Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Term Ranker: A Graph-Based Re-Ranking Approach

Muhammad Tahir Khan[a], Yukun Ma[a], and Jung-jae Kim[b]
[a]{Rolls-Royce@NTU Corporate Lab, School of Computer Engineering,Nanyang Technological University, Singapore}
[b]Data Analytics Department, Institute for Infocomm Research, Singapore
tahir@ntu.edu.sg, mayu0010@e.ntu.edu.sg, jjkim@i2r.a-star.edu.sg

Abstract

Term extraction is to extract domain relevant terms from a domain specific, unstructured corpus, which in an organisational setting can be used for categorisation and information retrieval. Previous statistical approaches to automatic term extraction rely on term frequencies, which may not only hamper the accuracy but also lower the rank of or even discard domain relevant yet infrequent terms. This paper aims at minimising the impact of term frequency and thus improving precision of top-*k* terms, by using a graph based ranking algorithm with the aids of latent vector representation of terms and term relations embedded in patents instead of general-domain knowledge sources. We show that the proposed method outperforms all the previous works significantly.

1 Introduction

In an organisational setting, unstructured data is considered crucial and useful in improving operations or in creating new business opportunities for an organisation (Blumberg and Atre 2003). Utilising this information requires extraction of relevant terms, which can be used in categorisation and enhancing search over unstructured data to retrieve relevant information. However, manually extracting relevant terms from domain specific corpora is a labour intensive and time consuming activity, and thus there is a need for maximum automation of this process.

Term extraction is to automatically extract domain relevant terms from unstructured data. Current approaches to term extraction can be classified into two categories: (i) statistical and (ii) graph based approaches. Statistical approaches select linguistically admissible terms and utilise frequency based methods such as C-value (Frantzi, Ananiadou, and Mima 2000) and TF-IDF (Salton and McGill 1986) to rank the terms according to their relevance. On the downside, these techniques rely on large text corpus to provide reliable statistical information for extracting and ranking terms accordingly, which may result in low ranks for infrequent domain relevant terms. On the other hand, graph-based approaches (Brin and Page 1998; Mihalcea and Tarau 2004) minimise the impact of term frequency during the process of term extraction. These approaches are also proved useful when used in combination with statistical approaches for improving the rank of extracted terms (Lossio-Ventura et al. 2014). Basic idea behind the graph based approaches is to build a graph from input documents by extracting key phrases, adding edges whose weights are estimated based on co-occurrences between the key phrases, and ranking nodes by using a centrality algorithm. As domain-relevant yet infrequent terms are not extracted by statistical methods, however, domain-relevant yet infrequent co-occurrence relations also have negative effect on term extraction.

Recent progress in distributional semantic modelling (DSM) (Turian, Ratinov, and Bengio 2010; Pennington, Socher, and Manning 2014; Mikolov et al. 2013; Levy, Goldberg, and Dagan 2015; Cambria et al. 2015) provides us with powerful tools to address the data sparseness issue of co-occurrence metric. DSM learns the vector representation of words, called word embeddings, based on the distribution of their contexts, so that the resultant vector elements do not directly reflect individual word frequencies nor co-occurrences. We can use the vectors to measure similarity between words, which may replace the co-occurrences as graph edges. As one of most successful DSMs, word2vec tool (Mikolov et al. 2013), has achieved state-of-the-art performance for many NLP applications, we adopt this tool for our work on term extraction.

Furthermore, there are a number of efforts (Rospocher et al. 2012; Vivaldi and Rodríguez 2010; Gazendam, Wartena, and Brussee 2010; Cambria and Hussain 2015) that utilise external knowledge sources such as ontology and thesaurus, which include semantic relations (e.g. synonymy, is-a) between terms, in order to improve the statistical and graph based approaches of term extraction. These efforts are based on the assumption that such resources are already available in given domains, but there is often no such comprehensive knowledge source in a specialised technical domain, and manually building one is time consuming and labour intensive.

Alternatively, we propose to utilise term relations embedded in patent documents, which are available for most technical domains and contain structural information that can be utilised to extract domain specific terms (Judea, Schütze, and Brügmann 2014), as follows: As shown in Figure 1, patent documents often have numbers (or subscripts) that are preceded by technical terms e.g., ***compressor 1***, where the terms preceding the same number may have the same or

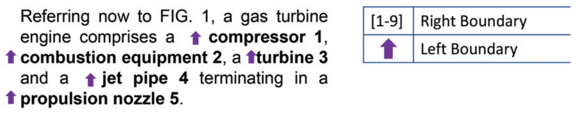

Referring now to FIG. 1, a gas turbine engine comprises a compressor 1, combustion equipment 2, a turbine 3 and a jet pipe 4 terminating in a propulsion nozzle 5.
[1-9] Right Boundary
Left Boundary

Figure 1: Structural information in patent for term extraction

similar meaning. We extract those terms with subscripts and group those with the same subscript, called term maps. The terms in the same term map are treated like the terms under the same category in thesaurus and ontology. Note that the domains of our interests in this paper do not have any comprehensive thesaurus or ontology.

All together, we present a novel graph-based method of term extraction, called "Term Ranker", which utilizes both word embeddings and term relations from patent structure. We aim at improving precision of top k terms by minimising the impact of term frequency during the term extraction process. The method builds a well connected graph whose edges' weights are measured by using the term similarity based on word embeddings and merges similar terms in the graph, where the terms in a term map from patents are considered similar. It then utilises a graph-based ranking model (Mihalcea and Tarau 2004) to identify 'central' and its 'neighbouring' terms in the graph as domain relevant.

We evaluate the performance of Term Ranker against two sets of domain text corpora: Aerospace domain and information technology. The results reveal that Term Ranker as compared to baselines achieves a better performance in terms of precision@K. In the following section, we describe state of the art approaches related to term extraction. In Section 3, we present the proposed term extraction method, and its experiment results in Section 4.

2 Related Work

Current approaches to term extraction mostly rely on frequency based metrics and on the availability of external knowledge sources. In (Ittoo and Bouma 2013), authors proposed a two stage method for term extraction. First, it uses a domain-independent corpus (i.e. Wikipedia) in contrast to a corpus specific to a given domain, in order to assign high scores to terms that are likely relevant to the given domain. Second, it ranks multi-word terms based on their collocational strength to filter out non-relevant terms. This work is different from our approach as it relies on frequency based metric for extracting terms.

The authors of (Vivaldi and Rodríguez 2010) extracted Wikipedia categories relevant for a domain of interest and their associated category pages as candidate terms. However, such an approach is not suitable for highly specialised domain e.g., aerospace domain, since Wikipedia does not cover all terms of highly specialised domains. Other semantic resources (e.g., WordNet) are also used to improve term extraction. The work of (Zhang, Yoshida, and Tang 2009) utilized an ontology to improve the precision of term extraction as follows: Given a multi-word term, the method locates the individual units (or words) of the term in the ontology and assigns a higher rank to the term if the individual units are closer to each other in the ontology hierarchy. Also, the work of (Rospocher et al. 2012) utilises the WordNet to detect synonym terms, which can help in resolving the impact of frequency on the extraction process by ranking higher a synonym term even though it is ranked lower by the statistical method. Differently from our approach, these approaches rely on generic semantic resources that do not provide good coverage for specialised domains. Additionally, they are based on the assumption that an ontology is already available for a domain, which is not true for most of specific domains.

3 Method

Our method "Text Ranker" has the following four main components, as illustrated in Figure 2:

- Extraction of single-word and multi-word terms by using a statistical approach
- Identification of groups of semantically similar terms (hereafter, referred as *TermMap*) from patents
- Estimating term similarity based on term embeddings
- Graph refinement and node centrality ranking

This method can be applied for any domains that have patents or semantic resources like thesaurus available. The following sections describe more details of each component.

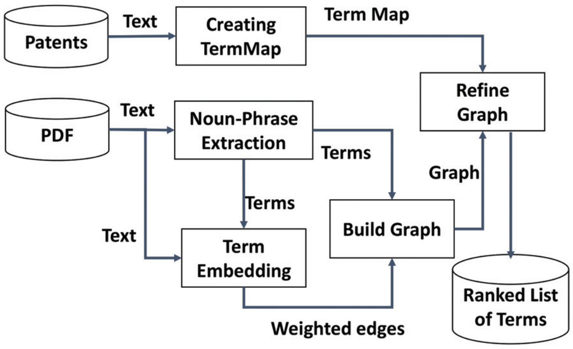


Figure 2: A graph-based re-ranking approach

3.1 Generating term candidates

The first step of term extraction process is corpus preprocessing, which includes removal of non content bearing text, tokenisation, and lemmatisation (see Section on domain-specific corpus collection for our evaluation). We have implemented pre-processing scripts by using regular expressions to remove undesirable content e.g., symbols, tables, equations, e-mail addresses. The text is lemmatised using Stanford NLP software[1] to reduce inflectional forms of a word to a common base form, which helps us measure the distribution of candidate terms in base forms instead of inflected forms (Yarowsky 1994).

[1] http://nlp.stanford.edu/software/corenlp.shtml

The second step is to extract terms (single or multi-word) from pre-processed corpus that characterise and represent the target domain. We have applied a linguistic filter to find candidate terms as follows: extracting noun phrases by using Stanford Parser[2] (Klein and Manning 2003) and filtering out those that are unlikely to be technical terms by using a stop-words list (i.e. generic words in a language).

The final terminology list produced is validated for their domain specificity by using the measure of termhood, defined as "the degree that a linguistic unit is related to domain-specific concepts" (Kageura and Umino 1996). We use C-value (Frantzi, Ananiadou, and Mima 2000) and TF-IDF (Salton and McGill 1986), which are termhood measures to filter and associate weight to terms according to their relevance for the domain. C-value is a domain independent method that favours the extraction of multi-word terms. It uses frequency *f(a)* of a term a if it is not included in other terms (line 1), where $|a|$ indicates the number of words in the term a. When a term a is part of longer terms, the frequency of a is reduced by the average frequency of the terms in T_a as shown in (line 2), where T_a represents the set of terms that contain the term a.

$$C - value(a) = \begin{cases} log_2\left(|a| \times f(a)\right) \;\; if\, a\, is\, not\, nested & (1) \\ \\ log_2\left(|a| \times f(a) - \frac{\sum_{b \epsilon T_a} f(b)}{|T_a|}\right) \;\; otherwise & (2) \end{cases}$$

On the other hand, TF-IDF (term frequency-inverse document frequency) method (Salton and McGill 1986) computes termhood for both single-word and multi-word terms. The idea behind TF-IDF is that a term is important for a document if it has high frequency in this document and also occurs only in few other documents. Mathematically it is defined as follows:

$$tf\text{-}idf_{t,d} = \underbrace{(1 + log(tf_{t,d}))}_{tf\ part} \underbrace{* \, log\left(1 + \frac{|D^{(c)}|}{|D_t^{(c)}|}\right)}_{idf\ part}$$

In order to establish relevance of a term (t) to a domain corpus (c), we use the sum of TF-IDF values of the term for all the documents in the corpus as proposed by (Manning and Schütze 1999), as follows: $tf\text{-}idf_t^{(c)} = \sum_{\forall d \epsilon D_t^{(c)}} tf\text{-}idf_{t,d}$

After calculating both termhood values, we normalise them by their maximum values and linearly combine to compute a final domain score as follows, where α and β are weights to emphasize $C\text{-}value(t)$ or $TF\text{-}IDF(t)$:

$$Score(t,c) = \alpha . \frac{C\text{-}value(t)}{max_{t \epsilon c}(C\text{-}value(t))} + \beta . \frac{TF\text{-}IDF(t)}{max_{t \epsilon c}(TF\text{-}IDF(t))}$$

Given a domain corpus c, we compute $Score(t, c)$ for all extracted terms t and choose k top-scored terms as domain-relevant candidates. The values of k for our evaluation are specified in Section 4.

Table 1: Term maps from patents in aerospace domain

Nozzle Guide Vane, Guide Vane, Outlet Guide Vane
Stator Vane, Variable Stator Vane
Compressor Shaft, Compressor Blade

3.2 Term Map construction

A patent is a technical document disclosing an invention, which includes a written description and figures. The text in a patent contains references to important concepts involved in the invention, as exemplified in Figure 1. Extracting these terms requires identification of the left and right boundaries of a term. While the number (or subscript) marks the right boundary of a term, we use linguistic criteria (*JJ—NN*)*(*NN*) to extract a valid term i.e., detecting the left boundary of a term.

Patents not only provide a list of technical terms but can also be used to build a resource of semantically similar terms. In a patent, terms with the same subscript are often synonyms or variations of a technical term. Consider Example 1, in which the subscripts ***42*** and ***36*** associate {***outer fan duct wall adapter**, **outer wall duct wall adapter***} and {***box structure**, **bifurcation structure***}, respectively. We group such terms to build a *term map*.

Example 1 *"The forward edge of the **outer fan duct wall adapter 42** is sized.....The rearward edge of the **outer wall duct wall adapter 42** is sized to mate .."*

*" The **box structure 36** has spaced, substantially parallel....so that the edges of the gaps abut the side walls of the **bifurcation structure 36**"*

Additionally, we merge overlapping term maps to cater for scenarios, where a term appears in multiple documents and is grouped with different synonymous terms. When merging term maps, however, we discard shared terms that are uni-grams, which often mislead the merging. More example term maps extracted from patents in aerospace domain are shown in Table 1.

3.3 Term embeddings

As discussed in the Introduction, we use word embeddings (i.e. vector representation of word) to compute term similarity, which will be used as weights of edges between the terms in a graph (see Section 3.4). Figure 3(a) depicts relations between terms and their edge weights, even though these terms did not appear together in text corpus.

Specifically, our word embeddings are learnt using skip-gram model (Mikolov et al. 2013), which is a neural network model trained to predict local context within a window, given a centre word. The objective function is to maximize the following log probability, where k is the size of local context window, and T total number of word types.

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-k \le j \le k, j \ne 0} \log \frac{\exp(v_c'^{\top} v_w)}{\sum_{c'} \exp(v_{c'}'^{\top} v_w)}$$

To obtain such vector representation of multi-word terms as well as single-word terms (called "term embeddings"),

[2] http://nlp.stanford.edu/software/lex-parser.shtml

we utilise the candidate selection process described in Section 3.1 to detect multi-word terms and then convert them into single tokens before training the neural network of skip-gram model. Once term embeddings have been learned, we select top 10 similar terms for each candidate based on cosine similarity between term vectors and add edges from the candidate to the 10 terms in a graph. Namely, the weight of the edge connecting two term candidates (t_1, t_2) is defined as follows:

$$E(t_1, t_2) = \frac{v_{t_1} \cdot v_{t_2}}{|v_{t_1}| \cdot |v_{t_2}|}$$

, where v_{t_1} and v_{t_2} are the term embeddings for term candidates t_1 and t_2, respectively, and $E(t_1, t_2)$ is the weight.

3.4 Graph refinement

The objective of graph refinement step is to improve the precision@k of term extraction process by re-ranking domain relevant terms to the top of term list. In order to achieve this, we adapt TextRank, a graph-based ranking model for text processing (Mihalcea and Tarau 2004), by integrating in-domain knowledge from patents (see Section 3.2) and term similarity from term embeddings as edge weights.

Basic idea behind previous graph based approaches is to build a graph from input documents by extracting all key phrases, adding edges among them based on co-occurrence relation, and ranking nodes by using a centrality algorithm (Brin and Page 1998; Mihalcea and Tarau 2004). The key differences of our approach from the previous ones are that we use a statistical method to filter top-*k* terms as the graph nodes, use term embeddings to give weights to the edges regardless of the scores from the statistical method, and use term maps from patents to merge similar terms. Figure 3 depicts an example graph with the weights from term embeddings and a revised one with similar terms merged.

Once the term graph is constructed, the next step is to assign a score to each node. For this purpose, we adopt TextRank (Mihalcea and Tarau 2004) graph-based ranking algorithm. It uses eigenvector centrality measure and implements the concept of voting. Eigenvector centrality measures the centrality of a node based on the importance of its surrounding nodes. In detail, when a node links to another node, it is casting a vote for its neighbour, and the importance of a node is correlated to the number of such votes. In addition, a node is considered important if it has a connection to another node with high score. In summary, the score of node is computed based on the votes it has, the importance of node casting vote for it, and the edge weights.

Example 2 ***[Graph generation]:*** *The graph in Figure 3(a) has weighted edges between nodes that are derived from term embeddings. In the current state, it has three nodes with many votes: "Stator Vane, Core Engine, and Combustion System", ordered according to number of edges. There are other relevant nodes that are ranked low because of fewer connections and low scoring neighbours e.g., "High Pressure Compressor, Creep Resistance, Inlet Air, and Nozzel Guide Vane.*

[Node merging]: *We utilise the term maps built from patent documents e.g., {Compressor Shaft, Compressor Blade}, {Stator Vane, Variable Stator Vane}, {Nozzle Guide Vane, Guide Vane, Outlet Guide Vane}. Based on the term map information, we merge the nodes in the graph, as illustrated in Figure 3(b), which creates more 'central' nodes. As a result, the nodes that were ranked low are now connected to more high scoring nodes: For example, "Inlet Air" and "High Pressure Compressor" are connected to {Stator Vane, Variable Stator Vane}, and {Nozzle Guide Vane, Guide Vane, Outlet Guide Vane}, respectively. Following the TextRank algorithm, these nodes will be ranked higher because of their connection to high scoring nodes.*

4 Evaluation

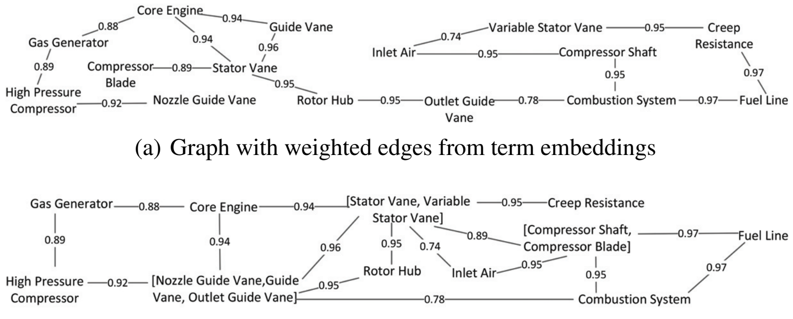
We compare our method with previous solutions for term extraction including statistical and graph based approaches. The methods are evaluated in terms of precision@k terms for the aerospace and information technology domains. We have used a combination of C-value and TF-IDF termhood measures as a baseline system (described in Section 3.1) to extract terms. Our evaluation results show that the proposed approach improves the precision over the baseline solutions. We briefly describe the text corpora utilised in our experiments, along with the sources used to build domain specific knowledge source, in Section 4.1. In Section 4.2, we compare the performance of our approach against the previous systems.

4.1 Data Sets

Domain-specific corpora This work has been carried out in very specialised domains of aerospace and information technology (IT). We have crawled from the Web to build in-domain text corpora as follows: A set of domain relevant keywords was created to crawl the web. Web pages matching the search criteria of the keywords are downloaded. The set of keywords for the aerospace domain is composed of 115 in-domain concepts (e.g., engine nozzle, Aerofoils, Foreign Object damage). That for the IT domain is composed of 30 keywords (e.g. Computer aided design, Knowledge management, Model driven development). In order to reduce the search space and restrict the crawler to our topic of interest (i.e., jet engine), these keywords were combined with filtering criteria e.g., Jet Engine. Hence, a query passed to the crawler is the combination of a keyword and the filtering criteria e.g., "Aerofoils + Jet Engine ". The crawler downloaded 1,300 and 1,500 Web pages for the two domains, respectively.

Patent Set We collected 208 patents related to the jet engines published by the organisation "X" for building a knowledge base of semantically equivalent terms. Since the patents are published by the organisation "X", this set of patents will provide good coverage for the documents extracted with in-domain keywords.

For the second domain, we collected 742 patents using the same keywords used for crawling text documents. These patent collected, however, are very diverse in nature due to the broadness of the domain.



(a) Graph with weighted edges from term embeddings

(b) Term map based graph refinement

Figure 3: TermRanker graph refinement

Table 2: Ranked list of terms from the baseline

Terms	Score	Frequency
Jet engine	0.959	2941
combustion chamber	0.493	1468
high temperature	0.163	452
high speed	0.132	359
overall pressure ratio	0.117	221
fan casing	0.024	66
electronic control unit	0.02	36
hush kit	0.018	47

4.2 Term Ranking & Performance Evaluation

We extracted terms by using the baseline system defined in Section 3.1. An example term list along with term frequencies and scores from the baseline is illustrated in Table 4.2. These examples show that the baseline termhood measures (C-value and TF-IDF) favour terms with high frequencies, which results in extraction of frequent but non relevant terms (e.g., *high speed*). This reliance on frequency not only hampers the precision of the baseline system but also lowers the rank or even discards domain relevant terms due to their infrequent presence in texts e.g., *fan casing, electronic control unit*, with frequencies *66, and 36*, respectively.

Since evaluating the entire ranked lists is tedious, we narrowed down the terms to top-*k* terms. We manually inspected the top-500 candidates and, based on our knowledge of the domains, marked them as true positive or false positive. To ensure the accuracy of our evaluation, we have verified the marked list with two domain experts. Table 3 presents and compares the results of our proposed method, the previous systems, including the baseline (i.e. statistical method) and TextRank (i.e. previous graph-based method), and our method using the term embeddings, but not using the patent term maps (labelled as "term embedings"). As it can be seen from the results, Term Ranker outperformed all the other systems significantly.

The second column of Table 3 lists results of the TextRank, which performs significantly poor as compared to the baseline (first column). The graph based approach is expected to minimise the impact of frequency on term extraction by ranking terms according to their relations to other terms and to the terms that are considered important in a graph, but it depends on co-occurrence relation between terms, which is found unreliable due to the data sparseness problem, as discussed in the Introduction. Our analysis of the results reveals that the majority of the terms ranked higher by TextRank are uni-grams with significant presence in text.

In order to mitigate this issue, instead of using co-occurrences, we have utilised term embeddings to capture semantic and syntactic similarities among terms. As shown in the third column, it improves the result over TextRank and also mostly over the baseline. Note that the improvements are significant in both domains of aerospace and IT, which may indicate that the usage of term embeddings are domain independent and effective in both domains.

The fourth column presents the results of our approach, which refines the graph built upon the term embeddings by further incorporating the term maps extracted from patent documents, which significantly outperforms the baseline. When comparing our methods with and without the usage of the term maps from patents, we found that the improvement of using the term maps is more obvious in the aerospace domain than in the IT domain, in which the term maps from patents are quite diverse and thus may lose their impact on the re-ranking.

Term Ranker improves the precision@k by reducing the impact of frequency on the term ranking process and by improving the ranks of terms that are previously penalised by their insignificant presence in a corpus. Table 4.2 shows such example terms: For example, the rank of the term *electronic control unit* has been improved significantly over the baseline system, while the ranks of the terms *high temperature* and *high speed* have been lowered.

The term *combustion chamber* with a high initial rank is moved down the list because of its fewer term similarity edges to top-k terms and no connection to a central node in a graph. In this paper, we focused on term similarity, but did not consider other semantic relations (e.g. part-of), where *combustion chamber* is a part of a top ranked concept *jet engine*, which we leave to future work.

Table 3: Precision Comparison of TermRanker with baseline systems

(a) Aerospace

P@k	C-TFIDF	TextRank	Term-Embedding	Term-Ranker
100	0.74	0.47	0.67	0.84
200	0.68	0.39	0.69	0.75
300	0.657	0.41	0.667	0.7
400	0.605	0.41	0.662	0.685
500	0.582	0.448	0.64	0.65

(b) Information System

P@k	C-TFIDF	TextRank	Word-Embedding	Term-Ranker
100	0.71	0.43	0.7	0.73
200	0.57	0.32	0.62	0.64
300	0.53	0.29	0.61	0.63
400	0.50	0.34	0.59	0.59
500	0.48	0.42	0.56	0.57

Table 4: Re-ranked term list with Term Ranker

Baseline Rank	Current Rank	Term
654	6	electronic control unit
1	8	jet engine
458	17	fan casing
790	62	hush kit
2	759	combustion chamber
7	823	high temperature
13	869	high speed

5 Conclusions

A graph based method is proposed to improve the precision of top-*k* terms extracted by a statistical method. It utilises TextRank, a graph-based ranking model, to minimise the impact of frequency on term extraction by ranking terms according to their relations to other terms and to the centroids in a graph. The original TextRank algorithm relies on co-occurrence relation between terms to build graph, which is affected by frequency of non relevant frequent terms and also by distance between two relevant words in text. In contrast, Term Ranker mitigates this issue by learning vector representations of term candidates (i.e. term embeddings) and utilising it to capture similarities along with relation strength between terms for adding edges between nodes, and hence building a well connected graph. We capture semantically similar terms (called term maps) from patents and use this information to put together semantically similar terms in a graph, thus increasing the number of central nodes and improving the rank of infrequent terms.

References

Blumberg, R., and Atre, S. 2003. The problem with unstructured data. *DM REVIEW*.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks*.

Cambria, E., and Hussain, A. 2015. *Sentic computing: a common-sense-based framework for concept-level sentiment analysis*, volume 1. Springer.

Cambria, E.; Fu, J.; Bisio, F.; and Poria, S. 2015. Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*, 508–514.

Frantzi, K.; Ananiadou, S.; and Mima, H. 2000. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries* 3(2):115–130.

Gazendam, L.; Wartena, C.; and Brussee, R. 2010. Thesaurus based term ranking for keyword extraction. In *DEXA*.

Ittoo, A., and Bouma, G. 2013. Term extraction from sparse, ungrammatical domain-specific documents. *Expert Systems with Applications*.

Judea, A.; Schütze, H.; and Brügmann, S. 2014. Unsupervised training set generation for automatic acquisition of technical terminology in patents. In *COLING*.

Kageura, K., and Umino, B. 1996. Methods of automatic term recognition: A review. *Terminology*.

Klein, D., and Manning, C. D. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Levy, O.; Goldberg, Y.; and Dagan, I. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*.

Lossio-Ventura, J. A.; Jonquet, C.; Roche, M.; and Teisseire, M. 2014. Yet another ranking function for automatic multiword term extraction. In *Advances in Natural Language Processing*.

Manning, C. D., and Schütze, H. 1999. *Foundations of statistical natural language processing*. MIT press.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into texts. Association for Computational Linguistics.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS'13*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the EMNLP*.

Rospocher, M.; Tonelli, S.; Serafini, L.; and Pianta, E. 2012. Corpus-based terminological evaluation of ontologies. *Applied Ontology*.

Salton, G., and McGill, M. J. 1986. Introduction to modern information retrieval.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th ACL*.

Vivaldi, J., and Rodríguez, H. 2010. Finding domain terms using wikipedia. In *LREC*.

Yarowsky, D. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *Proceedings of the 32nd ACL*.

Zhang, W.; Yoshida, T.; and Tang, X. 2009. Using ontology to improve precision of terminology extraction from documents. *Expert Systems with Applications*.