrium” which are longer, then have a better score, are extracted too. With TopicRank, we aim to circumvent this by ranking clusters of single and multi-word expressions instead of words.

Wan and Xiao (2008) use a small number of nearest neighbor documents to compute more accurate word co-occurrences and reinforce edge weights in the word graph. Borrowing co-occurrence information from multiple documents, their approach improves the word ranking performance. Instead of using words, Liang et al. (2009) use keyphrase candidates as vertices. Applied to Chinese, their method uses query log knowledge to determine phrase boundaries. Tsatsaronis et al. (2010) propose to connect vertices employing semantic relations computed using WordNet (Miller, 1995) or Wikipedia. They also experiment with different random walk algorithms, such as HITS (Kleinberg, 1999) or modified PageRank.

Liu et al. (2010) consider the topics of words using a Latent Dirichlet Allocation model (Blei et al., 2003, LDA). As done by Haveliwala (2003) for Information Retrieval, they propose to decompose PageRank into multiple PageRanks specific to various topics. A topic-biased PageRank is computed for each topic and corresponding word scores are combined. As this method uses a LDA model, it requires training data. With TopicRank, we also consider topics, but our aim is to use a single document, the document to be analyzed.

3 TopicRank

TopicRank is an unsupervised method that aims to extract keyphrases from the most important topics of a document. Topics are defined as clusters of similar keyphrase candidates. Extracting keyphrases from a document consists in the following steps, illustrated in Figure 1. First, the document is preprocessed (sentence segmentation, word tokenization and Part-of-Speech tagging) and keyphrase candidates are clustered into topics. Then, topics are ranked according to their importance in the document and keyphrases are extracted by selecting one keyphrase candidate for each of the most important topics.

Section 3.1 first explains how the topics are identified within a document, section 3.2 presents the approach we use to rank them and section 3.3 describes the keyphrase selection.

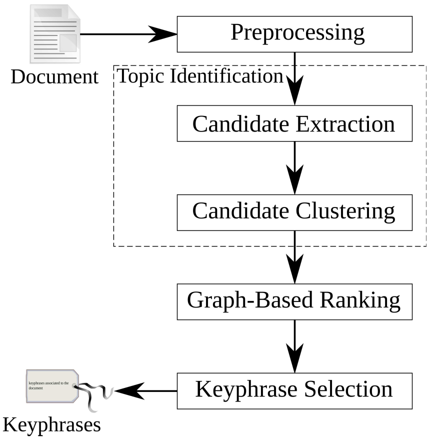


Figure 1: Processing steps of TopicRank.

3.1 Topic Identification

Keyphrases describe the most important topics of a document, thus the first step is to identify the keyphrase candidates that represent them. Hulth (2003) stated that most keyphrases assigned by human readers are noun phrases. Hence, the most important topics of a document can be found by extracting their most significant noun phrases. We follow Wan and Xiao (2008) and extract the longest sequences of nouns and adjectives from the document as keyphrase candidates. Other methods use syntactically filtered n-grams that are most likely to contain a larger number of candidates matching with reference keyphrases, but the n-gram restricted length is a problem. Indeed, n-grams do not always capture as much information as the longest noun phrases. Also, they are less likely to be grammatically correct.

In a document, a topic is usually conveyed by more than one noun phrase. Consequently, some keyphrase candidates are redundant in regard to the topic they represent. Existing graph-based methods (TextRank, SingleRank, etc.) do not take that fact into account. Keyphrase candidates are usually treated independently and the information about the topic they represent is scattered throughout the graph. Thus, we propose to group similar noun phrases as a single entity, a topic.

We consider that two keyphrase candidates are similar if they have at least 25% of overlapping words[1]. Keyphrase candidates are stemmed to reduce their inflected word forms into root forms[2]. To automatically group similar candidates into topics, we use a Hierarchical Agglomerative Clustering (HAC) algorithm. Among the commonly used linkage strategies, which are complete, average and single linkage, we use the average linkage, because it stands as a compromise between complete and single linkage. In fact, using a highly agglomerative strategy such as complete linkage is more likely to group topically unrelated keyphrase candidates, whereas a strategy such as single linkage is less likely to group topically related keyphrase candidates.

3.2 Graph-Based Ranking

TopicRank represents a document by a complete graph in which topics are vertices and edges are weighted according to the strength of the semantic relations between vertices. Then, TextRank's graph-based ranking model is used to assign a significance score to each topic.

3.2.1 Graph Construction

Formally, let $G = (V, E)$ be a complete and undirected graph where V is a set of vertices and the edges E a subset[3] of $V \times V$. Vertices are topics and the edge between two topics t_i and t_j is weighted according to the strength of their semantic relation. t_i and t_j have a strong semantic relation if their keyphrase candidates often appear close to each other in the document. Therefore, the weight $w_{i,j}$ of their edge is defined as follows:

$$w_{i,j} = \sum_{c_i \in t_i} \sum_{c_j \in t_j} \text{dist}(c_i, c_j) \tag{1}$$

$$\text{dist}(c_i, c_j) = \sum_{p_i \in \text{pos}(c_i)} \sum_{p_j \in \text{pos}(c_j)} \frac{1}{|p_i - p_j|} \tag{2}$$

where $\text{dist}(c_i, c_j)$ refers to the reciprocal distances between the offset positions of the candidate keyphrases c_i and c_j in the document and where $\text{pos}(c_i)$ represents all the offset positions of the candidate keyphrase c_i.

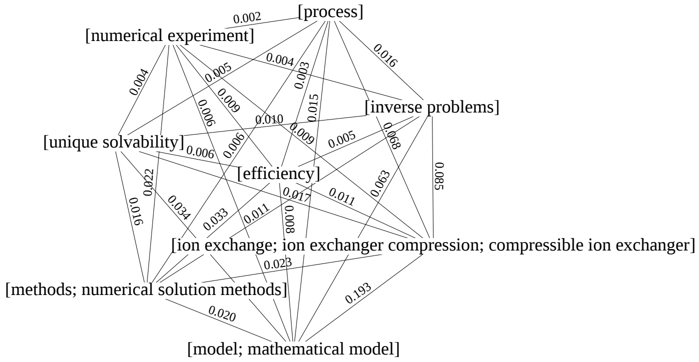
Our approach to construct the graph differs from TextRank. G is a complete graph and topics are therefore interconnected. The completeness of the graph has the benefit of providing a more exhaustive view of the relations between topics. Also, computing weights based on the distances between offset positions bypasses the need for a manually defined parameter, such as the window of words used by state-of-the-art methods (TextRank, SingleRank, etc).

[1] The value of 25% has been defined empirically.

[2] We chose to use stems because of the availability of stemmers for various languages, but using lemmas is another possibility that could probably work better.

[3] $E = \{(v_1, v_2) \mid \forall v_1, v_2 \in V,\ v_1 \neq v_2\}$

Inverse problems for a mathematical model of ion exchange in a compressible ion exchanger
A mathematical model of ion exchange is considered, allowing for ion exchanger compression in the process of ion exchange. Two inverse problems are investigated for this model, unique solvability is proved, and numerical solution methods are proposed. The efficiency of the proposed methods is demonstrated by a numerical experiment.



Keyphrases assigned by human annotators:
ion exchange; mathematical model; inverse problems; numerical solution methods; unique solvability; compressible ion exchanger; ion exchanger compression
Keyphrases assigned by TopicRank:
ion exchange; ***mathematical model***; ***inverse problems***; ***numerical solution methods***; process; ***unique solvability***; efficiency; numerical experiment

Figure 2: Sample graph build by TopicRank from Inspec, file *2040.abstr*.

Figure 2 shows a sample graph built for an abstract from one of our evaluation datasets (Inspec). Vertices are topics, represented as clusters of lexically similar keyphrase candidates, and connected with all the others. In the example, we see the naivety of our clustering approach. Indeed, the clustering succeeds to group “ion exchanger”, “ion exchanger compression” and “compressible ion exchanger”, but the clustering of “methods” with “numerical solution methods” and “model” with “mathematical model” may be ambiguous as “methods” and “model” can be used to refer to other methods or models.

3.2.2 Subject Ranking

Once the graph is created, the graph-based ranking model TextRank, proposed by Mihalcea and Tarau (2004), is used to rank the topics. This model assigns a significance score to topics based on the concept of “voting”: high-scoring topics contribute more to the score of their connected topic t_i:

$$S(t_i) = (1 - \lambda) + \lambda \times \sum_{t_j \in V_i} \frac{w_{j,i} \times S(t_j)}{\displaystyle\sum_{t_k \in V_j} w_{j,k}} \quad (3)$$

where V_i are the topics voting for t_i and λ is a damping factor generally defined to 0.85 (Brin and Page, 1998).

3.3 Keyphrase Selection

Keyphrase selection is the last step of TopicRank. For each topic, only the most representative keyphrase candidate is selected. This selection avoids redundancy and leads to a good coverage of the document topics, because extracting k keyphrases precisely covers k topics.

To find the candidate that best represents a topic, we propose three strategies. Assuming that a topic is first introduced by its generic form, the first strategy is to select the keyphrase candidate that appears first in the document. The second strategy assumes that the generic form of a topic is the one that is most frequently used and the third strategy selects the centroid of the cluster. The centroid is the candidate that is the most similar to the other candidates of the cluster[4].

4 Experimental Settings

4.1 Datasets

To compare the keyphrases extracted by TopicRank against existing methods, we employ four standard evaluation dataset of different languages, document sizes and domains.

The first dataset, formerly used by Hulth (2003), contains 2000 English abstracts of journal papers from the Inspec database. The 2000 abstracts are divided into three sets: a training set, which contains 1000 abstracts, a validation set containing 500 abstracts and a test set containing the 500 remaining abstracts. In our experiments we use the 500 abstracts from the test set. Several reference keyphrase sets are available with this dataset. Just as Hulth (2003), we use the uncontrolled reference, created by professional indexers.

The second dataset was built by Kim et al. (2010) for the keyphrase extraction task of the SemEval 2010 evaluation campaign. This dataset is

[4] The similarity between two candidates is computed with the stem overlap measure used by the clustering algorithm.

Corpus	Documents				Keyphrases		
	Type	Language	Number	Tokens average	Total	Average	Missing
Inspec	Abstracts	English	500	136.3	4913	9.8	21.8%
SemEval	Papers	English	100	5179.6	1466	14.7	19.3%
WikiNews	News	French	100	309.6	964	9.6	4.4%
DEFT	Papers	French	93	6844.0	485	5.2	18.2%

Table 1: Dataset statistics (missing keyphrases are counted based on their stemmed form).

composed of 284 scientific articles (in English) from the ACM Digital Libraries (conference and workshop papers). The 284 documents are divided into three sets: a trial set containing 40 documents, a training set, which contains 144 documents and a test set containing 100 documents. In our experiments we use the 100 documents of the test set. As for the reference keyphrases, we use the combination of author and reader assigned keyphrases provided by Kim et al. (2010).

The third dataset is a French corpus that we created from the French version of WikiNews[5]. It contains 100 news articles published between May 2012 and December 2012. Each document has been annotated by at least three students. We combined the annotations of each document and removed the lexical redundancies. All of the 100 documents are used in our experiments.

The fourth dataset is a French corpus made for the keyphrase extraction task of the DEFT 2012 evaluation campaign (Paroubek et al., 2012). It contains 468 scientific articles extracted from Érudit. These documents are used for two tasks of DEFT and are, therefore, divided in two datasets of 244 documents each. In our experiments we use the test set of the second task dataset. It contains 93 documents provided with author keyphrases.

Table 1 gives statistics about the datasets. They are different in terms of document sizes and number of assigned keyphrases. The Inspec and WikiNews datasets have shorter documents (abstract and news articles) compared to SemEval and DEFT that both contain full-text scientific articles. Also, the keyphrases provided with the datasets are not always present in the documents (less than 5% of missing keyphrases for Wikinews and about 20% of missing keyphrases for the other datasets). This induces a bias in the results. As explained by Hasan and Ng (2010), some researchers avoid this problem by removing missing keyphrases from the references. In our experiments, missing keyphrases have not been removed. However, we evaluate with stemmed forms of candidates and reference keyphrases to reduce mismatches.

4.2 Preprocessing

For each dataset, we apply the following preprocessing steps: sentence segmentation, word tokenization and Part-of-Speech tagging. For word tokenization, we use the TreebankWordTokenizer provided by the python Natural Language ToolKit (Bird et al., 2009) for English and the Bonsai word tokenizer[6] for French. For Part-of-Speech tagging, we use the Stanford POS-tagger (Toutanova et al., 2003) for English and MElt (Denis and Sagot, 2009) for French.

4.3 Baselines

For comparison purpose, we use three baselines. The first baseline is TF-IDF (Spärck Jones, 1972), commonly used because of the difficulty to achieve competitive results against it (Hasan and Ng, 2010). This method relies on a collection of documents and assumes that the k keyphrase candidates containing words with the highest TF-IDF weights are the keyphrases of the document. As TopicRank aims to be an improvement of the state-of-the-art graph-based methods for keyphrase extraction, the last two baselines are TextRank (Mihalcea and Tarau, 2004) and SingleRank (Wan and Xiao, 2008). In these methods, the graph is undirected, vertices are syntactically filtered words (only nouns and adjectives) and the edges are created based on the co-occurrences of words within a window of 2 for

[5] The WikiNews dataset is available for free at the given url: `https://github.com/adrien-bougouin/WikinewsKeyphraseCorpus`.

[6] The Bonsai word tokenizer is a tool provided with the Bonsai PCFG-LA parser: `http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html`.

Methods	Inspec			SemEval			WikiNews			DEFT		
	P	R	F	P	R	F	P	R	F	P	R	F
TF-IDF	32.7	38.6	33.4	13.2	8.9	10.5	33.9	35.9	34.3	10.3	19.1	13.2
TextRank	14.2	12.5	12.7	7.9	4.5	5.6	9.3	8.3	8.6	4.9	7.1	5.7
SingleRank	34.8	40.4	**35.2**	4.6	3.2	3.7	19.4	20.7	19.7	4.5	9.0	5.9
TopicRank	27.6	31.5	27.9	14.9	10.3	**12.1**†	35.0	37.5	**35.6**†	11.7	21.7	**15.1**†

Table 2: Comparison of TF-IDF, TextRank, SingleRank and TopicRank methods, when extracting a maximum of 10 keyphrases. Results are expressed as a percentage of precision (P), recall (R) and f-score (F). † indicates TopicRank's significant improvement over TextRank and SingleRank at 0.001 level using Student's t-test.

TextRank and 10 for SingleRank. As well as their window size, they differ in the weighting of the graph: TextRank has an unweighted graph and SingleRank has a graph weighted with the number of co-occurrences between the words. A graph-based ranking model derived from PageRank (Brin and Page, 1998) ranks each vertex and extracts multi-word keyphrases according to the ranked words. In TextRank, the k-best words are used as keyphrases and the adjacent sequences in the document are collapsed into multi-word keyphrases. Although k is normally proportional to the number of vertices in the graph, we set it to a constant number, because experiments conducted by Hasan and Ng (2010) show that the optimal value of the ratio depends on the size of the document. In SingleRank, noun phrases extracted with the same method as TopicRank are ranked by a score equal to the sum of their words scores. Then, the k-best noun phrases are selected as keyphrases.

For all the baselines, we consider keyphrase candidates which have the same stemmed form as redundant. Once they are ranked we keep the best candidate and remove the others. This can only affect the results in a positive way, because the evaluation is performed with stemmed forms, which means that removed candidates are considered equal to the retained candidate.

4.4 Evaluation Measures

The performances of TopicRank and the baselines are evaluated in terms of precision, recall and f-score (f1-measure) when a maximum of 10 keyphrases are extracted ($k = 10$). As said before, the candidate and reference keyphrases are stemmed to reduce the number of mismatches.

5 Results

To validate our approach, we designed three experiments. The first experiment compares TopicRank[7] to the baselines[8], the second experiment individually evaluates the modifications of TopicRank compared to SingleRank[9] and the last experiment compares the keyphrase selection strategies. To show that the clusters are well ranked, we also present the results that could be achieved with a "perfect" keyphrase selection strategy.

Table 2 shows the results of TopicRank and the three baselines. Overall, our method outperforms TextRank, SingleRank and TF-IDF. The results of TopicRank and the baselines are lower on SemEval and DEFT (less than 16% of f-score), so we deduce that it is more difficult to treat long documents than short ones. On Inspec, TopicRank fails to do better than all the baselines, but on SemEval, WikiNews and DEFT, it performs better than TF-IDF and significantly outperforms TextRank and SingleRank. Also, we observe a gap between TF-IDF's and the two graph-based baselines results. Although TopicRank is a graph-based method, it overcomes this gap by almost tripling the f-score of both TextRank and SingleRank.

Table 3 shows the individual modifications of TopicRank compared to SingleRank. We evaluate SingleRank when vertices are keyphrase candidates (+phrases), vertices are topics (+topics) and when TopicRank's graph construction is used

[7] Results reported for TopicRank are obtained with the first position selection strategy.

[8] TopicRank and the baselines implementations can be found at the given url: `https://github.com/adrien-bougouin/KeyBench/tree/ijcnlp_2013`.

[9] The second experiment is performed with SingleRank instead of TextRank, because SingleRank also uses a graph with weighted edges and is, therefore, closer to TopicRank.

Methods	Inspec			SemEval			WikiNews			DEFT		
	P	R	F	P	R	F	P	R	F	P	R	F
SingleRank	34.8	40.4	35.2	4.6	3.2	3.7	19.4	20.7	19.7	4.5	9.0	5.9
+phrases	21.5	25.9	22.1	9.6	7.0	8.0†	28.6	30.1	28.9†	10.5	19.7	13.5†
+topics	26.6	30.2	26.8	14.7	10.2	11.9†	31.0	32.8	31.4†	11.5	21.4	14.8†
+complete	34.9	41.0	**35.5**	5.5	3.8	4.4	20.0	21.4	20.3	4.4	9.0	5.8
TopicRank	27.6	31.5	27.9	14.9	10.3	**12.1**†	35.0	37.5	**35.6**†	11.7	21.7	**15.1**†

Table 3: Comparison of the individual modifications from SingleRank to TopicRank, when extracting a maximum of 10 keyphrases. Results are expressed as a percentage of precision (P), recall (R) and f-score (F). † indicates a significant improvement over SingleRank at 0.001 level using Student's t-test.

with word vertices (+complete). Using keyphrase candidates as vertices significantly improves SingleRank on SemEval, WikiNews and DEFT. On Inspec, it induces a considerable loss of performance caused by an important deficit of connections that leads to connected components, as shown in Figure 3. When we look at the distribution of "fuzzy" into the graph, we can see that it is scattered among the connected components and, therefore, increases the difficulty to select "fuzzy Bayesian inference techniques" as a keyphrase (according to the reference). The other datasets contain longer documents, which may dampen this problem. Overall, using topics as vertices performs better than using keyphrase candidates. Using topics significantly outperforms SingleRank on SemEval, WikiNews and DEFT. As for the new graph construction, SingleRank is improved on Inspec, SemEval and WikiNews. Results on DEFT are lower than SingleRank, but still competitive. Although the improvements are not significant, the competitive results point out that the new graph construction can be used instead of the former method, which requires to manually define a window of words. Experiments show that the three contributions are improvements and TopicRank benefits from each of them.

Table 4 shows the results of TopicRank when selecting either the first appearing candidate, the most frequent one or the centroid of each cluster. Selecting the first appearing keyphrase candidate is the best strategy of the three. It significantly outperforms the frequency and the centroid strategies on SemEval, WikiNews and DEFT. On SemEval and DEFT, we observe a huge gap between the results of the first position strategy and the others. The two datasets are composed of scientific articles where the full form of the main topics are often introduced at the beginning and then, conveyed by abbreviations or inherent concepts (e.g. the file *C-17.txt* from SemEval contains *packet-switched network* as a keyphrase where *packet* is more utilized in the content). These are usually more similar to the generic form and/or more frequent, which explains the observed gap.

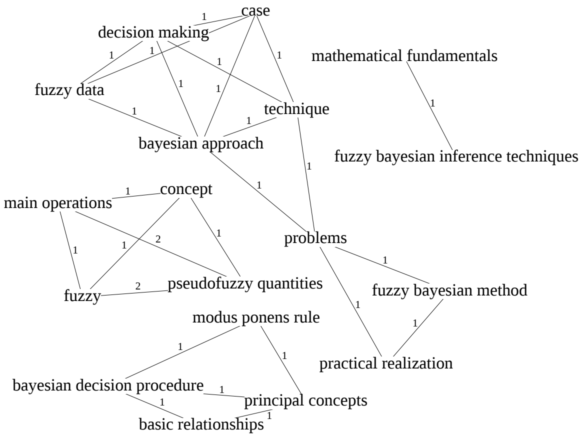


Figure 3: Connected component problem with the method SingleRank+phrases. Example taken from Inspec, file *1931.abstr*.

To observe the ranking efficiency of TopicRank, we also evaluate it without taking the keyphrase selection strategy into account. To do so, we extract the top-ranked clusters and mark the reference keyphrases into them. We deduce the upper bound results of our method by computing the precision, recall and f-score where the number of correct matches is equal to the number of clusters containing at least one reference keyphrase. The upper bound results show that our method could possibly perform better than all the baselines for the four datasets. Even on Inspec, the loss of performance can be bypassed by a more efficient keyphrase selection strategy.

Methods	Inspec			SemEval			WikiNews			DEFT		
	P	R	F	P	R	F	P	R	F	P	R	F
First position	27.6	31.5	27.9	14.9	10.3	12.1†	35.0	37.5	35.6†	11.7	21.7	15.1†
Frequency	26.7	30.2	26.8	1.7	1.2	1.4	25.7	27.6	26.2	1.9	3.8	2.5
Centroid	24.5	28.0	24.7	1.9	1.2	1.5	28.1	29.9	28.5	2.6	5.0	3.4
Upper bound	36.4	39.0	**35.6**	37.6	25.8	**30.3**	42.5	44.8	**42.9**	14.9	28.0	**19.3**

Table 4: Comparison of the keyphrase candidate selection strategies against the best possible strategy (upper bound), when extracting a maximum of 10 keyphrases. Results are expressed as a percentage of precision (P), recall (R) and f-score (F). † indicates the first position strategy's significant improvement over the frequency and the centroid strategies at 0.001 level using Student's t-test.

6 Conclusion and Future Work

In this paper we presented TopicRank, an unsupervised method for keyphrase extraction. TopicRank extracts the noun phrases that represent the main topics of a document. The noun phrases are clustered into topics and used as vertices in a complete graph. The resulting graph stands as a topical representation of the document. Topics are scored using the TextRank ranking model and keyphrases are then extracted by selecting the most representative candidate from each of the top-ranked topics. Our approach offers several advantages over existing graph-based keyphrase extraction methods. First, as redundant keyphrase candidates are clustered, extracted keyphrases cover the main topics of the document better. The use of a complete graph also captures the relations between topics without any manually defined parameters and induces better or similar performances than the state-of-the-art connection method that uses a co-occurrence window. We conducted experiments on four standard evaluation datasets of different languages, document sizes and domains. Results show that TopicRank outperforms TF-IDF and significantly improves the state-of-the-art graph-based methods on three of them.

In future work, we will further improve the topic identification and the keyphrase selection. More precisely, we will develop an evaluation process to determine cluster quality and then focus on experimenting with other clustering algorithms and investigate the use of linguistic knowledge for similarity measures. As for the keyphrase selection, our experiments show that the current method does not provide the best solution that could be achieved with the ranked clusters. We plan to improve it using machine learning methods.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful advice and comments. This work was supported by the French National Research Agency (TermITH project – ANR-12-CORD-0029).

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1):107–117.

Pascal Denis and Benoît Sagot. 2009. Coupling an Annotated Corpus and a Morphosyntactic Lexicon for State-of-the-Art POS Tagging with Less Human Effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pages 110–119, Hong Kong, December. City University of Hong Kong.

Juhyun Han, Taehwan Kim, and Joongmin Choi. 2007. Web Document Clustering by Using Automatic Keyphrase Extraction. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 56–59, Washington, DC, USA. IEEE Computer Society.

Kazi Saidul Hasan and Vincent Ng. 2010. Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 365–373, Stroudsburg, PA, USA. Association for Computational Linguistics.

Taher H. Haveliwala. 2003. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796.

Anette Hulth. 2003. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Stroudsburg, PA, USA. Association for Computational Linguistics.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM)*, 46(5):604–632, sep.

Weiming Liang, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2009. Extracting Keyphrases from Chinese News Articles Using TextRank and Query Log Knowledge. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pages 733–740, Hong Kong, December. City University of Hong Kong.

Marina Litvak and Mark Last. 2008. Graph-Based Keyword Extraction for Single-Document Summarization. In *Proceedings of the Workshop on Multi-Source Multilingual Information Extraction and Summarization*, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to Find Exemplar Terms for Keyphrase Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, pages 257–266, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic Keyphrase Extraction Via Topic Decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.

Olena Medelyan and Ian H. Witten. 2008. Domain-Independent Automatic Keyphrase Indexing with Small Training Sets. *Journal of the American Society for Information Science and Technology*, 59(7):1026–1040, may.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order Into Texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: a Lexical Database for English. *Communications of the Association for Computational Linguistics*, 38(11):39–41.

Patrick Paroubek, Pierre Zweigenbaum, Dominic Forest, and Cyril Grouin. 2012. Indexation libre et contrôlée d'articles scientifiques. Présentation et résultats du défi fouille de textes DEFT2012 (Controlled and Free Indexing of Scientific Papers. Presentation and Results of the DEFT2012 Text-Mining Challenge) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, Workshop DEFT 2012: DÉfi Fouille de Textes (DEFT 2012 Workshop: Text Mining Challenge)*, pages 1–13, Grenoble, France, June. ATALA/AFCP.

Karen Spärck Jones. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21.

Takashi Tomokiyo and Matthew Hurst. 2003. A Language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, pages 33–40, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

George Tsatsaronis, Iraklis Varlamis, and Kjetil Nørvåg. 2010. SemanticRank: Ranking Keywords and Sentences Using Semantic Graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1074–1082, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, pages 855–860. AAAI Press.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill Manning. 1999. KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries*, pages 254–255, New York, NY, USA. ACM.