Linköping University

Text Mining Project

732A92

Autumn Term 2017

Automatic Keyword Extraction from Patent Documents

Author
Arian Barakat

Examiner
Marco Kuhlmann

Contents

"If I have seen further it is by standing on ye shoulders of Giants".

Sir Isaac Newton

1 Introduction

1.1 Motivation

Patents are legal documents with a set of exclusive rights that are granted by a sovereign state to an inventor or assignee for a limited period of time. In exchange for such an exclusivity, the assignee is asked to give up a detailed public disclosure of the invention but also present related technical and scientific background that forms the state-of-the-art basis for the invention [5]. It can arguably be claimed that patents are usually more detailed and exhaustive than scientific papers and that the body of knowledge formed by the patents, for that reason, plays a vital role in today's society in terms of technological progression.

According to World Intellectual Property Organisation (WIPO), approximately 2.9 million patents were filed during 2015, an estimated increase of 7.8 percent from the previous year [1]. With the number of documents reaching such quantities, it can hardly be overstated that the activity of patent information retrieval is no easy task to manage, especially for the layman with little to no training in the art of patent search. A task that is growing harder by the year as documents pile up in an accumulative manner.

A patent search can be conducted with different objectives in mind, such as state-of-the-art search, patentability or freedom to operate. Although the complexity of the search may vary depending on the objective, the process of finding relevant documents still remains as an iterative and time-consuming process. Using keywords is one of the main techniques for many, if not all, of the mentioned search intentions. However, finding relevant keywords can often be a difficult task as the process may involve multiple documents as well as unfamiliar vocabulary. By introducing a system that automatically extracts relevant keywords from a set of documents, the user may achieve time-savings and valuable insights, expert as a non-expert.

1.2 Aim

This project in text mining aims to develop an unsupervised algorithm for automatic extraction of keywords from patent data that could be used as

input for ad-hoc searches as well as for standing queries. The project also aims to contribute to the text mining community by proposing an efficient graph-based ranking algorithm, HarmonicRank. The ranking algorithm is designed with a modular structure in mind, with the components being based on key attributes derived from graph information and frequency measures extracted from the patent portfolio. [1]

1.3 Delimitation

The problem of finding relevant keywords can be approached in many ways. This project, however, has limited itself to the paradigm of unsupervised learning to avoid the need for pre-labeled data. Furthermore, the experiments conducted in this project has been limited to the use of patent abstracts as text input. Patent documents contain a wide range of valuable information in addition to abstracts and this information can, with advantage, be used to possibly obtain a relatively more coherent set of keywords.

1.4 Outline

This report starts with a brief presentation of related theory and research, followed by a description of the developed *Portfolio2Keyword* and *HarmonicRank* algorithm. The data, as well as methods used for both the experiments and evaluation of the algorithm, will be introduced in section 4 with the results and discussion presented in the subsequent section. Last but not least, in section 6, the reader will be given conclusions and suggestions for further research.

[1] For a demonstration of the algorithm, visit github.com/arianbarakat/portfolio2keyword

2 Theory and Related Work

2.1 Premise

The intuitive idea behind keyword extraction is to identify a set of words that jointly represent a corpus as good as possible. A set of keywords should, in theory and preferably, be able to provide a concise description of a corpus' content but also be fit for use as features for document categorization, clustering or for quantifying semantic similarity with other documents. This means that words that occur rarely or only in a handful of documents in a corpus are less desired. In many of the common techniques used for Information Retrieval (IR) applications, it is often preferred to identify words that discriminate documents in a corpus in order to achieve higher precision. This conceptual idea is for instance utilized by the TF-IDF algorithm, where words that occur in a few documents are assigned higher weights through the *inverse document frequency* (IDF) component. This approach of favoring discriminating words can be viewed as a direct conflict to the goal of keyword extraction and many of these common techniques can, for that reason, be considered as inappropriate for identifying keywords.

The literature suggests that the process of keyword extraction typically occurs in two steps; (1) extraction of candidate keywords using some heuristics to keep the number of candidates at a minimum and (2) ranking these candidates using supervised or unsupervised learning. Under the umbrella of supervised learning, methods such as frequency measures, reformulating the problem into a classification-problem or using external knowledge-databases have been proposed [3]. However, one obvious disadvantage of these methods is the need of predefined "correct" keywords, which in many cases might not be available or feasible to curate. Alternatively, by using the paradigm of unsupervised learning, it is to possible to identify the underlying structure of a corpus, and thereby extracting keywords, without the assistance of pre-labeled/training data.

2.2 Candidate Keyword Generation

The first step of extracting keywords from a corpus is to enumerate a set of candidate keywords. The obvious solution would be to consider all words in the corpus as possible keywords, however, from a computationally and content-bearing point of view this is not a reasonable approach. One way of keeping the set of candidates at a minimum is to use some heuristics rules. Typical heuristics are the reduction of inflectional form of words (lemmatization/stemming), removal of stop-words and punctuation but also to only

include words with certain Part-Of-Speech-tags (POS-tags). The idea behind POS-tag filtering is that word-types of a certain kind are more relevant than others depending on the domain. [3]. In patent documents, such word-types could be adjectives, verbs or nouns as patents often describe different properties and behavior of things.

Although it is conventional to only use linguistic stop-words such as *the, of* and *is*, it can be beneficial to also include domain-specific stop-words in order to avoid spurious candidate keywords. In the domain of intellectual property, such words could for instance be *method* or *claim*.

2.3 Graph Representation

The process of finding relevant keywords involves intuitively the step of determining the importance of a particular word in relation to other words from a set of candidates. This can also be extended by claiming that candidates that occur together with other candidate keywords from the corpus might be a good set of keywords. A natural way of representing this concept is through the mathematical structure of a graph model. Let us formally define an undirected graph $G = (V, E)$, where the candidate keywords form the vertices/nodes (V) and the word co-occurrences the edges (E). By defining an undirected graph we are assuming that the order of how the words co-occur (direction of the edges) is not of interest and thereby implying that edges (v, u) and (u, v) are equivalent, thus $(v, u) = (u, v)$. To demonstrate such a representation, let us use the following text as an example:

> "Text mining, also referred to as text data mining, roughly equivalent to text analytics, is the process of deriving high-quality information from text."

Figure 1 present a graphical representation of the example text above. The text has been exposed to normalization and removal of stop-words. Heuristics such as lemmatization and stemming has not been applied to this particular example for easier understanding how the text maps onto a graphical model, however, these heuristics can preferably be used to obtain the base form of a word. As seen in the figure, words that are listed as candidates form the nodes whereas the edges and the corresponding edge label indicate a co-occurrence of two candidates and how often this co-occurrence occur in the text.

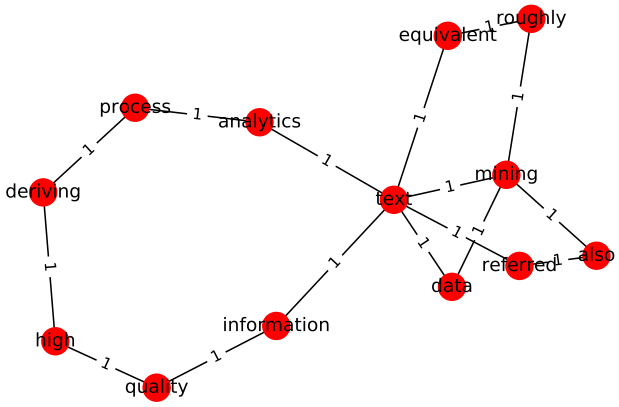

Figure 1: Graph representation of example text

The approach of mapping text into the abstract structure of nodes and edges enables us to model the underlying structure and meaningful relations between words in the corpus. Most importantly, this also allows us to deploy graph-based algorithms to extract significant words as keywords.

2.4 Ranking

2.4.1 Measures and Metrics

Quantifying the importance of a node in a graph is not a new problem and this is often denoted as *centrality* in the context of graph theory or network analysis. The importance of a node is often application-dependent and the definition of centrality can, therefore, have different meanings. As a result, many centrality measures have been developed and it has been recognized that a centrality measure that is optimal for one application can very well be sub-optimal for other applications.

One of the simplest and common measures of centrality is the so-called *degree centrality*, which is defined as the number edges connected to a node. The idea is that nodes that have relatively more connections are more influential compared to other, less connected, nodes as more information flows through them. To illuminate this, let us have look at the example from the previous section, figure 1. From the figure, it can be seen that the most influential nodes, in term of degree centrality, are the words *text* and *mining* with the degree measure of 5 respectively 4. In theory, by using these words as keywords one should have enough information to concisely summarise the

example sentence.

Other centrality measures commonly used are the *eigenvector centrality* as well the *Katz centrality* measure, which can be seen as extensions and improvements to the degree centrality measure. One of the characteristics of these measures is that they also consider the relative degree of influence of neighboring nodes in the process of assigning a ranking score. This means that not all neighbors are equivalent and that high-scoring nodes contribute more to the score than low-scoring neighbors. In a sense, instead of assigning a scoring point for each neighboring node as the degree centrality does, the metrics gives a score proportional to the sum of the neighbor's respective centrality measure [7].

Although there may exist several other centrality measures, this project has chosen to limited itself to a short description of the metrics mentioned above, as these form the central part of the baseline ranking algorithm used in the purpose of evaluating the proposed ranking algorithm. It's also worth pointing out that there may exist other aspects when ranking a candidate keyword besides centrality measures. Such aspects could be term but also document-term frequencies.

2.4.2 Related Ranking Algorithms

Analysis of social networks and link-structures of the web using graph-based algorithms have shown a great success. One of the more well-known algorithms is the PageRank algorithm introduced in 1998 as a method to 'bring order to the web' in the early days of Google [2]. Although the algorithm was first introduced as a means of measuring the importance of websites, similar line of thinking has been applied to lexical or semantic graphs by algorithms such as DivRank [4] and TextRank [6]. All of these three algorithms characterize themselves by an iterative behavior, where the DivRank algorithm is based on a reinforced random walk through the network while the PageRank and the TextRank iteratively calculates a probability distribution of websites from site citations to achieve convergence. It is worth pointing that the TextRank is based on the PageRank algorithm despite that it identifies itself as a distinct algorithm. Although the PageRank is traditionally applied to directed graphs, it can also be utilized on undirected graphs by substituting an undirected edge with a bidirectional edge.

In an attempt to avoid the iterative behavior of the algorithms mentioned above, and thereby increasing the efficiency, Rose et al. introduced the Rapid Automatic Keyword Extraction (RAKE) algorithm in 2010. The hallmark of the RAKE algorithm is its computational efficiency while at the same

time being able to achieve high precision and recall. As the algorithm is based on observations that keywords frequently contain multiple words but rarely contain standard punctuation or stop words, the authors introduced a ranking score based on $\deg(w)/\text{freq}(w)$. This metric score favors words that predominantly occur in longer candidate keywords and thereby successfully reflecting the authors' observations [8].

The ranking algorithms introduced in this projects derives its desire of an efficient algorithm from [8] but with the clear distinction of striving towards a modular structure, where future key-ranking features can easily be added.

3 Portfolio2Keyword Algorithm

3.1 Basis

The starting point and the goal of the algorithm have been to develop a simple-to-use and interactive algorithm that aids the user in extracting keywords from a collection of curated documents. The goal builds around the idea that users with access to relevant keywords will be able to construct more accurate queries for a patent search and thereby obtaining higher recall and precision. The algorithm is by no means a complete toolset and it is mainly aimed to solve one of the many difficulties of patent search.

The interactive aspect of the algorithm has been implemented in such a way that it allows the user to find neighboring words with a high ranking and relevant keyword. Also, the feature of removing undesired keywords (nodes) has been implemented, which inherently changes the relationship between keywords in the graph, and can preferably be used as a means of curating domain-specific stopwords. The former feature has been designed with the intention to provide the user a simple, yet powerful, tool to construct keyword phrases or multi-word keywords out of single keywords.

3.2 Outline

The algorithm consist of three seperate components as previously descriped in section 2 with an additional optional postprocessing step. These are:

1. Candidate generation
2. Graph generation and term/candidate co-occurance count
3. Keyword ranking
4. User interaction (Optional postprocessing)

In step (1), each document is normalized and tokenized into words, whereas each word is transformed into its base form. The tokens are also annotated with POS-tags, which are used in cases when the user is only interested in keywords of a certain kind. Followed by this, any standard or custom set of stopwords are removed from the text and a set of candidate keywords is thereafter generated.

Step (2) of the algorithm involves the extraction of bigrams (sliding window) from the documents and a count of the candidate occurrence and candidate co-occurrence is carried out. The final act of step (2) is the construction

of a weighted undirected graph, with the candidates forming the vertices and the candidate co-occurrences the edges similarly to figure 1.

In the final step (3), each candidate is being assigned a ranking score by the ranking algorithm and a sorted list, with the highest ranking keyword appearing first, is returned to the user.

In the optional postprocessing step, the user is being presented the possibility of removing undesired keywords and creating multi-word keywords from neighboring nodes. In the case of keyword removal, the ranking algorithm is rerun and the user is presented with an updated list of relevant keywords.

3.3 HarmonicRank

The ranking algorithm introduced in this project builds around the hypothesis that the relative importance (ranking) of keywords can be determined from a set of key-attributes associated with that particular word. In the context of graph-based keyword extraction, these key-attributes could range from any sort of centrality measure to simple frequency measures, including term-document frequency. The latter attribute can intuitively be considered as one of the more significant ones as we aim to find keywords that represent not only single documents but the corpus as a whole.

Finding the most significant attributes to include in the ranking algorithm is no different from any other machine learning problem and can very well be a topic for further research. The attributes used in the proposed algorithm is by no means claimed to be the most optimal ones and they are simply introduced from an intuitive point of view. Let us formally introduce the key-attributes used in the algorithm as follows:

$$x_{1,v} = \frac{\deg(v)}{\Delta(G)} = \frac{\text{Degree of vertex v}}{\text{Maximum Degree of Graph G}}$$

$$x_{2,v} = \frac{\sum_{i=1}^{k} w_i(e_{v \subseteq v})}{\max \sum_{i=1}^{k} w_i(e_{v \subseteq V})} = \frac{\text{Sum of edge weights for vertex v}}{\text{Maximum of Sum of edge weights for Vertex v in Graph G}}$$

$$x_{3,v} = \frac{n_t}{N} = \frac{\text{\# of Documents containing term t/vertex v}}{\text{\# of Documents in the Corpus/Portfolio}}$$

One advantageous property of these attributes is that they can be extracted with a single pass over the undirected graph or during the preprocessing of

documents in step 1 and 2 of the portfolio2keyword algorithm. This means that the ranking algorithm can avoid an iterative behavior, similarly to the one introduced by the PageRank algorithm. It can easily be seen that the attributes will be bounded by $[0, 1]$ and a natural way of combining these values into a final ranking score is through equation 1, which simply is the harmonic mean.

$$R_v = \left(\frac{\sum_{i=1}^{3} x_i^{-1}}{3} \right)^{-1} \tag{1}$$

4 Method and Material

4.1 Data

The data used for the experiments and the evaluation in this project is a collection of 906 patent documents within the technological domain of Wind Power, curated and provided by IAMIP Sverige AB. Each document provides information such as title, abstract, patent claims, IPC- and CPC-class[2], citations as well as figures. In addition to this information, a subjective ranking (discrete values from 1 to 5) of documents is also included as meta-data in the collection to indicate document relevance associated with a specific task.

They majority of the patents documents belongs to, and can shortly be described by, the following CPC-classes and subclasses:

- **Y02:** Technologies or applications for mitigation or adaptation against climate change
  - **Y02E:** Reduction of greenhouse gases [ghg] emission, related to energy generation, transmission or distribution
    - **Y02E10/00:** Energy generation through renewable energy sources
    - **Y02E10/76:** Power conversion electric or electronic aspects
    - **Y02E10/723:** Control of turbines
    - **Y02E10/56:** Power conversion electric or electronic aspects
- **H02:** Generation; Conversion or distribution of electrical power
  - **H02J:** Circuit arrangements or systems for supplying or distributing electric power ; systems for storing electric energy
    - **H02J3/00:** Circuit arrangements for ac mains or ac distribution networks
    - **H02J3/386:** Wind energy
  - **H02P:** Control or regulation of electric motors, electric generators or dynamo-electric converters ; controlling transformers, reactors or choke coils

[2] The Cooperative Patent Classification (CPC) is a patent classification system, which has been jointly developed by the European Patent Office (EPO) and the United States Patent and Trademark Office (USPTO). For further details of the CPC-class system, see Espacenet

- **H02P2101/00:** Special adaptation of control arrangements for generators
- **H02P2101/15:** for wind-driven turbine

Despite the multitude of attributes, only the patent abstracts will be used for the experiments.

4.2 Experiment and Evaluation

Evaluating the performance of a keyword extraction algorithm is certainly not an easy task, especially in the absence of objectively "correct" set of keywords. The lack of a gold standard makes it inherently ineffective to use common metrics such as precision or recall. A time-consuming and expensive solution to this problem is to use human evaluation as mentioned in [3], which is being utilized for the experiments in this project.

The baseline ranking algorithm used for evaluating the proposed ranking algorithm is the PageRank algorithm as previously described in section 2.4.2. Both the baseline and the proposed algorithm will be executed in two setups; (1) Using candidate keywords of all word-types (all POS-tags) and (2) using candidate keywords tagged with a subset of POS-tag, namely nouns, verbs and adjectives. We will hereafter denote these setups as setup 1 and setup 2. Both ranking algorithms will share the same graph and set of candidates generated in step (1) and (2) of the algorithm, as described in 3.2, when executed.

The performance of the algorithm will be measured on the two following aspects:

1. Efficiency - Running time of ranking algorithm
2. Semantic Coherence - Relevance of extracted keywords

The first performance aspect will, for obvious reasons, be an objective measure while the second measuring aspect is judged subjectively by patent search experts at IAMIP. The extracted keywords from each algorithm are presented to the experts anonymously to mitigate the possibility of subjective bias. The experiments are executed on a machine with an Intel® Core i5-2435M processor (@ 2 x 2,4 GHz) and a system memory of 8 GB (DDR3 @ 1333 MHz).

5 Results and Discussion

5.1 Summary and Execution Time

Table 1 presents the summary of the result for both setups for step (1) and (2) of the portfolio2keyword algorithm. From the table, it can be seen that setup 2 (a subset of POS-tags) contains approximately 25% fewer candidates compared to setup 1 (all POS-tags). The effect of this is reflected in the execution times as setup 2 runs around 10 seconds faster compared to the first setup. Although the number of candidates has been reduced by only allowing a certain subset of POS-tags, it can be seen that running time does not differ significantly for the two setups. This indicates that the majority of the execution time is a product of the actual text processing.

It is worth pointing out that the steps, step (1) and (2), of the algorithm have not been implemented for parallel processing nor have they been subjected to code-optimisation. This means that these running times can very well be reduced once such improvements have been done.

	Execution Time (sec)	# of Generated Candidates
All POS-tags	128.540	4468
Subset of POS-tags	116.587	3392

Table 1: Summary of Step 1 & 2 of Algorithm by setup

The execution times of the ranking step for each ranking algorithm and setup is presented in table 2. From the table, we can conclude that the HarmonicRank algorithm runs approximately ten times more efficient than the baseline algorithm as running times differ by a factor of 10 in both setups. One possible source of the differences can be the iterative behavior of the PageRank vs. the single pass behavior of the HarmonicRank. Although the proposed ranking algorithm shows promising results for both setups with two different graph sizes, it is too early to generalize this to a statement of overall performance.

	PageRank	HarmonicRank
All POS-tags	2.4502	0.2429
Subset of POS-tags	1.8622	0.1864

Table 2: Execution time of ranking step in seconds (step 3 of algorithm) by setup and ranking algorithm

5.2 Extracted Keywords

All POS-tags		Subset of POS-tags	
PageRank	HarmonicRank	PageRank	HarmonicRank
power	power	power	power
wind	wind	wind	wind
current	converter	current	current
converter	current	converter	converter
voltage	voltage	voltage	voltage
circuit	control	circuit	control
control	circuit	control	circuit
side	system	side	system
system	output	system	side
unit	side	unit	output
output	generator	device	generator
device	device	generator	device
generator	unit	output	unit
module	energy	energy	energy
dc	connect	module	module
energy	module	generation	dc
generation	dc	dc	supply
supply	supply	air	generation
connect	generation	supply	frequency
end	frequency	end	grid

Table 3: Top 20 Extracted Keywords, by setup and ranking algorithm

Table 3 displays the top 20 extracted keywords using the two ranking algorithms for each setup. It is quite clear that both algorithms manage to extract similar words in both setups with minor differences. It can also be noticed that the keywords from each setup are very similar, which indicate that the set of the selected POS-tags was seemingly suitable.

According to the patent experts at IAMIP, both algorithms manage to extract the essence of the portfolio with the presented keywords in both setups. They claimed that the list of keywords was a good representation of the portfolio and that the possibility of building multi-word keyword from neighboring words would be a valuable tool for both search experts as well as for novice users in the industry. The PageRank algorithm was chosen as the superior ranking method of the two algorithms as the PageRank shows the tendency of assigning a higher ranking to relevant words such as *generation*

and *dc*. However, it is worth mentioning that this decision was only reached if a final winner were required as both algorithms show similar results.

6 Conclusion

In this project in text mining, the reader has been presented an algorithm for automatic keyword extraction from a collection of patent documents within the domain of wind technology, namely Portfolio2Keyword. The goal of the algorithm was to provide users with a simple-to-use and interactive tool with the intention to aid the user in the process patent information retrieval. The algorithm is by no means a complete toolset but can nonetheless introduce time-savings and valuable insights to the user as manual keyword extraction usually reach far longer running times than those presented in section 5. Although the experiments were conducted within the domain of intellectual property, it is worth pointing out that the algorithm can be applied to a wide range of other domains and applications.

A second aim of the project was to introduce an efficient ranking algorithm that, despite its simplicity, extracts semantic coherent keywords to be used as building blocks for future queries. The results in section 5 showed promising results in terms of efficiency and relevance when compared to the widely used PageRank algorithm by patent search experts at IAMIP. Although the PageRank was chosen as the superior ranking algorithm, the harmonicRank showed similar semantic coherent keywords while at the same time exhibiting faster running times in the range of factor 10 compared to the baseline algorithm. Despite that both algorithms manage to perform well on the semantic coherence aspect in both setups, it is worth pointing out that corpus only consisted of short abstracts and that only a few of the included documents had been subjectively ranked as a 5-star document. Arguably, for further increase in relevance, the user can preferably utilize full-text documents or reduce the noise by only including highly relevant documents.

The features used in the modular proposed ranking algorithm were selected from an intuitive point of view and they are not claimed to be the optimal ones. Selecting significant features is no different from any other machine learning problem and this topic can very well be suited for further research. It is also suggested that further experiments are conducted on different technological domains and portfolios before general conclusions of the performance are made.

References

[1] WIPO (World Intellectual Property Organisation). http://www.wipo.int/ipstats/en/charts/ipfactsandfigures2016.html, 12 2017.

[2] S Brin and L Page. The anatomy of a large scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1/7):107–17, 1998.

[3] Kazi Saidul Hasan and Vincent Ng. Automatic Keyphrase Extraction: A Survey of the State of the Art. *Association for Computational Linguistics Conference (ACL)*, pages 1262–1273, 2011.

[4] Qiaozhu Mei, Jian Guo, and Dragomir Radev. DivRank: the interplay of prestige and diversity in information networks. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1009–1018, 2010.

[5] John Tait Anthony J. Trippe Mihai Lupu, Katja Mayer. *Current Challenges in Patent Information Retrieval*. Springer Heidelberg Dordrecht London New York, 2011.

[6] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. *Proceedings of EMNLP*, 85:404–411, 2004.

[7] M. E. J. Newman. *Networks, An Introduction*. Oxford University Press Inc., New York, 2010.

[8] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory*, (March):1–20, 2010.