Keyword Extraction using Multiple Novel Features

Shaohui YANG*, Bo ZHANG, Shun LI, Chaohui YU, Qianting HAO
Information Center, Ministry of Environmental Protection, Beijing 100029, China

Abstract

In this paper, we propose a novel approach for keyword extraction. Different from previous keyword extraction methods, which identify keywords based on the document alone, this approach introduces Wikipedia knowledge and document genre to extract keywords from the document. Keyword extraction is accomplished by a classification model utilizing not only traditional word based features but also features based on Wikipedia knowledge and document genre. In our experiment, this novel keyword extraction approach outperforms previous models for keyword extraction in terms of precision-recall metric and breaks through the plateau previously reached in the field.

Keywords: Natural Language Processing; Keyword Extraction

1 Introduction

Compared with a document, keyword provides a compact information representation and therefore helps people quickly capture the main idea of a document without spending time on the details.

Traditional keyword extraction methods [1] just focus on the document alone to define features to construct a feature space for machine learning methods. Common features based on document itself include word frequency, word position, word length, proper noun and so on. The limitation of traditional feature definition is that they only use explicit information shown in the document and have no access to the world knowledge possessed by humans. Thus, previous approaches using word based features cannot perform well when meeting facts not mentioned in the training set. Moreover, document genre also influences keyword extraction because some genre related words are more possible to be keywords under their corresponding genre. For example, if the genre is politics, document words with political meaning such as nationalism and justice, tend to appear as keywords.

In this paper, we derive several novel features based on outside knowledge and document genre. These new features in conjunction with traditional features based on document alone construct a feature space for machine learning methods in keyword extraction. For outside knowledge, we consider Wikipedia as a source of background knowledge in this paper. For document genre,

*This research is funded by the National Natural Science Foundation of China (NSFC41071243).
*Corresponding author.
Email address: yang.shaohui@mep.gov.cn (Shaohui YANG).
we first identify the document genre using a classifier, and then derive a genre oriented feature based on the similarity degree between document words and genre related words. The augmented feature space provides keyword extraction with additional valuable information.

The contributions of this paper are twofold. First, we propose feature generation technique to define new features based on Wikipedia. Second, document genre is taken into account in feature generation. The rest of the paper is organized as follows. Sec. 2 provides a description of related work. Our approach is described in Sec. 3 and Sec. 4. The evaluation of our algorithm is shown in Sec. 5. The conclusion is in Sec. 6.

2 Related Work

Traditional keyword extraction methods only use information explicitly contained in a document such as word frequency and word position. In [2], a simple approach based on word frequency is proposed for keyword extraction. The TextRank algorithm introduced in [3] uses three statistical properties including \( tf \times idf \), distance, and key phrase frequency. The method proposed in [5] uses \( tf \times idf \), word position, POS of a word as well as the linkage between adjacent words as word features for keyword extraction. In [1], lexical chain features are used.

Recently, people have started to use Wikipedia for keyword extraction. Most closely related to our work here is the project called “Wikify!” [6], which uses the link structure of Wikipedia to derive a novel word feature for keyword extraction. [7] utilized article titles and the link structure of Wikipedia to construct a semantic graph for keyword extraction. Unlike their approach, when we extract keywords, we utilize not only information explicitly contained in the document such as word frequency, position, and length but also the background knowledge of the document, which is acquired from Wikipedia via analyzing the inlink, outlink, category, and infobox information of the documents related articles in Wikipedia.

3 Acquiring Document Background Knowledge using Wikipedia

We propose to acquire the background knowledge of a document using Wikipedia. Overall speaking, our Wikipedia based document background knowledge acquisition process consists of two main steps: 1) given a document, to generate a query for retrieving the document’s background knowledge through searching the Wikipedia corpus; and 2) to extract key information from the Wikipedia search results to derive the background knowledge of the input document.

3.1 Generating a Wikipedia search query

Our keyword extraction method relies on Wikipedia to provide the background knowledge for the input document for making more informed keyword extraction decisions. Since Wikipedia being an online multilingual encyclopedia is a giant document corpus, which consists of 4.3 million articles in English, it is not practical nor necessary to try to utilize all the knowledge facts contained in Wikipedia. Instead, we only attempt to retrieve almost relevant part of the knowledge. To
this end, given an input document, we generate a Wikipedia search query to retrieve the most related knowledge for the input document from the Wikipedia corpus.

To generate a recalling query to find the most relevant background knowledge of the input document using Wikipedia search, we construct the query based on the key facts conveyed in the document. This is done through selecting important content words from the input document. Specially, we first apply a modified version of the TextRank algorithm [3] to detect important sentences in a document. Our modification of the original TextRank algorithm is in how the pairwise sentence similarity is calculated. Instead of word spelling matching, we measure the similarity between the two sentences through Eq. (1):

\[
\text{Similarity}(S_i, S_j) = \sum_{W_p \in S_i} \sum_{W_q \in S_j} \sigma_1(W_p, W_q) \frac{\log(|S_i|)}{\log(|S_i|) + \log(|S_j|)},
\]

where \(|S_i|\) denotes the word length of the sentence \(S_i\) and \(\sigma_1(W_p, W_q)\) is the pairwise word similarity between the two words \(W_p\) and \(W_q\). Here we use the algorithm proposed in [4] to measure the pairwise word similarity \(\sigma_1\), which is based on WordNet and takes into account all the potential semantic relationships existing between two words.

Given a few key sentences selected from the input document through the above process, we then perform stop-words removal and word stemming for these sentences. The remaining words constitute our Wikipedia search query.

3.2 Searching the Wikipedia corpus

Once the Wikipedia search query for the input document is generated, we call on the full text search engine, Zettair [8], to retrieve articles from the Wikipedia XML corpus. The search results are returned as a ranked list of Wikipedia articles and their corresponding similarity (relevance) scores with respect to the search query. We denote the set of Wikipedia articles retrieved after submitting a Wikipedia search query as \(\Pi\). The \(r\)-th Wikipedia article in the search result set \(\Pi\) is denoted as \(p_r\). The similarity score of the article \(p_r\) as returned by the Zettair search engine is denoted as \(z(p_r)\). We denote the size of \(\Pi\) as \(N\), i.e., the number of articles in the set. Next we explain how to extract the most useful background information from the Wikipedia search results.

3.3 Extracting background knowledge

3.3.1 Extracting the inlink title set and the outlink title set for a Wikipedia article

Wikipedia is organized as a hyperlinked text corpus, which allows readers to browse and navigate through its content following the link structure. An inlink points from another Wikipedia article to the current Wikipedia article while an outlink points from the current Wikipedia article to another Wikipedia article.

For each article \(p_r\) in the Wikipedia search result set \(\Pi\), we can produce its corresponding inlink set, \(IL(p_r)\) and outlink set, \(OL(p_r)\). For either set, we extract all the titles of the articles contained in the set, which gives us an inlink title set \(IT(p_r)\) and an outlink title set, \(OT(p_r)\) respectively.
3.3.2 Extracting the category set for a Wikipedia article

Category is another important type of information in Wikipedia, which appears at the bottom of a Wikipedia article to indicate the topics covered in the article. Category information is organized as a graph structure in Wikipedia. Users can navigate through such category graph to locate Wikipedia articles of their interests even when they do not know the exact titles of the articles. Each Wikipedia article may be associated with a set of categories. We represent all the categories that a Wikipedia article \( p_r \) is associated with as the category set of the article, which is denoted as \( C(p_r) \).

3.3.3 Extracting the infobox attribute value set for a Wikipedia article

Each infobox is generated using a certain infobox template. An infobox template typically carries several attributes for describing the key facts of the subject in a Wikipedia article. The values for each attribute field are manually filled by human editors to form the infobox for a Wikipedia article.

To make a template widely useable, editors often choose some common words as attribute names. Hence attribute names themselves carry very little entity specific information. The most revealing texts are the attribute values. In view of this fact, we extract all the infobox attribute values of a Wikipedia article \( p_r \), and organize them into an attribute value set \( IV(p_r) \).

4 Keyword Extraction using Wikipedia

Our keyword extraction problem is a classification problem, in which a word in a document is to be classified as either a keyword or not. We use a learning based approach to tackle this classification problem.

4.1 Word features

4.1.1 Deriving word features using Wikipedia

1) Word Inlink and Outlink Features We derive a word inlink feature \( S_I \) and an outlink feature \( S_O \) using the inlink and outlink information in the corresponding Wikipedia search result set \( \Pi \) of the input document. For each document word \( x_i \), its inlink and outlink features, \( S_I(x_i) \), \( S_O(x_i) \), are computed as follows:

\[
S_I(x_i, \Pi) = \frac{\sum_{p_r \in \Pi} \left[ z(p_r) \cdot \sum_{k \in IT(p_r)} \sigma_1(x_i, k) \right]}{\sum_{p_r \in \Pi} z(p_r) \cdot |IT(p_r)|}; (2)
\]

\[
S_O(x_i, \Pi) = \frac{\sum_{p_r \in \Pi} \left[ z(p_r) \cdot \sum_{k \in OT(p_r)} \sigma_1(x_i, k) \right]}{\sum_{p_r \in \Pi} z(p_r) \cdot |OT(p_r)|}; (3)
\]
where \( z(p_r) \) is the score of the Wikipedia article \( p_r \), \( IT(p_r) \) represents the inlink title set for the Wikipedia article \( p_r \), \( OT(p_r) \) represents the outlink title set of the Wikipedia article \( p_r \), \( |X| \) is the size of the set \( X \), and \( \sigma_1(x_i, k) \) is the pairwise word similarity between the words \( x_i \) and \( k \) where \( k \) is an entry either in the inlink title set \( IT(p_r) \) or the outlink title set \( OT(p_r) \). By the above definition, the more similar a word \( x_i \) is to the entries in the inlink title set \( IT(p_r) \) or the outlink title set \( OT(p_r) \), the higher the word’s inlink or outlink feature value will be.

2) Word Category Feature We introduce a word category feature \( S_C \) using the category information of articles in the Wikipedia search set \( \Pi \). This feature measures the similarity between a word \( x_i \) and all the category entries in the category set \( C(p_r) \) of a Wikipedia article \( p_r \). Formally, we define the word category feature \( S_C \) as follows:

\[
S_C(x_i, \Pi) = \frac{\sum_{p_r \in \Pi} \left[ z(p_r) \cdot \sum_{c \in C(p_r)} \sigma_2(x_i, c) \right]}{\sum_{p_r \in \Pi} z(p_r) \cdot |C(p_r)|},
\]

where \( z(p_r) \) is the score of the Wikipedia article \( p_r \), \( C(p_r) \) represents the category set of the Wikipedia article \( p_r \), \( |C(p_r)| \) is the set size, and \( \sigma_2(x_i, c) \) is the category similarity between the words \( x_i \) and \( c \) based on the Wikipedia category graph [9].

3) Word Infobox Feature We also use Wikipedia’s infobox information to derive a word infobox feature \( S_F \). The word infobox feature can be calculated as follows:

\[
S_F(x_i, \Pi) = \frac{\sum_{p_r \in \Pi} \left[ z(p_r) \cdot \sum_{k \in IV(p_r)} \sigma_1(x_i, k) \right]}{\sum_{p_r \in \Pi} z(p_r) \cdot |IV(p_r)|},
\]

where \( IV(p_r) \) is the infobox value set of the Wikipedia article \( p_r \), others have been given in the above.

4.1.2 Word-document genre fitness feature

Here we introduce the word-document genre fitness feature to characterize a word’s likelihood to show up in a document’s keywords from a genre agreement’s perspective.

Given the document genre similarity \( \theta(d_i, d_j) \) [13], we can now define the word-document genre fitness \( \gamma(w_i, d_j) \) between an arbitrary document \( d_j \) and document word \( w_i \) as follows. We use a multi-genre document corpus [14], which consists of 1539 webpages that have been manually classified into 20 genres such as “personal”, “informative”, “journalistic”, “commercial/promotional”, “scientific”, “entertainment” etc. Given \( d_j \), we find 300 articles from our corpus which yield the highest document genre similarities with \( d_j \). We denote these articles as \( d_{j,1}, \ldots, d_{j,300} \) respectively. For each \( d_{j,k} (k = 1, \ldots, 300) \), we then collect all the non-stop words in its title. For each such word, we derive a weighted word occurrence count, where the weight is based on document genre similarity. Without loss of generality, assuming for a word \( w_i \), among the above 300 articles, it occurs in the titles of the articles \( d_{j,1}, \ldots, d_{j,n} \). The genre weighted word occurrence of \( w_i \) is calculated as \( WO(w_i) = \sum_{k=1}^{n} \theta(d_j, d_{j,k}) \). Suppose all the non-stop words that occur in the titles
of these 300 articles are \( w_1, \cdots, w_m \). Then we can derive the genre weighted word frequency for a word \( w_k \) to show up in the title of a document as
\[
WF(w_k) = \frac{WO(w_k)}{\sum_{t=1}^{n} WO(w_t)}.
\]
Based on genre weighted word frequencies of all the title words \( w_1, \cdots, w_m \), we can now define the word-document genre fitness feature as:
\[
\gamma(w_i, d_j) = \sum_{k=1}^{m} WF(w_k) \sigma_1(w_k, w_i),
\]
(6)
where \( \sigma_1(w_k, w_i) \) measures the semantic similarity between the words \( w_k \) and \( w_i \), which was introduced earlier in Sec. 3.1.

4.1.3 Other common word features

We also use the following common word features which can be directly derived from the input document: word frequency feature, word position feature, specific name feature, relative word length feature and conclusion sentence feature.

4.2 Keyword extraction through a learning based approach

Given word features prepared in the above, we can then apply a machine learning based approach to extract keywords. Our training set consists of full text documents and their headlines. For simplicity, all the non-stop words in a documents headline are considered as the keywords of the document. In our current experiment, we downloaded 817 articles from http://news.google.com/ to establish the training set, where the word lengths for the majority of articles are between 300 to 400 words. We then apply the support vector machine (SVM) method to the keyword extraction task.

5 Experiment

Because there is no commonly available data set for keyword extraction, we first construct our own keyword extraction groundtruth data set through collecting 200 recent online articles posted on the BBC and CNN websites using a web crawler. We then ask 10 master’s students in the computer science department to extract keywords manually from these articles. After carrying out this manual keyword extraction process, we construct a data set consisting of 200 articles and their corresponding keywords.

To quantitatively study the effectiveness of our novel word features for keyword extraction, we conducted a series of controlled experiments where different feature subsets are used in our method. We report the performance of these variants of our method in Table 1, the results of which clearly show the necessity and effectiveness of engaging all the novel word features introduced in this paper for keyword extraction.

We also compare the performance of our algorithm and some peer methods such as TF×IDF [2], Yaoo! Term Extraction [18], Wikify! [3], community detection based keyword extraction algorithm [7] by widely used precision, recall and F-rate measurements. These results confirm the advantage of our method for keyword extraction.
Table 1: Performance comparison of our method using different feature sets

<table>
<thead>
<tr>
<th>feature set</th>
<th>precision</th>
<th>recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>All features</td>
<td>0.456</td>
<td>0.513</td>
<td>0.483</td>
</tr>
<tr>
<td>No link features</td>
<td>0.298</td>
<td>0.323</td>
<td>0.310</td>
</tr>
<tr>
<td>No category feature</td>
<td>0.342</td>
<td>0.361</td>
<td>0.351</td>
</tr>
<tr>
<td>No genre feature</td>
<td>0.336</td>
<td>0.341</td>
<td>0.338</td>
</tr>
<tr>
<td>No infobox feature</td>
<td>0.381</td>
<td>0.372</td>
<td>0.376</td>
</tr>
<tr>
<td>Common word features</td>
<td>0.275</td>
<td>0.281</td>
<td>0.278</td>
</tr>
</tbody>
</table>

Table 2: Performance comparison between different keyword extraction methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFxIDF</td>
<td>0.210</td>
<td>0.312</td>
<td>0.251</td>
</tr>
<tr>
<td>Yahoo! Term Extraction</td>
<td>0.231</td>
<td>0.362</td>
<td>0.282</td>
</tr>
<tr>
<td>Wikify!</td>
<td>0.285</td>
<td>0.421</td>
<td>0.340</td>
</tr>
<tr>
<td>Communities detection</td>
<td>0.312</td>
<td>0.435</td>
<td>0.373</td>
</tr>
<tr>
<td>Our method</td>
<td>0.456</td>
<td>0.513</td>
<td>0.483</td>
</tr>
</tbody>
</table>

Finally, we tested the top-k precision of our method for keyword extraction, and then compared the performance of our method with that of existing keyword extraction methods. The results are illustrated in Fig. 1. These results very positively confirm the advantage of our method for keyword extraction.

![Fig. 1: Top-k precision on test data](image)

6 Conclusion

In this paper, we propose some novel word features for keyword extraction through acquiring the background knowledge of a document using Wikipedia. To define these word features, we
explore Wikipedia article’s inlink, outlink, category, and infobox information. We also introduce a word-document genre fitness feature based on a multi-genre document corpus to observe the keyword extraction bias imposed by the genre of the document. Experimental results have proved that using these novel word features, we can achieve superior performance in keyword extraction than other state-of-the-art algorithms.

References