Keyphrase Annotation with Graph Co-Ranking

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

Abstract

Keyphrase annotation is the task of identifying textual units that represent the main content of a document. Keyphrase annotation is either carried out by extracting the most important phrases from a document, keyphrase extraction, or by assigning entries from a controlled domain-specific vocabulary, keyphrase assignment. Assignment methods are generally more reliable. They provide better-formed keyphrases, as well as keyphrases that do not occur in the document. But they are often silent on the contrary of extraction methods that do not depend on manually built resources. This paper proposes a new method to perform both keyphrase extraction and keyphrase assignment in an integrated and mutual reinforcing manner. Experiments have been carried out on datasets covering different domains of humanities and social sciences. They show statistically significant improvements compared to both keyphrase extraction and keyphrase assignment state-of-the art methods.

1 Introduction

Keyphrases are words and phrases that give a synoptic picture of what is important within a document. They are useful in many tasks such as document indexing (Gutwin et al., 1999), text categorization (Hulth and Megyesi, 2006) or summarization (Litvak and Last, 2008). However, most documents do not provide keyphrases, and the daily flow of new documents makes the manual keyphrase annotation impractical. As a consequence, automatic keyphrase annotation has received special attention in the NLP community and many methods have been proposed (Hasan and Ng, 2014).

The task of automatic keyphrase annotation consists in identifying the main concepts, or topics, addressed in a document. Such task is crucial to access relevant scientific documents that could be useful for researchers. Keyphrase annotation methods fall into two broad categories: keyphrase extraction and keyphrase assignment methods. Keyphrase extraction methods extract the most important words or phrases occurring in a document, while assignment methods provide controlled keyphrases from a domain-specific terminology (controlled vocabulary).

The automatic keyphrase annotation task is often reduced to the sole keyphrase extraction task. Unlike assignment methods, extraction methods do not require domain specific controlled vocabularies that are costly to create and to maintain. Furthermore, they are able to identify new concepts that have not been yet recorded in the thesaurus or ontologies. However, extraction methods often output ill-formed or inappropriate keyphrases (Medelyan and Witten, 2008), and they produce only keyphrases that actually occur in the document.

Observations made on manually assigned keyphrases from scientific papers of specialized domains show that professional human indexers both extract keyphrases from the content of the document and assign keyphrases based on their knowledge of the domain (Liu et al., 2011). Here, we propose an approach that mimics this behaviour and jointly extracts and assigns keyphrases. We use two graph representations, one for the document and one for the specialized domain. Then, we apply a co-ranking algorithm to perform both keyphrase extraction and assignment in a mutually reinforcing manner. We perform

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2945–2955, Osaka, Japan, December 11-17 2016.

experiments on bibliographic records in three domains belonging to humanities and social sciences: linguistics, information science and archaeology. Along with this approach come two contributions. First, we present a simple yet efficient assignment extension of a state-of-the-art graph-based keyphrase extraction method, TopicRank (Bougouin et al., 2013). Second, we circumvent the need for a controlled vocabulary by leveraging reference keyphrases from training data and further take advantage of their relationship within the training data.

2 Related Work

2.1 Keyphrase extraction

Keyphrase extraction is the most common approach to tackle the automatic keyphrase annotation task. Previous work includes many approaches (Hasan and Ng, 2014), from statistical ranking (Salton et al., 1975) to binary classification (Witten et al., 1999), through graph-based ranking (Mihalcea and Tarau, 2004) of keyphrase candidates. As our approach uses graph-based ranking, we focus on the latter. For a detailed overview of keyphrase extraction methods, refer to (Hasan and Ng, 2010; Hasan and Ng, 2014).

Since the seminal work of Mihalcea and Tarau (2004), graph-based ranking approaches to keyphrase extraction are becoming increasingly popular. The original idea behind these approaches is to build a graph from the document and rank its nodes according to their importance using centrality measures.

In TextRank (Mihalcea and Tarau, 2004), the input document is represented as a co-occurrence graph in which nodes are words. Two words are connected by an edge if they co-occur in a fixed-sized window of words. A random walk algorithm is used to iteratively rank the words, then extract the keyphrases by concatenating the most important words.

The random walk algorithm simulates the "voting concept", or recommendation: a node is important if it is connected to many other nodes, and if many of those are important. Thus, let $G = (V, E)$ be an undirected graph with a set of vertices V and a set of edges E, and let $E(v_i)$ be the set of nodes connected to the node v_i. The score $S(v_i)$ of a vertex v_i is initialized to 1 and computed iteratively until convergence using the following equation:

$$S(v_i) = (1 - \lambda) + \lambda \sum_{v_j \in E(v_i)} \frac{S(v_j)}{|E(v_j)|} \tag{1}$$

where λ is a damping factor that has been set to 0.85 by Brin and Page (1998) for a trade-off between ranking accuracy and fast convergence.

Following up the work of Mihalcea and Tarau (2004), Wan and Xiao (2008) added edge weights (co-occurrence numbers) to the random walk and further improved the graph with co-occurrence information borrowed from similar documents. To extract keyphrases from a document, they first look for five similar documents, then use them to add new edges between words within the graph and reinforce the weight of existing edges. Liu et al. (2010) biased multiple graphs with topic probabilities drawn from LDA (Latent Dirichlet Allocation) (Blei et al., 2003), to rank the words regarding each graph and to merge the rankings together. This method performs as many rankings as the number of topics and gives higher importance scores to high-ranking words for as many topics as possible. By doing so, Liu et al. (2010) increase the topic coverage provided by the extracted keyphrases.

Most recently, Zhang et al. (2013) and Bougouin et al. (2013) explored further the value of topics for keyphrase extraction. Zhang et al. (2013) used graph co-ranking to improve the method of Liu et al. (2010) by introducing LDA topics right inside the graph. Bougouin et al. (2013) proposed to represent topics as clusters of similar keyphrase candidates within the document (i.e. words and phrases from the document), to rank these topics instead of the words and to extract the most representative candidate as keyphrase for each important topic. As our work extends that of Bougouin et al. (2013), we present a detailed description of their method in Section 3.1.

2.2 Keyphrase assignment

Keyphrase assignment provides keyphrases for every document of a specific domain using a controlled vocabulary. Dissimilar to keyphrase extraction, keyphrase assignment also aims to provide keyphrases that do not occur within the document. This task is more difficult than keyphrase extraction and has, therefore, seldom been employed for automatic keyphrase annotation. The state-of-the art method for keyphrase assignment is KEA++ (Medelyan and Witten, 2006).

KEA++ uses a domain-specific thesaurus to assign keyphrases to a document. First, keyphrase candidates are selected among the n-grams of the document. N-grams that do not match a thesaurus entry are either removed or substituted by a synonym that matches a thesaurus entry. This candidate selection approach induces a limitation of keyphrase assignment, refered to as keyphrase indexing by Medelyan and Witten (2006), because it only assigns keyphrases if they occur within the document. Second, KEA++ exploits the semantic relationships between keyphrase candidates within the thesaurus as the main feature of a Naive Bayes classifier. Compared to similar methods without domain specific resources, KEA++ achieves better performance. However, such resources are not readily available for most domains, and if so, they could be quickly out of date. The application scenario of KEA++ are thus restricted.

Our proposition is to model with graphs both keyphrase extraction and assignment and to take benefit of this unified modelling to perform accurate keyphrase annotation.

3 Co-ranking for Keyphrase Annotation

This section presents TopicCoRank[1], our keyphrase annotation method built on the existing method TopicRank (Bougouin et al., 2013) to which we add keyphrase assignment. We first detail TopicRank, then present our contributions.

3.1 TopicRank

TopicRank is a graph-based keyphrase extraction method that relies on the following five steps:

1. **Keyphrase candidate selection.** Following previous work (Hasan and Ng, 2010; Wan and Xiao, 2008), keyphrase candidates are selected from the sequences of adjacent nouns and adjectives that occur within the document (`/(N|A)+/`).

2. **Topical clustering.** Similar keyphrase candidates c are clustered into topics based on the words they share. Bougouin et al. (2013) use a Hierarchical Agglomerative Clustering (HAC) with a stem overlap similarity (see equation 2) and an average linkage. At the beginning, each keyphrase candidate is a single cluster, then candidates sharing an average of $1/4$ stemmed words with the candidates of another cluster are iteratively added to the latter.

$$\text{sim}(c_i, c_j) = \frac{|\text{stems}(c_i) \cap \text{stems}(c_j)|}{|\text{stems}(c_i) \cup \text{stems}(c_j)|} \tag{2}$$

   where $\text{stems}(c_i)$ is the set of stemmed words of the keyphrase candidate c_i.

3. **Graph construction.** A complete graph is built, in which nodes are topics and edges are weighted according to the strength of the semantic relation between the connected topics. The closer are the pairs of candidates $\langle c_i, c_j \rangle$ of two topics t_i and t_j within the document, the stronger is their semantic relation $w_{i,j}$:

$$w_{i,j} = \sum_{c_i \in t_i} \sum_{c_j \in t_j} \text{dist}(c_i, c_j) \tag{3}$$

$$\text{dist}(c_i, c_j) = \sum_{p_i \in \text{pos}(c_i)} \sum_{p_j \in \text{pos}(c_j)} \frac{1}{|p_i - p_j|} \tag{4}$$

   where $\text{pos}(c_i)$ represents all of the offset positions of the first word of the keyphrase candidate c_i.

[1] TopicCoRank is open source and publicly available at `https://github.com/adrien-bougouin/KeyBench/tree/coling_2016/`

4. **Topic ranking.** Topics t are ranked using the importance score $S(t_i)$ of the TextRank formula, as modified by Wan and Xiao (2008) to leverage edge weights:

$$S(t_i) = (1 - \lambda) + \lambda \sum_{t_j \in E(t_i)} \frac{w_{ij} S(t_j)}{\sum\limits_{t_k \in E(t_j)} w_{jk}} \tag{5}$$

5. **Keyphrase selection.** One keyphrase candidate is selected from each of the N most important topics: the first occurring keyphrase candidate.

Our work extends TopicRank to assign domain-specific keyphrases that do not necessarily occur within the document. First, we add a second graph representing the domain and unify it to the topic graph. Second, we define a co-ranking scheme that leverages the new graph. Finally, we redefine the keyphrase selection step for both extracting and assigning keyphrases.

3.2 Unified graph construction

TopicCoRank operates over a unified graph that connects two graphs representing the document topics, the controlled keyphrases and the relations between them (see Fig. 1). The controlled keyphrases are the keyphrases that were manually assigned to training documents. Considering the manually assigned keyphrases as the controlled vocabulary circumvents the need for a manually produced controlled vocabulary and also allows us to further take advantage of the semantic relatonship between the domain-specific (controlled) keyphrases. Because controlled keyphrases are presumably non-redundant, we do not topically cluster them as we do for keyphrase candidates.

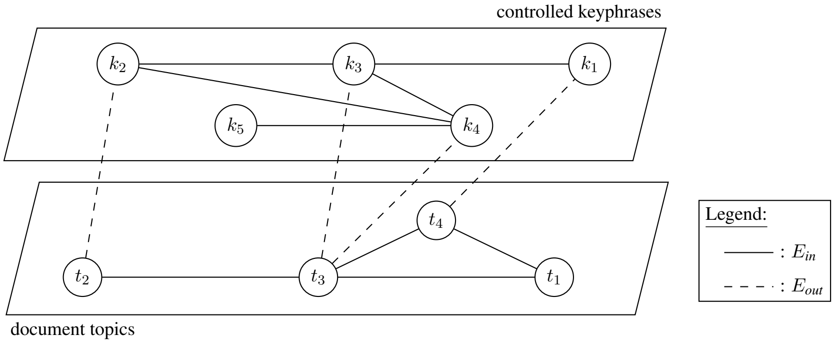


Figure 1: Example of a unified graph constructed by TopicCoRank and its two kinds of edges

Let $G = (V = T \cup K, E = E_{in} \cup E_{out})$ denote the unified graph. Topics $T = \{t_1, t_2, ..., t_n\}$ and controlled keyphrases $K = \{k_1, k_2, ..., k_m\}$ are vertices V connected to their fellows by edges $E_{in} \subseteq T \times T \cup K \times K$ and connected to the other vertices by edges $E_{out} \subseteq K \times T$ (see Fig. 1).

To unify the two graphs, we consider the controlled keyphrases as a category map and connect the document to its potential categories. We create an unweighted edge $\langle k_i, t_j \rangle \in E_{out}$ to connect a controlled keyphrase k_i and a topic t_j if the controlled keyphrase is a member of the topic, i.e. a keyphrase candidate of the topic[2]. We create an edge $\langle t_i, t_j \rangle \in E_{in}$ or $\langle k_i, k_j \rangle \in E_{in}$ between two topics t_i and t_j or two controlled keyphrases k_i and k_j when they co-occur within a sentence of the document or as keyphrases of a training document, respectively. Edges $\langle t_i, t_j \rangle \in E_{in}$ are weighted by the number of times ($w_{i,j}$) topics t_i and t_j occur in the same sentence within the document. Edges $\langle k_i, k_j \rangle \in E_{in}$ are weighted by the number of times ($w_{i,j}$) keyphrases k_i and k_j are associated to the same document among the training documents. Doing so, the weighting scheme of edges E_{in} is equivalent for both topics and controlled keyphrases. This equivalence is essential to ensure that not only controlled keyphrases occurring in the document can be assigned by properly co-ranking topics and controlled keyphrases.

[2] To accept inflexions, such as plural inflexions, we follow Bougouin et al. (2013) and perform the comparison with stems.

3.3 Graph-based co-ranking

TopicCoRank gives an importance score $S(t_i)$ or $S(k_i)$ to every topic or controlled keyphrase using graph co-ranking (see equations 6 and 7). Our graph co-ranking simulates the voting concept based on inner and outer recommendations.

The inner recommendation is similar to the recommendation computed in previous work (Bougouin et al., 2013; Mihalcea and Tarau, 2004; Wan and Xiao, 2008). The inner recommendation R_{in} comes from nodes of the same graph (see equation 8). A topic or a controlled keyphrase is important if it is strongly connected to other topics or controlled keyphrases, respectively.

The outer recommendation influences the ranking of topics by controlled keyphrases and of controlled keyphrases by topics. The outer recommendation R_{out} comes from nodes of the other graph (see equation 9). A topic or a controlled keyphrase gain more importance if it is connected to important controlled keyphrases or an important topic, respectively.

$$S(t_i) = (1 - \lambda_t)\ R_{out}(t_i) + \lambda_t\ R_{in}(t_i) \tag{6}$$

$$S(k_i) = (1 - \lambda_k)\ R_{out}(k_i) + \lambda_k\ R_{in}(k_i) \tag{7}$$

$$R_{in}(v_i) = \sum_{v_j \in E_{\text{in}}(v_i)} \frac{w_{ij} S(v_j)}{\sum\limits_{v_k \in E_{\text{in}}(v_j)} w_{jk}} \tag{8}$$

$$R_{out}(v_i) = \sum_{v_j \in E_{\text{out}}(v_i)} \frac{S(v_j)}{|E_{out}(v_j)|} \tag{9}$$

where v_i is a node representing a keyphrase or a topic. λ_t and λ_k are parameters that control the influence of the inner recommendation over the outer recommendation ($0 \leqslant \lambda_t \leqslant 1$ and $0 \leqslant \lambda_k \leqslant 1$) for the topics and the controlled keyphrases, respectively.

3.4 Keyphrase annotation

Keyphrases are extracted and assigned from the N-best ranked topics and controlled keyphrases, regardless of their nature.

We extract topic keyphrases using the former TopicRank strategy. Only one keyphrase is extracted per topic: the keyphrase candidate that first occurs within the document.

We assign controlled keyphrases only if they are directly or transitively connected to a topic of the document. If the ranking of a controlled keyphrase has not been affected by a topic of the document nor by controlled keyphrases connected to topics, then its importance score is not related to the content of the document and it should not be assigned.

At this step, two variants of TopicCoRank performing either extraction or assignment can be proposed, namely TopicCoRank$_{extr}$ and TopicCoRank$_{assign}$. If keyphrases are only extracted from the topics, we obtain TopicCoRank$_{extr}$. If keyphrases are only assigned from the controlled keyphrases, we obtain TopicCoRank$_{assign}$.

4 Experimental Setup

4.1 Datasets

We conduct our experiments on data from the DEFT-2016 benchmark datasets (Daille et al., 2016)[3] in three domains: linguistics, information Science and archaeology. Table 1 shows the factual information

[3] Data has been provided by the TermITH project for both DEFT-2016 and this work. Parallely, the subset division has been modified for the purpose of DEFT-2016. Therefore, we use the same data as DEFT-2016, but the subset division is different. The subset division we used for our experiences can be found here: `https://github.com/adrien-bougouin/KeyBench/tree/coling_2016/datasets/`

about the datasets. Each dataset is a collection of 706 up to 718 French bibliographic records collected from the database of the French Institute for Scientific and Technical Information[4] (Inist). The bibliographic records contain a title of one scientific paper, its abstract and its keyphrases that were annotated by professional indexers (one per bibliographic record). Indexers were given the instruction to assign reference keyphrases from a controlled vocabulary and to extract new concepts or very specific keyphrases from the titles and the abstracts. Each dataset is divided into three sets: a test set, used for evaluation; a training set (denoted as train), used to represent the domain; and a development set (denoted as dev), used for parameter tuning.

Corpus	**Linguistics**			**Information Science**			**Archaeology**		
	train ⊃ dev		test	train ⊃ dev		test	train ⊃ dev		test
Documents	515	100	200	506	100	200	518	100	200
Tokens/Document	161	151	147	105	152	157	221	201	214
Keyphrases	8.6	8.8	8.9	7.8	10.0	10.2	16.9	16.4	15.6
Missing Keyphrases (%)	60.6	63.2	62.8	67.9	63.1	66.9	37.0	48.4	37.4

Table 1: Dataset statistics. "Missing" represents the percentage of keyphrases that cannot be retrieved within the documents.

The amount of missing keyphrases, i.e. keyphrases that cannot be extracted from the documents, shows the importance of keyphrase assignment in the context of scientific domains. More than half of the keyphrases of linguistics and information science domains can only be assigned, which confirms that these two datasets are difficult to process with keyword extraction approaches alone.

4.2 Document preprocessing

We apply the following preprocessing steps to each document: sentence segmentation, word tokenization and Part-of-Speech (POS) tagging. Sentence segmentation is performed with the PunktSentenceTokenizer provided by the Python Natural Language ToolKit (NLTK) (Bird et al., 2009), word tokenization using the Bonsai word tokenizer[5] and POS tagging with MElt (Denis and Sagot, 2009).

4.3 Baselines

To show the effectiveness of our approach, we compare TopicCoRank and its variants (TopicCoRank$_{extr}$ and TopicCoRank$_{assign}$) with TopicRank and KEA++. For KEA++, we use the thesauri maintained by Inist[6] to index the bibliographic records of Linguistics, Information Science and Archaeology.

4.4 TopicCoRank setting

The λ_t and λ_k parameters of TopicCoRank were tuned on the development sets, and set to 0.1 and 0.5 respectively. This empirical setup means that the importance of topics is much more influenced by controlled keyphrases than other topics, and that the importance of controlled keyphrases is equally influenced by controlled keyphrases and topics. In other words, the domain has a positive influence on the joint task of keyphrase extraction and assignment.

5 Experimental Results

This section presents and analyses the results of our experiments. For each document of each dataset, we compare the keyphrases outputed by each method to the reference keyphrases of the document. From the comparisons, we compute the macro-averaged precision (P), recall (R) and f1-score (F) per dataset and per method.

[4] http://www.inist.fr

[5] 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

[6] Thesauri are available from: http://deft2016.univ-nantes.fr/download/traindev/

5.1 Macro-averages results

Table 2 presents the macro-averaged precision, recall and f1-score in percentage when 10 keyphrases are extracted/assigned for each dataset by TopicRank, KEA++, TopicCoRank$_{extr}$, TopicCoRank$_{assign}$ and TopicCoRank. First, we observe that the assignment baseline KEA++ mostly achieves the lowest performance, which is surprising compared to the performance reported by Medelyan and Witten (2006). The first reason for this observation is that KEA++ is restricted to thesauri entries while most keyphrases are missing within our documents. The second reason is that KEA++ relies on rich thesauri that contain an important amount of semantic relations between the entries, while our (real application) thesauri have a modest amount of semantic relations between the entries.

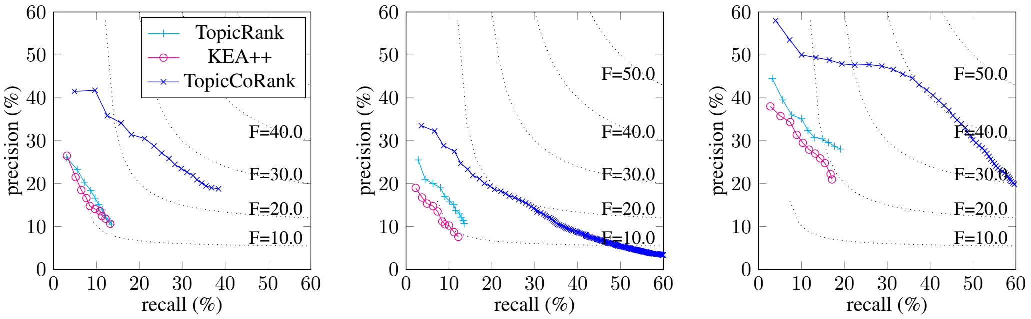
Overall, using graph co-ranking significantly outperforms TopicRank and KEA++. Comparing TopicRank to TopicCoRank$_{extr}$ shows the positive influence of the domain (controlled keyphrases) on the ranking of the topics. TopicCoRank$_{assign}$ outperforms every method, including TopicCoRank$_{extr}$ and TopicCoRank. Controlled keyphrases are efficiently ranked and the predominance of missing keyphrases in the dataset leads to a better performance of TopicCoRank$_{assign}$ over TopicCoRank.

Method	**Linguistics**			**Information Science**			**Archaeology**		
	P	R	F	P	R	F	P	R	F
TopicRank	11.82	13.1	11.9	12.1	12.8	12.1	27.5	19.7	21.8
KEA++	11.6	13.0	12.1	9.5	10.2	9.6	23.5	16.2	18.8
TopicCoRank$_{extr}$	15.9	18.2	16.7†	15.9	16.2	15.6†	39.6	26.4	31.0†
TopicCoRank$_{assign}$	**25.8**	**29.6**	**27.2**†	**19.9**	**20.0**	**19.5**†	**49.6**	**33.3**	**39.0**†
TopicCoRank	24.5	28.3	25.9†	19.4	19.6	19.0†	46.6	31.4	36.7†

Table 2: Results of TopicCoRank and the baselines at 10 keyphrases for each dataset. Precision (P), Recall (R) and F-score (F) are reported in percentages. † indicates a significant F-score improvement over TopicRank and KEA++ at 0.001 level using Student's t-test.

5.2 Precision/recall curves

Additionally, we follow Hasan and Ng (2010) and analyse the precision-recall curves of TopicRank, KEA++ and TopicCoRank. To generate the curves, we vary the number of evaluated keyphrases (cut-off) from 1 to the total number of extracted/assigned keyphrases and compute the precision and recall for each cut-off. Such representation gives a good appreciation of the advantage of a method compared to others, especially if the other methods achieve performances in the *Area Under the Curve* (AUC).



(a) Linguistics (b) Information Science (c) Archaeology

Figure 2: Precision-recall curves of TopicRank, KEA++ and TopicCoRank for each dataset

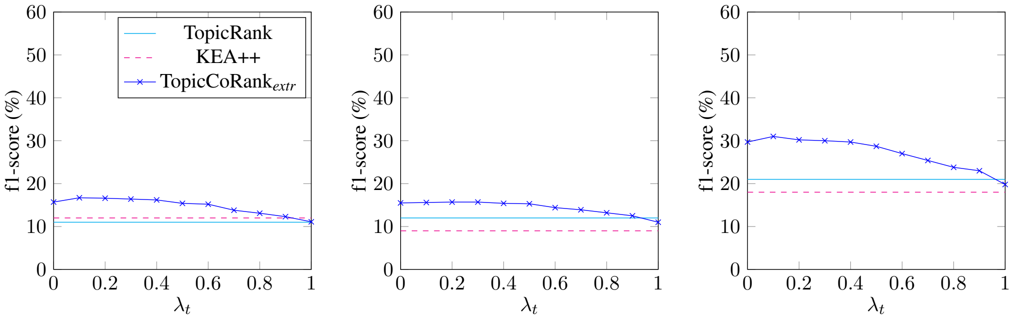
Figure 2 shows the precision/recall curves of TopicRank, KEA++ and TopicCoRank on each dataset. The final recall for the methods does not reach 100% because the candidate selection method does not provide keyphrases that do not occur within the document, as well as candidates that do not fit the POS tag pattern `/(N|A)+/`. Also, because TopicRank and TopicCoRank topically cluster keyphrase candidates and output only one candidate per topic, their final recall is lowered every time a wrong keyphrase is chosen over a correct one from the topic.

We observe that the curve for TopicCoRank is systematically above the others, thus showing improvements in the area under the curve and not just in point estimate such as f1-score. Also, the final recall of TopicCoRank is much higher than the final recall of TopicRank and KEA++.

5.3 Extraction vs. assignment

As TopicCoRank is the first method for simultaneously extracting and assigning keyphrases, we perform an additional experiment that shows to which extent extraction and assignment contribute to the final results. To do so, we show the behavior of the extraction and the assignment depending on the influence of the inner recommendation on the ranking for each (test) document of each dataset.

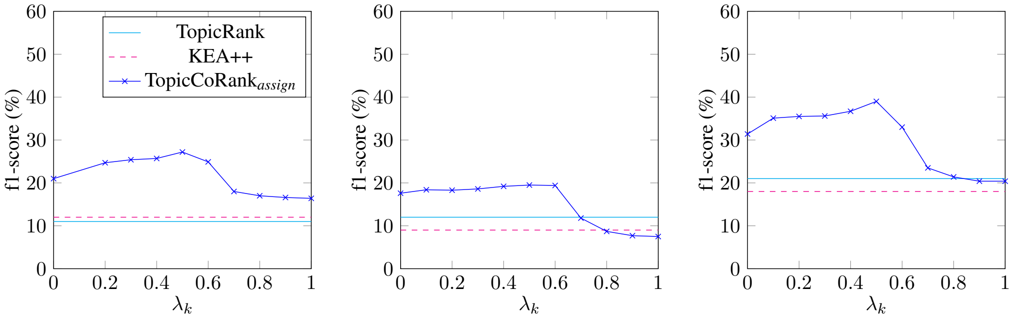
Fig. 3 shows the behavior of TopicCoRank$_{extr}$ when λ_t varies from 0 to 1. When $\lambda_t = 0$, only the domain influences the ranking of the topics. Slightly equivalent to KEA++, TopicCoRank$_{extr}$ with $\lambda_t = 0$ mainly extracts keyphrases from topics connected to controlled keyphrases. When $\lambda_t = 1$, the domain does not influence the ranking and the performance of TopicCoRank$_{extr}$ is in the range of TopicRank's performance. Overall, the performance curve of TopicCoRank$_{extr}$ decreases while λ_t increases. Thus, the experiment demonstrates that the domain has a positive influence on the keyphrase extraction.



(a) Linguistics (b) Information Science (c) Archaeology

Figure 3: Behavior of TopicCoRank$_{extr}$ depending on λ_t ($\lambda_k = 0.5$)

Fig. 4 shows the behavior of TopicCoRank$_{assign}$ when λ_k varies from 0 to 1. When $\lambda_k = 0$, only the document influences the ranking of the controlled keyphrases. As for TopicCoRank$_{extr}$ when $\lambda_t = 0$, TopicCoRank$_{assign}$ is slightly similar to KEA++ when $\lambda_k = 0$. When $\lambda_k = 1$, TopicCoRank$_{assign}$ always outputs the same keyphrases: the ones that are the most important in the domain. The first half of the curve increases, showing that the relations between the controlled keyphrases have a positive influence on the ranking of the controlled keyphrases. Conversely, the second half of the curve decreases. Thus, the sole domain is not sufficient for keyphrase annotation.



(a) Linguistics (b) Information Science (c) Archaeology

Figure 4: Behavior of TopicCoRank$_{assign}$ depending on λ_k ($\lambda_t = 0.1$)

Toucher : le tango des sens. Problèmes de sémantique lexicale **(The French verb 'toucher': the tango of senses. A problem of lexical)**

A partir d'une hypothèse sur la sémantique de l'unité lexicale 'toucher' formulée en termes de forme schématique, cette étude vise à rendre compte de la variation sémantique manifestée par les emplois de ce verbe dans la construction transitive directe 'C0 toucher C1'. Notre étude cherche donc à articuler variation sémantique et invariance fonctionnelle. Cet article concerne essentiellement le mode de variation co-textuelle : en conséquence, elle ne constitue qu'une première étape dans la compréhension de la construction des valeurs référentielles que permet 'toucher'. Une étude minutieuse de nombreux exemples nous a permis de dégager des constantes impératives sous la forme des 4 notions suivantes : sous-détermination sémantique, contact, anormalité, et contingence. Nous avons tenté de montrer comment ces notions interprétatives sont directement dérivables de la forme schématique proposée.

Keyphrases : *Français* (French); *modélisation* (modelling); *analyse distributionnelle* (distributional analysis); *interprétation sémantique* (semantic interpretation); *variation sémantique* (semantic variation); *transitif* (transitive); *verbe* (verb); *syntaxe* (syntax) and *sémantique lexicale* (lexical semantics).

Figure 5: Example of a bibliographic record in Linguistics (`http://cat.inist.fr/?aModele=afficheN&cpsidt=16471543`)

5.4 Qualitative example

To show the benefit of TopicCoRank, we compare it to TopicRank on one of our bibliographic records in Linguistics (see Figure 5). Over the nine reference keyphrases, TopicRank successfully identifies two of the reference keyphrases: "lexical semantics" and "semantic variation". TopicCoRank successfully identifies seven of them: "lexical semantics", "verb", "semantic variation", "French", "syntax", "semantic interpretation" and "distributional analysis".

TopicCoRank mostly outperforms TopicRank because it finds keyphrases that do not occur within the document: "French", "syntax", "semantic interpretation", and "distributional analysis". Some keyphrases, such as "French", are frequently assigned because they are part of most of the bibliographic records of our dataset[7] (48.9% of the Linguistics records contain "French" as a keyphrase); Other keyphrases, such as "semantic interpretation", are assigned thanks to their strong connection with controlled keyphrases occurring in the abstract (e.g. "lexical semantics").

Interestingly, the performance of TopicCoRank is not only better thanks to the assignment. For instance, we observe keyphrases, such as "verb", that emerge from topics connected to other topics that distribute importance from controlled keyphrases (e.g. "semantic variation").

6 Conclusion

In this paper, we have proposed a co-ranking approach to performing keyphrase extraction and keyphrase assignment jointly. Our method, TopicCoRank, builds two graphs: one with the document topics and one with controlled keyphrases (training keyphrases). We designed a strategy to unify the two graphs and rank by importance topics and controlled keyphrases using a co-ranking vote. We performed experiments on three datasets of different domains. Results showed that our approach benefits from both controlled keyphrases and document topics, improving both keyphrase extraction and keyphrase assignment baselines. TopicCoRank can be used to annotate keyphrases in scientific domains in a close way of professional indexers.

Acknowledgments

This work was supported by the French National Research Agency (TermITH project – ANR-12-CORD-0029) and by the TALIAS project (grant of CNRS PEPS INS2I 2016, `https://boudinfl.github.io/talias/`).

References

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

[7] Yet, TopicCoRank does not assign "French" to every bibliographic records.

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

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 543–551, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

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.

Béatrice Daille, Sabine Barreaux, Florian Boudin, Adrien Bougouin, Damien Cram, and Amir Hazem. 2016. Indexation d'articles scientifiques présentation et résultats du défi fouille de textes deft 2016. In *Actes de 12e Défi Fouille de Texte (DEFT)*, pages 1–12, Paris, France, July. Association pour le Traitement Automatique des Langues.

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.

Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill Manning, and Eibe Frank. 1999. Improving Browsing in Digital Libraries with Keyphrase Indexes. *Decision Support Systems*, 27(1):81–104.

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 (COLING)*, pages 365–373, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, Maryland, June. Association for Computational Linguistics.

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

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, 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(EMNLP)*, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiyuan Liu, Xinxiong Chen, Yabin Zheng, and Maosong Sun. 2011. Automatic Keyphrase Extraction by Bridging Vocabulary Gap. In *Proceedings of the 15th Conference on Computational Natural Language Learning*, pages 135–144, Stroudsburg, PA, USA. Association for Computational Linguistics.

Olena Medelyan and Ian H Witten. 2006. Thesaurus Based Automatic Keyphrase Indexing. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 296–297. ACM.

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 (EMNLP)*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Gerard Salton, Andrew Wong, and Chungshu Yang. 1975. A Vector Space Model for Automatic Indexing. *Communication ACM*, 18(11):613–620, November.

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.

Fan Zhang, Lian'en Huang, and Bo Peng. 2013. WordTopic-MultiRank: A New Method for Automatic Keyphrase Extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 10–18, Nagoya, Japan, October. Asian Federation of Natural Language Processing.