Automatic Keyphrase Extraction based on NLP and Statistical Methods

Martin Dostal and Karel Ježek

Department of Computer Science and Engineering, Faculty of Applied Sciences
University of West Bohemia, Plzeň, Czech Republic
{madostal, jezek_ka}@kiv.zcu.cz

Abstract. In this article we would like to present our experimental approach to automatic keyphrase extraction based on statistical methods and Wordnet-based pattern evaluation. Automatic keyphrases are important for automatic tagging and clustering because manually assigned keyphrases are not sufficient in most cases. Keyphrase candidates are extracted in a new way derived from a combination of graph methods (TextRank) and statistical methods (TF*IDF). Keyword candidates are merged with named entities and stop words according to NL POS (Part Of a Speech) patterns. Automatic keyphrases are generated as TF*IDF weighted unigrams. Keyphrases describe the main ideas of documents in a human-readable way. Evaluation of this approach is presented in articles extracted from News web sites. Each article contains manually assigned topics/categories which are used for keyword evaluation.

Keywords: keyphrase extraction, Wordnet, TextRank, TFIDF, NLP

1 Introduction

In this paper we would like to present our experimental approach to automatic keywords and keyphrase extraction. Our approach is very useful in cases where we don't have manual keywords assigned by the author or where these keywords are not sufficient. Our approach builds on ideas from TextRank [1] and RAKE [2] with a combination of statistical (TF*IDF) and NLP methods.

Keywords are defined as a sequence of one or more words and provide a compact description of a document's content. Keywords are often used to define queries within information retrieval systems because they are easy to define, remember, and share. Keywords are usually corpus independent and can be applied across multiple different systems. For example, the Phrasier [3] system lists documents related to a primary document's keywords and supports keyword anchors as hyperlinks between documents. The Keyphind system [4] uses keywords as the basic building block for an IR system especially for summarization and clustering tasks. Hulth [5] (2004) describes Keegle, a system that provides extracted keywords for web pages found by a Google search engine.

Keyphrases consist of two or more keywords and named entities. In our approach, keyphrases consist of keywords, named entities and stop words. Stop words can be an important part of a keyphrase, which increase the readability and intelligibility of a phrase in natural language. This idea was inspired by the RAKE system for automatic keyword extraction from individual documents.

2 Related graph algorithms

In this section, we would like to discuss graph-based ranking algorithms, especially Google's PageRank [6] and its implementation for text document processing.

Google's PageRank [6] is the first and most popular graph-based ranking algorithm which has been successfully used by social networks, citation analysis, and link-structure analysis of the World Wide Web. This algorithm is a way of deciding on the importance of a node within a graph. The importance of a node is evaluated based on global information recursively drawn from the graph. The graph node is important when it is often recommended by other nodes. In this special case, if other web documents contain links in the form of URI to this one.

The TextRank graph-based algorithm is a ranking model for graphs extracted from text documents. The text is split into tokens which represent the nodes of the graph. Nodes are connected with weighted edges based on the lexical or semantic relations, for example. TextRank algorithms use a co-occurrence relation controlled by the distance between word occurrences within a sliding window of maximum N words. The best results were achieved for a maximum of two words which correspond with N-gram based algorithms.

3 Algorithm for automatic keyword extraction

In this section, we would like to discuss our approach to automatic keyword extraction for documents. TextRank can be used in individual documents without any other knowledge. Graph nodes ranking is made upon co-occurrences of tokens and the score of the other nodes with edges to the current one. In our algorithm, the TF*IDF score is used for better text token evaluation instead of the measure based only on n-grams. The next idea is based on two versions of the keyword characteristic. The keyword extraction problem can be divided into:

- Individual keyword extraction – individual words with a special and important meaning, generally in the form of a noun or named entity.
- Keyphrase extraction and derivation – phrases contain two or more keywords and other information in a human-readable form. This phrase can contain verbs and stop words for better readability. Short phrases can be joined by their co-occurrence percentage number.

Our algorithm can be divided into three main steps:

1. Text preprocessing
2. Keyword extraction

3. Keyphrase extraction

During the text preprocessing phase, the article's content is divided into tokens and non-significant characters are removed. Named entities are recognized by the elementary method: the first upper-case letter with corpus-based statistic is good enough for this recognition. The tokens are divided by their POS tag and generally only nouns and adjectives can be declared as potential keywords. The next idea is to choose only common words instead of unique ones. Keywords are often used for clustering and a keyword is useless when it is assigned to only a few articles. This is the reason why we remove tokens with low frequency and set up rules for general nouns. There is no such rule for named entities. The remaining tokens and named entities are declared as keywords candidates. We calculate the TF*IDF score only for these keyword candidates.

The keyword extraction phase contains only the method for removing useless candidates whose TF*IDF score is lower than 1/5 of a maximal value. This boundary can be changed by the number of requested automatic keywords.

The keyphrase extraction part can be described by these steps:

- NLP method – interesting n-grams are chosen. The choice is based on their POS tag patterns and the corpus frequency is counted only for these n-grams. These n-grams can be marked as keyphrase candidates.
- A score of importance is counted for each keyphrase candidate. This score contains the n-gram corpus frequency and TF*IDF score for each word. The score is used for document keyphrase selection.
- Derivation – keyphrase candidates are merged with named entities or individual keywords if their co-occurrence is significant for this document.

Used POS patterns:

A) POS patterns for 3-grams:
   - (N or named entity), (V or A or stop word), (N or named entity)
   - 3x (named entity)
     example: Tim Berners Lee

B) POS patterns for 2-grams
   - A, (N or named entity)
   - 2x (named entity)
     example: Bill Gates

Used tags:

- N – noun,
- V – verb,
- A – adjective,
- “stop word“ – a word from stop list,
- “named entity” – a potential named entity based on simple patterns like word in the middle of the sentence with first capital letter.

Keywords and keyphrases are ordered by their TF*IDF score and only the most important keywords and keyphrases are used. The number of used items naturally depends on the requested number of these items. This number can be chosen directly, or by percentage measure from the score of the best item.

4 Evaluation

To evaluate performance, we tested our system against a collection of newspaper articles about technology that were extracted from the Web. These articles contain manually assigned keywords from a controlled dictionary. These keywords were chosen by the author of the article. Such keywords are marked as topics in our case. These topics cover the article content very sketchily and capture only the basic idea of the article. For example, the content of the article contains the word “Google”, but the manually assigned topic was “Google Inc”. This can be enough for article classification, but for automatic keyword evaluation it is a further challenge.

We have decided that our news corpus is a collection from the “real world” and if the results are satisfying, it would be usable for other general data collections. Another reason was that this approach is very experimental and we had no better article collection for evaluation. Initially, we thought that these results would not be satisfactory, but we were surprised by their high relevance. The size of the data sets was chosen based on the number of the articles that were available for the each subset evaluation.
We have used three data sets:

A) Corpus of 500 articles with a small number of manually assigned topics (600 different topics) by the author.
B) Corpus of 50 random articles with a small number of manually assigned topics. Each article contains approximately 3 manual topics.
C) Corpus of 50 random articles with manually assigned topics (by author) and expanded, by 2-3 another human annotators, to other important topics covered by the article. Each article contains approximately 5 manual topics.

The statistics of precision and recall for part A) are shown in Table 1. These values are calculated for a different boundary configuration of accepted keywords. This threshold is set as the percentage difference from the score of the most important keyword. Precision and recall are reduced because of topic classification difficulties. There are about 600 various topics in this data set and some of them are very similar for the human annotator, but not for our topic classification module.

The topic classification is done only based on the name of the topic, so for example: topics “Google Inc.” and “Google operating system” have the same score for these automatic keywords: “Google” and “Android”.

Table 1. Precision and recall for the corpus of 500 articles.

Boundary	**1%**	**10%**	**30%**	**50%**	**70%**	**90%**
Precision	13.2%	18.9%	27%	31.8%	38.2%	40.8%
Recall	46.1%	33%	22.6%	16.5%	12.8%	10.53%

The statistics of precision and recall for part B) are shown in Table 2. This corpus contains 50 random original articles. The main difference between evaluation A) and B) is in the number of classification topics. There are about 120 topics used for classification so the precision and recall are not distorted as much as in part A).

Table 2. Precision and recall for the corpus of 50 articles.

Boundary	**1%**	**10%**	**20%**	**30%**	**40%**	**50%**	**70%**	**90%**
Precision	30%	38.6%	42.4%	48%	50%	49.4%	50.7%	55.2%
Recall	49%	33%	26.9%	23.7%	22%	18.5%	13.6%	12.9%

The statistics of precision and recall for part C) are shown in Table 3. This corpus contains 50 articles with 2-3 additional human-annotated topics. The total number of manually added topics is about 5.

Table 3. The corpus of 50 articles with additional human annotations.

Boundary	**1%**	**10%**	**20%**	**30%**	**40%**	**50%**	**70%**	**90%**
Precision	37.4%	47.4%	53.9%	55.8%	59.12%	59.4%	60.6%	64%
Recall	54.6%	35.9%	29.3%	23.7%	22.4%	18.8%	14.1%	13.5%

5 Conclusion and future work

The proposed approach seems to be efficient enough to be comparable with other automatic keyword extraction systems. For example, the RAKE system achieved 33.7% precision with 41.5% recall and the undirected TextRank achieved 31.2% precision with 43.1% recall. Our approach achieved 37.4% precision and 54.6% recall for a small corpus with expanded number of annotations, including the problem of keyword generation and automatic clustering. We can assume that precision and recall will be a little bit lower for a bigger corpus. The most significant feature of the corpus is the number of exact manual annotations which are used for performance tests.

In the future, we would like to compare our approach with other methods on their data corpuses. These corpuses were not available at this moment so we had to use our data collection for the first evaluation tests. Automatic keywords will be used for mapping the Linked Data topics to the articles and graph-based knowledge extraction.

References

1. Mihalcea R. and Tarau P. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004* (ed. Lin D and Wu D), pp. 404–411. Association for Computational Linguistics, Barcelona, Spain.
2. Rose S., Engel D., Cramer N., Cowley D. *Automatic keyword extraction from individual documents.* In Text mining applications and theory 2010, pp. 3-19.
3. Jones S. and Paynter G. Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications. *Journal of the American Society for Information Science and Technology.*
4. Gutwin C, Paynter G, Witten I, Nevill-Manning C and Frank E. *Improving browsing in digital libraries with keyphrase indexes.* Decision Support Systems 27(1–2), 81–104, 1999.
5. Hulth A. *Combining machine learning and natural language processing for automatic keyword extraction.* Stockholm University, Faculty of Social Sciences, Department of Computer and Systems Sciences (together with KTH), 2004.
6. Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report. Stanford InfoLab, 1999.