# EXTRACTION OF TOPICAL QUERY FROM MULTI-DOMAIN DOCUMENT COLLECTION

**Arseni Anisimovich**

*IHS Global, Minsk State Linguistic University*
*Minsk, Belarus*
*E-mail: arseni.anisimovich@gmail.com*

Finding documents related to a search result might be a challenging task when possible relevant document are hidden behind a search engine. This paper describes a semi-supervised method for extraction a topical query describing main topic of the document that can be used to link two or more databases with help of a search engine.

*Keyword*: extraction, keyphrase extraction, tf-idf, topical query, document similarity, relevant document extraction.

## Introduction

Keyphrase might be defined as a phrase or a search term that is made up from multiple keywords, or a specific combination of keywords. Keyphrases reflect high-level content of given document and help readers to quickly conclude about document relevancy regarding their informational needs. Also, keyphrases are a fast and accurate measure of documents similarity, so they may provide a solid base for a list of information retrieval tasks: clustering and categorization of documents, document summarization, topic search, query suggestion and search results refinement, metadata and taxonomy generation.

Only a small amount of documents are provided with keyphrases, because authors assign keywords only when they are explicitly asked to do so and, as it was highlighted in [4], authors are unlikely to use valid keywords if they belong to socially or professionally disregarded set of words like ‘garbage’. Also, keyphrase extraction is a very labor-intensive task; therefore ways of extracting keyphrases using automatic techniques are of a high scientific interest.

Any document can be provided with keyphrases using two different approaches: keyphrase assignment and keyphrase extraction [9]. Keyphrase assignment (as a part of text categorization task) presumes, that potential keyword appears in a predefined controlled (manually or automatically) vocabulary – set of categories. Second approach, keyphrase extraction, is not restricted by any list of possible keyphrases; to the contrary, any phrase from a document can be extracted as a keyphrase. From machine learning point of view, first approach is strictly supervised, since a set of training documents is required to train a classifiers for each category, keyphrase extraction, on the contrary, leaves choice between supervised, unsupervised and semi-supervised keyphrase extraction.

## Related work

Works regarding keyphrase extraction start appearing in late 1990s when World Wide Web became a solid source of information and electronic documents reached some critical mass that should be managed in order to get required data. One of the most mentioned works in the field is *GenEx* algorithm by Peter Turney [9]. Genetic algorithm *Genitor* tunes

Turney's Extractor algorithm, which uses 12 parameters to determine whether or not the phrase is a keyphrase. Reported precision (expressed in average precision +/- Standard Deviation) is promising: 20-29% and that algorithm outperformed commercial solutions of its time (namely, Microsoft Word'97 and Verity's Search 97).

Same year appears KEA (keyword extraction algorithm) [2], the main idea of which is understanding keyphrase extraction problem as a classification problem and applying Naïve Bayes classificator to every given n-gram in text. Set of features, used to train the classifier is much smaller than Turney's set: only 2 features are used, distance of first phrase occurrence and tf–idf parameter of given n-gram. Reported precision is around 20-28% percent, however authors prove, that a Naïve Bayes classifier can be used for the task with a relatively small training corpus (around 20 documents with statistically unimportant changes of precision with corpus growth). This approach received its development in [10] with using a POS-tagger for phrase extraction with a very limited set of possible POS-chains for keyphrase candidates. Applying POS-tagger gave significant growth of precision (more than 5% for top-5 keyphrases).

Third representative of keyphrase extraction of late 90s uses *LocalMaxs* algorithm to rank uninterrupted n-grams [5]. Underlying idea in that work is evaluation of *glue* between words of an n-gram, present in document. Two metrics are used to measure likelihood of n n-gram to be keyphrase: Symmetrical Conditional Probability and Fair Dispersion Point Normalization. Second term indicates, that in the proposed solution every n-gram of size > 2 is represented as a set '*pseudo-bigrams*' to make the process more transparent and easy from computational point of view. Though reported precision is very high (81% for Portuguese and 77% for English [6]), the evaluation criterion is very soft, because any 'relevant' expression is considered positive example, if it is often met in analyzed text corpora. This approach extract many relevant multi-word units, however, they are not keyphrases representing document topics.

Another approach belongs to a latter period, when amount of user-generated content becomes of a great value to Information Retrieval scientists. Raise of Wikipedia as a free source of human knowledge provided an enormous amount of basic data in every subject domain. One representative is [7]. Work focuses on extracting keyphrase clusters based on semantic relatedness of terms using Wikipedia cross-linking of articles. Reported precision and recall are, respectively, 46.1% and 67.7%, the paper of same authors also reported higher results (52% precision and 73% recall, [8]). Term '*keyphraseness'*, used in this paper, prefers proper nouns as more likely keyphrases, which is confusing, as proper nouns may not reflect the sense of document, but its Named Entity metadata.

## Topical query extraction algorithm

The main goal of our research was to connect two databases with documents of scientific and technical subject domains. When user is provided with a document from one database we wanted to suggest relevant documents from another database that should be accessed using a search engine. Since we could not use methods like bag-of-words and similar approaches to process every document in two databases, extraction of a topic in a form of search query was necessary. Topic extraction is a sub-task of keyphrase extraction.

In our research we used no learning algorithm and took advantage of all data available. In order to extract most likely query topic of the document (*topical query*) we ranked all extracted queries with a score from 0 to 1 and then selected n best queries provided to search engine and/or user. Since we used an uncontrolled document collection and a manually crafted development corpus we may characterize our approach as semi-supervised. All process of topical query extraction will be described below and is as follows:

1. Extract all noun phrases and similar entities from document.
2. Collect all similar phrases into single block.
3. Rank every block by its relevance.

For noun phrase extraction we use our linguistic processor [11], which provides us with results of linguistic analysis, including POS-tagging, syntactic and semantic parsing. Syntactic structures were converted into noun phrases (a phrase which has a noun or indefinite pronoun as its head word) that were ranked.

Phrases in English can be expressed using different word order (*ATTR+MAIN* is almost equal to *MAIN* of *ATTR*, e. g. *stone wall* or *wall of stone*). But meaning behind words and words to rank remain the same, so for all phrases in the document, we create sets of phrases with same meaningful words. From every set we choose the show form that is the most suitable to the language and is more query-like (for example, users tend to be more laconic in queries and type those that have as few words as possible).

Extracted candidate phrases (containing words $t_1$ … $t_n$) from a document (D) in a document collection (C) were assigned with topical weight (W) using following formula:

$$W(t_1 \dots t_n) = L(n) * \sum_{i=1}^{n} T(t_i) \qquad (1)$$

Parameter T in (1) is a tf–idf (total frequency – inverted document frequency) parameter [3] that has proved to be a quite accurate measure of word ranking among a document collection. tf parameter is the number of word occurrences in document, idf is a number of documents containing given word. We use sum of phrase words tf–idfs normalized by sum of raw tf–idfs of all meaningful words in document and final formula for word's T parameter in (1) is:

$$T(t, D, C) = \frac{tf\,(t,D)*idf\,(t,C)}{\sum_{i=1}^{n} tf\,(t_i,D)*idf\,(t_i,C)} \qquad (2)$$

Another ranking criterion that proved itself very useful in our task is L, possibility of a phrase with length of n words to be a topical query. We obtained that distribution from our manually crafted development corpus containing 400 documents (see table 1).

*Table 1*

**Probability of keyphrase of length N to be topical query**

| Keyphrase Length | Keyphrase probability |
|---|---|
| 1 | 0,062678063 |
| 3 | 0,293447293 |
| 6 | 0,051282051 |

We also used positional criterion when comparing relative weights of key phrases. We presumed, that title of a document, when available, reflects main topic of the document and key terms (*t*), extracted from title, are ranked higher than terms extracted only from the body of a given document. This approach doesn't exclude valid key phrases from the body of document that is proved by the fact, that extracted queries did match title only in 67% of cases in a large multi-domain corpus.

## Development and evaluation corpus

As any classification or information extraction task, keyphrase extraction requires a corpus for evaluation and, in case of machine learning algorithms, training. However, the task of building such corpus is very ambiguous, because there is no strict understanding, what is a keyphrase. Usually, keyphrases or even keywords aren't assigned by every author to every document. Keyphrases aren’t always present in document’s body as a substring, thus they cannot be extracted.

Even academic and scientific papers share this problem, for example, proceedings of COLING 2012 contain keywords for articles, but, from 1138 documents processed, only 530 (46.57%) have keywords and only 78% of keyphrases are met in the document as a substring. We must indicate that it is a very high value for substring match, because keywords in our corpus (part of a dataset – 9896 documents) are assigned only in 18.77% of cases (1858 documents) and the substring match measure is 50.23% (see table 2).

Metadata considering keywords for a webpage cannot be used as a valid corpus because of everlasting battle between search engine optimization (SEO) enthusiasts and page ranking engineers, when keyword becomes a weapon and the position in search results becomes the prize. Keywords for webpages, if assigned, indicate none of the topic of document, its meaning, or its summary, but only terms and queries, which bring more audience and thus more ad-money.

*Table 2*

**Assigned keywords in different corpuses**

| | Size | Docs with keyphrases | Keyphrase matching text |
|---|---|---|---|
| **COLING2012** | 1138 docs | 46.57% | 78% |
| **Our corpus** | 9896 docs (part) | 18.77% | 50.23% |

However, since our task was different from simple keyphrase extraction, we needed a manually crafted corpus to tune parameters. We used 1.5 million documents collection as IDF source and 400 documents development corpus with assigned topics (describing topic of document with several topics for a document possible) as development data to tune parameters and ensure that chosen tf–idf formula provides valid results.

## Results evaluation and future work

For testing purpose, we extracted 18,000 random topical queries from which 100 were tested from search results broadening point of view by independent testing group. Three testing parameters were introduced: (a) amount of search results, brought by a topical query, (b) subjective conformity of query to the title of a document, (c) amount of factoid results to each query. Each query then was ranked with score from 0 (the worst) to 5.5 (the best). 18% of queries brought no results, and 81% of queries were considered as acceptable from reflecting document topic point of view. Overall, 69% of queries provided us with valid topical query that brought decent amount of results relevant to the topic of a document, which we assume to be a high value for the task.

However, since the task is not simply extracting keywords, our results on COLING2012 corpus are very low in the task of extraction of author-assigned keywords: accuracy is only 8%. In comparison, KEA algorithm scored twice as much – 15.58% of accuracy when trained on training part COLING data set. COLING2012 results evaluation, though, indicate that our approach extracts valid key phrases that describe the topic of research, but are not present in the list of keywords, when KEA extracts all kind of substrings including web addresses. We used similar approach to different set of documents, including patents and tech news. Extracted results prove consistency of method in the task of extracting document topic or suggested key phrases.

Deficiencies of this approach are closely linked with its ranking method. Though we extracted topical keyphrases with high accuracy, the tf–idf matrix is only applicable to a similar document collection and should be as up-to-date as possible. And updating that information will result in out-of-date results for previously processed documents. Also, length distribution (L parameter in (1)) is a subjective criterion that affects choice of top ranked keyphrase and requires manually crafted corpus.

Obvious way of excluding the influence of growing document collections and their statistical information can be document-centric approach to keyphrase extraction instead of collection-centric approach. RAKE algorithm [1] provides simple yet quite effective unsupervised document-centric approach to extraction of keyphrases. In future works we suggest to focus on unsupervised techniques that will take advantage of semantic analysis of a document.

## Bibliography

1. *Berry M. W., Cogan J.* Text Mining. Applications and Theory. Padstow, Cornwal: TJ International Ltd, 2010.
2. *Eibe F., Gordon W. Paynter, Ian H. Witten, Carl Gutwin, Craig G. Nevill-Manning* / Domain-specific keyphrase extraction // Proceedings of the 16th international joint conference on Artificial intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. P. 668–673.
3. *Salton G., Buckley C.* Term-weighting approaches in automatic text retrieval// Information Processing and Management: an International Journal. 1988. V. 24. № 5. P. 513–523.
4. *Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, Craig G. Nevill-Mannin* / KEA: practical automatic keyphrase extraction // Proceedings of the fourth ACM conference on Digital libraries. New York, NY, USA: ACM, 1999. P. 254–255.
5. *Joaquim Ferreira da Silva., Lopes Gabriel Pereira.* Extracting Multiword Terms from Document Collections // Proceedings of the VExTAL: Venezia per il Trattamento Automatico delle Lingue. Venezia, Italy: Springer Berlin Heidelberg, 1999. P. 22–24.
6. Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units / Joaquim Ferreira da Silva, Gaël Dias, Sylvie Guilloré, José Gabriel Pereira Lopes // Progress in Artificial Intelligence. Springer Berlin Heidelberg, 1999. P. 113–132.
7. *Grineva M., Grinev M., Lizorkin D.* Effective Extraction of Thematically Grouped Key Terms From Text // AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0. Stanford, CA, United States: AAAI Press, 2009. P. 29–44.
8. *Grineva M., Grinev M., Lizorkin D.* Extracting Key Terms From Noisy and Multi-theme Documents // Proceedings of the 18th international conference on World wide web. New York, NY, USA: ACM, 2009. P. 661–670.
9. *Turney P.* Learning to Extract Keyphrases from Text. 1999.
10. *Wu, Yi-fang Brook, Quanzhi Li, Razvan Stefan Bot, and Xin Chen* / Domain-specific Keyphrase Extraction // Proceedings of the 14th ACM international conference on Information and knowledge management. New York, NY, USA: ACM, 2005. P. 283–284.
11. *Чеусов А. В.* Разработка алгоритмов и технологии построения многоязычного базового лингвистического процессора: дис. ... канд. техн. наук: 05.13.17. Минск, 2013.