WikiRank:Improving Keyphrase Extraction Based on Background Knowledge

Yang Yu, Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
{yangyu,vince}@hlt.utdallas.edu

Abstract

Keyphrase is an efficient representation of the main idea of documents. While background knowledge can provide valuable information about documents, they are rarely incorporated in keyphrase extraction methods. In this paper, we propose WikiRank, an unsupervised method for keyphrase extraction based on the background knowledge from Wikipedia. Firstly, we construct a semantic graph for the document. Then we transform the keyphrase extraction problem into an optimization problem on the graph. Finally, we get the optimal keyphrase set to be the output. Our method obtains improvements over other state-of-art models by more than 2% in F1-score.

Keywords: Keyphrase Extraction, Knowledge Graph, Semantic Graph

1. Introduction

As the amount of published material rapidly increases, the problem of managing information becomes more difficult. Keyphrase, as a concise representation of the main idea of the text, facilitates the management, categorization, and retrieval of information. Automatic keyphrase extraction concerns "the automatic selection of important and topical phrases from the body of a document". Its goal is to extract a set of phrases that are related to the main topics discussed in a given document(Hasan and Ng, 2014).

Existing methods of keyphrase extraction could be divided into two categories: supervised and unsupervised. While supervised approaches require human labeling, at the same time needs various kinds of training data to get better generalization performance, more and more researchers focus on unsupervised methods.

Traditional methods of unsupervised keyphrase extraction mostly focus on getting information of document from word frequency and document structure(Hasan and Ng, 2014), however, after years of attempting, the performance seems very hard to be improved any more. Based on this observation, it is reasonable to suspect that the document itself possibly cannot provide enough information for keyphrase extraction task.

To get good coverage of the main topics of the document, Topical PageRank (Liu et al., 2010) started to adopt topical information in automatic keyphrase extraction. The main idea of Topical PageRank is to extract the top topics of the document using LDA, then sum over the scores of a candidate phrase under each topic to be the final score. The main problems with Topical PageRank are: First, The topics are too general. Second, since they are using LDA, they only classify the words to several topics, but don't know what the topics exactly are. However, the topical information we need for keyphrase extraction should be precise. As shown in Figure 1, the difference between a correct keyphrase *sheep disease* and an incorrect keyphrase *incurable disease* could be small, which is hard to be captured by rough topical categorization approach.

To overcome the limitations of aforementioned approaches, we propose WikiRank, an unsupervised automatic keyphrase extraction approach that links semantic meaning to text

The key contribution of this paper could be summarized as follows:

1. We leverage the topical information in knowledge bases to improve the performance of keyphrase extraction.

2. We model the keyphrase extraction as an optimization problem, and provide the corresponding solution as well as a pruning approach to reduce the complexity.

2. Existing Error Illustration with Example

Figure 1 shows part of an example document[1]. In this figure, the gold keyphrases are marked with bold, and the keyphrases extracted by the TextRank system are marked with parentheses. We are going to illustrate the errors exist in most of present keyphrase extraction systems using this example.

Overgeneration errors occur when a system correctly predicts a candidate as a keyphrase because it contains a word that frequently appears in the associated document, but at the same time erroneously outputs other candidates as keyphrases because they contain the same word(Hasan and Ng, 2014). It is not easy to reject a non-keyphrase containing a word with a high term frequency: many unsupervised systems score a candidate by summing the score of each of its component words, and many supervised systems use unigrams as features to represent a candidate. To be more concrete, consider the news article in Figure 1. The word Cattle has a significant presence in the document. Consequently, the system not only correctly predict *British cattle* as a keyphrase, but also erroneously predict *cattle industry*, *cattle feed*, and *cattle brain* as keyphrases, yielding overgeneration errors.

[1]Document from DUC-2001 Dataset *AP900322-0200 Government Boosts Spending to Combat Cattle Plague*

[2]Prefix "wiki" represents the namespace "https://en.wikipedia.org/wiki/"

(**Mad cow disease**) has killed 10,000 cattle, restricted the **export** market for Britain's (cattle industry)
wiki:Bovine_spongiform_encephalopathy wiki:Death wiki:Cattle wiki:Export wiki:Market_(economics) wiki:British_Empire wiki:Agribusiness

and raised fears about the safety of eating beef. The **government** insists the disease poses only a remote risk to human
wiki:Fear wiki:Safety wiki:Eating wiki:Beef wiki:Government wiki:Disease wiki:Risk

health, but scientists still aren't certain what causes the disease or how it is transmitted ... (**Mad cow disease**), or
wiki:Health wiki:Scientist wiki:Causality wiki:Transmission_(medicine) wiki:Bovine_spongiform_encephalopathy

Bovine Spongiform Encephalopathy, or **BSE**, was diagnosed only in 1986. The symptoms are very much like **scrapie**, a
wiki:Bovine_spongiform_encephalopathy wiki:Bovine_spongiform_encephalopathy wiki:Symptom wiki:Scrapie

(**sheep disease**) which has been in Britain since the 1700s. The (incurable disease) eats holes in the brains of its victims;
wiki:Sheep wiki:Great Britain wiki:Cure wiki:Cannibalism wiki:Human_Brain

in late stages a sick animal may act skittish or stagger drunkenly ... The **government** banned the use of sheep offal in
wiki:Disease wiki:Animal wiki:Government wiki:Ban_(law) wiki:Offal

(cattle feed) in June 1988, and later banned the use of (cattle brain), spleen ... has proposed a ban on **exports** of (**British**
wiki:Brain wiki:Spleen wiki:Ban_(law) wiki:Export wiki:United_Kingdom

cattle) older than 6 months ... has complained of "**BSE** hysteria" in the media and has insisted that the risk of the (disease
wiki:Cattle wiki:Bovine_spongiform_encephalopathy wiki:Mass_hysteria wiki:Mass_media wiki:Risk wiki:Disease

passing) to humans is "remote." ... known as (Creutzfeldt Jakob disease). About two dozen cases were reported in (Britain
wiki:Human wiki:Creutzfeldt-Jakob_disease wiki:Great_Britain

last year).

Figure 1: Part of the Sample Document [2]

Bold: Gold Keyphrase In parentheses: Keyphrase generated by TextRank algorithm Underlined: Keyphrase annotated to Wikipedia Entity by TagMe

Redundancy errors occur when a system correctly identifies a candidate as a keyphrase, but at the same time outputs a semantically equivalent candidate (e.g., its alias) as a keyphrase. This type of error can be attributed to the failure of a system to determine that two candidates are semantically equivalent. Nevertheless, some researchers may argue that a system should not be penalized for redundancy errors because the extracted candidates are in fact keyphrases. In our example, *bovine spongiform encephalopathy* and *bse* refer to the same concept. If a system predicts both of them as keyphrases, it commits a redundancy error.

Infrequency errors occur when a system fails to identify a keyphrase owing to its infrequent presence in the associated document. Handling infrequency errors is a challenge because state-of-the-art keyphrase extractors rarely predict candidates that appear only once or twice in a document. In the *Mad cow disease* example, the keyphrase extractor fails to identify *export* and *scrapie* as keyphrases, resulting in infrequency errors.

3. Proposed Model

The WikiRank algorithm includes three steps: (1) Construct the semantic graph including concepts and candidate keyphrases; (2)(optional) Prune the graph with heuristic to filter out candidates which are likely to be erroneously produced; (3) Generate the best set of keyphrases as output.

3.1. Graph Construction

3.1.1. Automatic Concept Annotation

This is one of the crucial steps in our paper that connects the plain text with human knowledge, facilitating the understanding of semantics. In this step, we adopt *TAGME* (Ferragina and Scaiella, 2010) to obtain the underlying concepts in documents.

TAGME is a powerful topic annotator. It identifies meaningful sequences of words in a short text and link them to a pertinent Wikipedia page, as shown in Figure 1. These links add a new topical dimension to the text that enable us to relate, classify or cluster short texts.

3.1.2. Lexical Unit Selection

This step is to filter out unnecessary word tokens from the input document and generate a list of potential keywords using heuristics. As reported in (Hulth, 2003), most manually assigned keyphrases turn out to be noun groups. We follow (Wan and Xiao, 2008a) and select candidates lexical unit with the following Penn Treebank tags: NN, NNS, NNP, NNPS, and JJ, which are obtained using the Stanford POS tagger (Toutanova et al., 2003), and then extract the noun groups whose pattern is zero or more adjectives followed by one or more nouns. The pattern can be represented using regular expressions as follows

$$(JJ) * (NN|NNS|NNP|NNPS)+$$

where JJ indicates adjectives and various forms of nouns are represented using NN, NNS and NNP .

3.1.3. Graph building

We build a semantic graph $G = [V; E]$ in which the set of vertices V is the union of the concept set C and the candidate keyphrase set P—i.e., $V = P \cup C$. In the graph, each unique concept $c \in C$ or candidate keyphrase $p \in P$ for document d corresponds to a node. The node corresponds to a concept c and the node corresponds to a candidate keyphrase p are connected by an edge $(c, p) \in E$, if the candidate keyphrase p contains concept c according to the annotation of *TAGME*. Part of the semantic graph of the sample document is shown in Figure 2. Concepts corresponding to 2 are shown in Table 1.

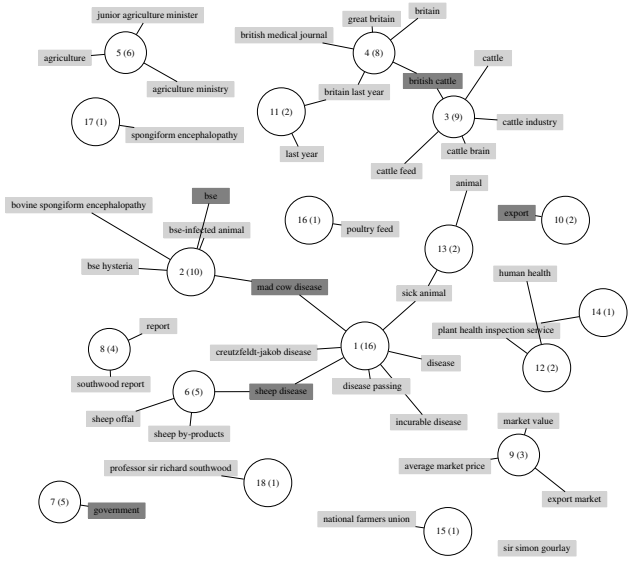

Figure 2: Part of the Semantic Graph of the Sample Document

Circle: Concept Rectangle: Candidate Keyphrase

Dark Rectangle: Gold Keyphrase

#	Concept	Frequency
1	Disease	16
2	Bovine spongiform encephalopathy	10
3	Cattle	9
4	Great Britain	8
5	United States Department of Agriculture	6
6	Sheep	5
7	Government	5
8	Report	4
9	Market (economics)	3
10	Export	2
11	Last year	2
12	Health	2
13	Animal	2
14	Animal and Plant Health Inspection Service	1
15	National Farmers Union of England and Wales	1
16	Poultry feed	1
17	Transmissible spongiform encephalopathy	1
18	Professor	1

Table 1: Part of the Concepts Annotated from the Sample Document

3.2. WikiRank

3.2.1. Optimization Problem

According to (Liu et al., 2010), good keyphrases should be relevant to the major topics of the given document, at the same time should also have good coverage of the major topics of the document. Since we represent the topical information with concepts annotated with *TAGME*, the goal of our approach is to find the set Ω consisting of k keyphrases, to cover concepts (1) as important as possible (2) as much as possible.

Let w_c denote the weight of concept $c \in C$. We compute w_c as the frequency c exists in the whole document d. To quantify how good the coverage of a keyphrase set Ω is, we compute the overall score of the concepts that Ω contains.

Consider a subgraph of G, G_{sub}, which captures all the concepts connected to Ω. In G_{sub}, the set of vertices V_{sub} is the union of the candidate keyphrase set Ω, and the set Adj_Ω of concepts that nodes in Ω connect to. The set of edges E_{sub} of G_{sub} is constructed with the edges connect nodes in Ω with nodes in Adj_Ω.

We set up the score of a concept c in the subgraph G_{sub} as following:

$$S(c) = \sum_{i=0}^{deg(c)} \frac{w_c}{2^i} \tag{1}$$

where w_c is the weight of c as we defined before, and $deg(c)$ is the degree of c in the subgraph G_{sub}. Essentially, $deg(c)$ is equal to the frequency that concept c is annotated in the keyphrase set Ω.

The optimization problem is defined as:

$$\begin{aligned} \max_{\Omega} \quad & \sum_{c \in Adj_\Omega} S(c) \\ \text{s.t.} \quad G_{sub} &= [V_{sub}; E_{sub}] \\ V_{sub} &= \Omega \cup Adj_\Omega \\ E_{sub} &= \{(c,p) | p \in \Omega, c \in Adj_\Omega\} \\ Adj_\Omega &= \{c | c \in \textstyle\sum_{p \in \Omega} Adj(p)\} \\ |\Omega| &\leq k \end{aligned} \tag{2}$$

The goal of the optimization problem is to find the candidate keyphrase set Ω, such that the sum of the scores of the concepts annotated from the phrases in Ω is maximized.

3.2.2. Algorithm

We propose an algorithm to solve the optimization problem, as shown in Algorithm 1. In each iteration, we compute the score s_p for all candidate keyphrases $p \in |P|$ and include the p with highest score into Ω, in which s_p evaluates the score of concepts added to the new set Ω by adding p into Ω.

3.3. Approximation Approach with Pre-pruning

In practice, computing score for all the candidate keyphrases is not always necessary, because some of the candidates are very unlikely to be gold keyphrase that we can remove them from our graph before applying the algorithm to reduce the complexity.

In this section, we introduce three heuristic pruning steps that significantly reduces the complexity of the optimization problem without reducing much of the accuracy.

Step 1. Remove the candidate keyphrase p from original graph G, if it is not connected to any concept.

The intuition behind this heuristic is straightforward. Since our objective function is constructed over concepts, if a candidate keyphrase p doesn't contain any concept, adding it to Ω doesn't bring any improvement to the objective function, so p is irrelevant to our optimization process. Pruning p would be a wise decision.

	DUC			Inspec			ICSI			Nus		
	P	R	F score	P	R	F score	P	R	F score	P	R	F score
SingleRank	26.21	24.45	25.30	25.21	24.10	24.64	3.42	2.49	2.88	0.23	0.98	0.37
Topical PageRank	27.33	23.92	25.51	25.58	24.31	24.93	3.98	2.68	3.20	0.64	1.38	0.87
Our System	28.72	26.44	27.53	28.14	25.97	27.01	4.71	3.96	4.30	7.27	12.16	9.10

Table 2: The Result of our System as well as the Reimplementation of SingleRank and Topical PageRank on four Corpora

Algorithm 1 Keyphrase Generalization

Input:
$|C|, P, W = \{w_1, \ldots, w_{|C|}\}$
k: ▷ Size of output keyphrase set
$M_{|P|\times|C|}$: ▷ Adjacency matrix
Output:
Ω ▷ The set of selected keyphrases
Initialization:
$\Omega \leftarrow \emptyset$
$S = \{s_1 \leftarrow 0, \ldots, s_{|P|} \leftarrow 0\}$

while $|\Omega| < k$ **do**
  for $p = 1$ to $|P|$ **do**
    $s_p \leftarrow 0$
    for $c = 1$ to $|C|$ **do**
      if $M_{p,c} \neq 0$ **then**
        $s_p = s_p + w_c$
      end if
    end for
  end for
  $q \leftarrow \arg\max_{q=1\ldots|P|}\ s_q$
  $\Omega \leftarrow \Omega \cup \{P_q\}$
  for $c = 1$ to $|C|$ **do**
    if $M_{q,c} \neq 0$ **then**
      $w_c \leftarrow w_c/2$
    end if
  end for
end while
return Ω

Step 2. Remove the candidate keyphrase p from original graph G, if it is only connected to one concept that only exists once in the document

If a candidate keyphrase contains fewer concepts, or the concepts connects to it barely exist in the document, we think this candidate keyphrase contributes less valuable information to the document. In practice, there are numerous (c, p) pairs in graph G that is isolated from the center of the graph. We believe they are irrelevant to the major topic of the document.

Step 3. For a concept c connecting to more than m candidate keyphrases, remove any candidate keyphrase $p \in Adj(c)$ which (1)Does not connect to any other concept. AND (2)The ranking is lower than mth among all candidate keyphrases connect to c.(In practice, m is usually 3 or 4.)

According to equation 1, if there are already m instances of concept c in the G_{sub}, adding the $m+1$th instance of c will only contribute $\frac{w_c}{2^m}$ to $S(c)$. At the same time, among all the candidate keyphrases connected to concept c, our optimization process always chooses the ones that connect to other concepts as well over the ones that do not connect to any other concept. Combining these two logic, a candidate satisfying the constrains of Step 3 is not likely to be picked in the best keyphrase set Ω, so we can prune it before the optimalization process.

4. Experiments and Results

4.1. Corpora

The **DUC-2001** dataset (Over, 2001), which is a collection of 308 news articles, is annotated by (Wan and Xiao, 2008b).

The **Inspec** dataset is a collection of 2,000 abstracts from journal papers including the paper title. This is a relatively popular dataset for automatic keyphrase extraction, as it was first used by (Hulth, 2003) and later by Mihalcea and (Mihalcea and Tarau, 2004) and (Liu et al., 2009).

The **NUS Keyphrase Corpus** (Nguyen and Kan, 2007) includes 211 scientific conference papers with lengths between 4 to 12 pages. Each paper has one or more sets of keyphrases assigned by its authors and other annotators. The number of candidate keyphrases that can be extracted is potentially large, making this corpus the most challenging of the four.

Finally, the **ICSI Meeting Corpus** (Janin et al., 2003), which is annotated by Liu et al. (2009a), includes 161 meeting transcriptions. Unlike the other three datasets, the gold standard keys for the ICSI corpus are mostly unigrams.

4.2. Result

For comparing with our system, we reimplemented SingleRank and Topical PageRank. Table 2 shows the result of our reimplementation of SingleRank and Topical PageRank, as well as the result of our system. Note that we predict the same number of phrase ($k = 10$) for each document while testing all three methods.

The result shows our result has guaranteed improvement over SingleRank and Topical PageRank on all four corpora.

5. Conclusion and Future Work

We proposed an unsupervised graph-based keyphrase extraction method WikiRank. This method connects the text with concepts in Wikipedia, thus incorporate the background information into the semantic graph and finally construct a set of keyphrase that has optimal coverage of the concepts of the document. Experiment results show the method outperforms two related keyphrase extraction methods.

We suggest that future work could incorporate more other semantic approaches to investigate keyphrase extraction

task. Introducing the results of dependency parsing or semantic parsing (e.g., OntoUSP) in intermediate steps could be helpful.

6. Bibliographical References

Ferragina, P. and Scaiella, U. (2010). Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA. ACM.

Hasan, K. S. and Ng, V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273, Baltimore, Maryland, June. Association for Computational Linguistics.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, Z., Li, P., Zheng, Y., and Sun, M. (2009). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, Z., Huang, W., Zheng, Y., and Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of Empirical Methods for Natural Language Processing*, pages 404–411.

Nguyen, T. D. and Kan, M.-Y. (2007). Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers*, ICADL'07, pages 317–326, Berlin, Heidelberg. Springer-Verlag.

Over, P. (2001). Introduction to DUC-2001: An intrinsic evaluation of generic news text summarization systems. In *Proceedings of the 2001 Document Understanding Conference*.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (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*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wan, X. and Xiao, J. (2008a). Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 969–976, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wan, X. and Xiao, J. (2008b). Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 855–860. AAAI Press.