Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4388–4394
Marseille, 11–16 May 2020

Varying Vector Representations and Integrating Meaning Shifts into a PageRank Model for Automatic Term Extraction

Anurag Nigam[1], **Anna Hätty**[1,2], **Sabine Schulte im Walde**[1]
[1]Institute for Natural Language Processing, University of Stuttgart, Germany
[2]Robert Bosch GmbH, Corporate Research, Renningen, Germany
{anurag.nigam, schulte}@ims.uni-stuttgart.de, anna.haetty@de.bosch.com

Abstract

We perform a comparative study for automatic term extraction from domain-specific language using a PageRank model with different edge-weighting methods. We vary vector space representations within the PageRank graph algorithm, and we go beyond standard co-occurrence and investigate the influence of measures of association strength and first- vs. second-order co-occurrence. In addition, we incorporate meaning shifts from general to domain-specific language as personalized vectors, in order to distinguish between termhood strengths of ambiguous words across word senses. Our study is performed for two domain-specific English corpora: ACL and do-it-yourself (DIY); and a domain-specific German corpus: cooking. The models are assessed by applying average precision and the roc score as evaluation metrices.

Keywords: Term Extraction, PageRank Algorithm, Vector Representations, Meaning Shifts

1. Introduction

Terms are linguistic expressions which characterize a domain, i.e., words and phrases which are typical of a document or a corpus from a specific domain (in contrast to general language usage). The automatic recognition of terms represents an important basis for further Natural Language Processing (NLP) tasks, such as thesaurus creation, automatic translation, and, in general, for domain knowledge acquisition and comprehension. Approaches for automatic term extraction can broadly be classified into four categories: linguistic (Justeson and Katz, 1995; Basili et al., 1997), statistical (Schäfer et al., 2015), hybrid (Frantzi et al., 1998; Maynard and Ananiadou, 1999) and machine-learning approaches (da Silva Conrado et al., 2013). Recently, word vector and deep learning approaches have emerged (Zadeh and Handschuh, 2014b; Amjadian et al., 2016; Wang et al., 2016).

A niche of approaches is represented by graph-based statistical methods. In particular, we only find a handful of ideas for how to exploit the PageRank algorithm for automatic term extraction. The underlying motivation is that graph-based ranking algorithms –relying on graphs whose vertices represent the vocabulary of a corpus and whose edges represent some relationship between the words in the vocabulary– have the potential to inform about the relative importance of vertices in a graph, and that important vertices are likely to represent terms. For example, Mihalcea and Tarau (2004) used co-occurrence counts as edge weights in a graph; Khan et al. (2016) used embedding similarity for the same purpose. Zhanga et al. (2017) incorporated semantic relatedness via a personalized PageRank algorithm for extracting terms from the corpus.

In this study, we present a comparison of vector space representations within a PageRank graph algorithm for automatic term extraction. We go beyond standard co-occurrence and investigate the influence of measures of association strength (Evert, 2005) and first- vs. second-order co-occurrence (Rapp, 2002; Sahlgren, 2006; Schlechtweg et al., 2019) as basis for connecting vocabulary words in domain-specific and general-language graphs with varying types of edge weights. Our study is performed for the domain-specific ACL corpus and a do-it-yourself (DIY) corpus for English, and a domain-specific German cooking corpus. For the German term extraction we further integrate meaning shift values suggested in a previous study (Hätty et al., 2019) to create personalized representations and to thus distinguish between termhood strengths of ambiguous words across word senses.

2. Related Work

The term extraction in Mihalcea and Tarau (2004) was probably the first approach to apply the PageRank algorithm. In a graph-based ranking model the terms extracted from the natural language text with a co-occurrence value within an *n*-sized window are represented as an undirected graph with nodes as single terms and co-occurrence values as unweighted edges. In their implementation the larger the text, the more terms were found.

Personalized PageRank (PPR) biases the graph network by using seed terms which enhance the process of the random walk. For example, De Groc et al. (2011) use a *bootstrapping procedure* which selects seed terms provided by the user and creates search queries that find the appropriate documents. The PPR algorithm is used to extract the terms from the documents. The process is iterative until it gets the highest ranking terms. Figure 1 shows the complete process of the algorithm. Florescu and Caragea (2017) hypothesized that adding keyword positions improves the term extraction, because intuitively the most important keywords tend to be further ahead in the dataset. Their graph is represented by words (nodes) and co-occurrences (edge weights). The teleport vector in PPR is biased over the first position of the keyword in the document (i.e. value 1/position initialized and normalized over all); frequent occurrences were averaged.

Khan et al. (2016) extend the work of (Mihalcea and Tarau, 2004) using frequency-based methods such as the C-value (Frantzi et al., 2000) and tf-idf (Salton and McGill, 1983) for ranking the terms according to their relevance. Figure 2 shows the baseline model of the Term Ranker. It first finds all the noun phrases with a specific part-of-speech pattern and then creates an initial ranking of term candidates using a combined measure of the C-value and tf-idf (as termhood measures). It generates the term map by finding similar/overlapping terms, e.g. *Nozzle Guide Vane, Guide Vane, Outlet Guide Vane*, to later on merge similar nodes in the graph. The embedding is learnt from skip-grams using SWTs and MWTs, after preprocessing the MWTs as one word, e.g. *Nozzle_Guide_Vane*.

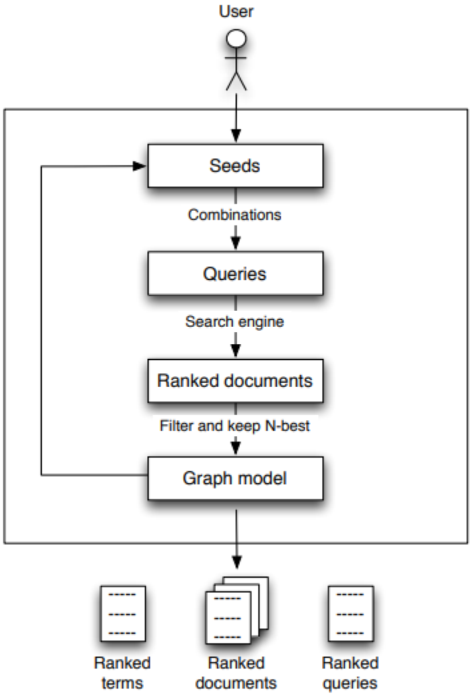

Figure 1: Components of the GRAWTCQ algorithm (De Groc et al., 2011).

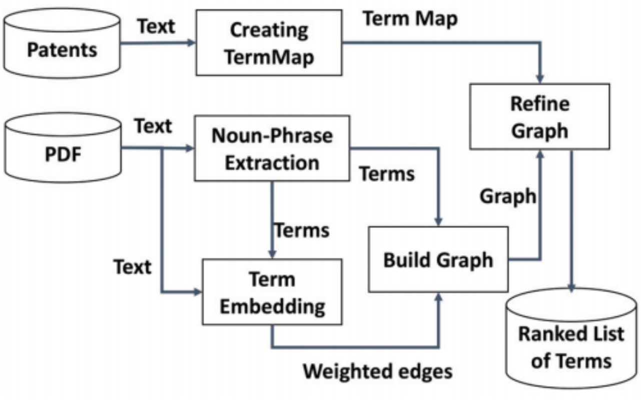

Figure 2: TermRanker graph based on re-ranking approach (Khan et al., 2016).

Existing ATE methods are extended by incorporating *semantic relatedness*. Zhanga et al. (2017) evaluated 13 state-of-the-art term extraction methods and re-ranked the already existing term candidate ranking. They compute the semantic relatedness *semrel(w,z)* for a word w based on pairs of w and other words z:

semrel(w,z) = cosine similarity of two word embdeddings

relrank(w) = returns similarity ranking of all words from term candidate set to target word w, based on semrel

The nodes in the graph represent the words from the term set connected with undirected edges. All the x top-ranked words from *relrank(w)* are connected with the target word w via edges. The approach incorporates the word embeddings into PPR to compute *semantic importance* scores for candidate terms from the semantic-relation graph.

The final term ranking is the medium of Base-TE score and the new PageRank/semantic relatedness score. The semantic importance score consists of two parts: (i) On the *document level* the PageRank scores are calculated and added for all occurrences of the word in the corpus; and (ii) on the *corpus level* semantic relatedness is calculated on word vectors trained on the whole corpus.

3. Motivation

General motivation. The PageRank algorithm has been used for term and keyphrase extraction in the past, but up to date there is no comparison of different edge-weighting methods. So far, Mihalcea and Tarau (2004) used co-occurences while Khan et al. (2016) and Zhanga et al. (2017) used embedding similarity.

Personalization/teleport-vectors have been used to "bias" the PageRank algorithm in the correct direction. Florescu and Caragea (2017) for keyphrase extraction used teleport vectors which contain {*1/first_position_in_document*}. The assumption is that keyphrases occur early in the document. Zhanga et al. (2017) created a set of seed terms and assigned a word to 1 if it belonged to the set of seed terms, and 0 otherwise.

Motivation for shift values. We created a new approach to bias the personalization vector by placing more weight on univocal words: we do this by using shift values as weighting. By predicting meaning shifts from general to domain-specific language in the cooking domain, we get a gradual score with clearly univocal words at one end of the range, and ambiguous or highly versatile words at the other end. We find univocal terms at the first end of the range; the randomly crawled general-language corpus contains some small amount of cooking content as well, and domain-specific and univocal cooking terms receive a low meaning shift value. We wanted to exploit this effect for the PageRank algorithm: by using the shift values within the personalization vector, we could bias the PageRank algorithm accordingly. With our method, it is biased more strongly for univocal words and less strongly against ambiguous or highly versatile words. Since we use a context-based approach for edge weighting, it should be beneficial that the algorithm gets less deviated from the correct paths.

4. Task and Data

Our study uses PageRank values for term extraction from a domain-specific corpus in comparison to a general-language corpus. The main aim of this paper is to perform a comparative study of the domain-specific and general corpus using different edge-weighting methods and using meaning shift predictions as values for the personalization vector in order to bias the network.

4.1. Corpora

DIY is an English domain-specific corpus extracted from do-it-yourself instructions on an English Bosch-empowered DIY homepage.[1]

Figure 3: Terms in DIY corpus as word cloud.

COOK is a German domain-specific corpus. We crawled cooking-related texts across several categories (recipes, ingredients, cookware and cooking techniques) and the cooking recipe websites *kochwiki.de* and *Wikibooks kochbuch*[2].

Figure 4: Terms in COOK corpus as word cloud.

ACL is an English corpus consisting of publications in the domain of computational linguistics. The corpus was created by (Zadeh and Handschuh, 2014a).

We chose the three domain corpora *COOK*, *ACL* and *DIY* to cover two languages and dissimilar domains, with one scientific (ACL) and two non-scientific (DIY & COOK). The technical terms are referred to as domain-specific terms. Furthermore, registers are diverse, the corpora cover formal technical text as well as user-written text.

Figure 5: Terms in ACL corpus as word cloud.

4.2. Data Pre-processing

We applied the following pre-processing steps:

- Removal of stopwords.
- Removal of pairs with character length ≤2.
- Removal of pairs with co-occurrence counts of 1.
- Removal of words with special characters.

4.3. Gold Standards and Evaluation

We used three term extraction gold standard datasets:

DIY: The gold standard is created by looking up the words of the DIY corpus in a set of term list from the web.[34] We allowed overlap with words in term lists, and to some extent tried to find the best size of overlap and minimal word length with reasonable results.

COOK: The gold standard is created by looking up those 1,125 words in PONS, Langenscheidt, Wiktionary and Wikipedia that were obtained from computing meaning shift for all the nouns, verbs and adjectives in the cooking corpus with frequency greater equal to 50. If a word entry contained "Gastronomie", "Kochkunst", "Kochen" or "Kochkunst and Gastronomie" (depending on tags in the respective dictionaries) it was considered a cooking term.

ACL: The gold standard created by (Zadeh and Handschuh, 2014a). It consists of single and multiword expressions taken from ACL publications. We confine to single words for our work. Furthermore, the gold standard distinguishes between two kinds of terms, which we cumulate to a single term class.

The gold standard datasets of the DIY, ACL, and COOK corpora are used to assess the correctness of our model predictions by applying average precision and the roc score as evaluation metrices.

Table 1 shows the binary class distribution in the gold standard datasets which represents 0 (non-term) as class 1 and 1 (term) as class 2.

[1] An example instruction can be found here: https://www.bosch-do-it.com/za/en/diy/knowledge/project-guides/wooden-deck-73489.jsp

[2] de.wikibooks.org/wiki/Kochbuch

[3] https://www.bosch-do-it.com/gb/en/diy/knowledge/encyclopaedia/

[4] https://en.wikipedia.org/wiki/Power_tool

Dataset	0	1
ACL	12,676	1,033
DIY	2,461	336
COOK	600	525

Table 1: Gold standard data distribution.

5. PageRank: Graph Construction and Edge Weighting

5.1. Graph Construction

The PageRank algorithm (Page et al., 1999) is used to compute the importance of the nodes in a graph connected with the edges. The algorithm works on the principle that each page holds a certain weight which depends on the links (forward and backward links) to that page. The PageRank PR of a given page A is computed as follows.[5]

$$PR(A) = (1{-}d){+}d(PR(T_1)/C(T_1){+}\ldots{+}PR(T_n)/C(T_n))$$

where $PR(A)$ refers to PageRank for page A, d denotes the dangling factor which can be set between 0 and 1, $C(A)$ denotes the number of outgoing edges from page A and $T_1 \ldots T_n$ are the pages pointing to page A.

Graph edges are created by word co-occurrences, i.e. an edge is created between two words if they co-occur within a window of words. We use matrix *M* to represent this, where the value of each matrix cell $M_{i,j}$ represents the number of co-occurrences of the word w_i and the context c_j: $\#(w_i,c_j)$. We set the concrete window size to 20, i.e. computing the co-occurrence value within the window of size 20 between the word w_i and context c_j.

5.2. Edge Weighting

For edge weighting, we use the following measures:

Point-wise Mutual Information (PMI). In PMI representations the co-occurrence counts in each matrix cell $M_{i,j}$ are weighted by the mutual information of target w_i and context c_j reflecting their degree of association. The values of the transformed matrix are

$$M_{\mathrm{i,j}}{}^{\mathrm{PMI}} = log\left(\frac{\#(w_{\mathrm{i}}, c_{\mathrm{j}})\sum_c \#(c)}{\#(w_{\mathrm{i}})\#(c_{\mathrm{j}})}\right)$$

Local Mutual Information (LMI). In LMI representations the co-occurrence counts in each matrix cell $M_{i,j}$ are weighted by the local mutual information of target w_i and context c_j reflecting their degree of association (Evert, 2005). The values of the transformed matrix are

$$M_{\mathrm{i,j}}{}^{\mathrm{LMI}} = \#(w_{\mathrm{i}}, c_{\mathrm{j}}) * log\left(\frac{\#(w_{\mathrm{i}}, c_{\mathrm{j}})\sum_c \#(c)}{\#(w_{\mathrm{i}})\#(c_{\mathrm{j}})}\right)$$

Cosine Similarity. In cosine similarity representations, each matrix cell is weighted by the cosine distance of target w_i and context c_j word vectors reflecting their degree of association. The values of the transformed matrix are:

$$M_{\mathrm{i,j}}{}^{\mathrm{Cosine}} = \frac{\vec{w_{\mathrm{i}}}\ \vec{c_{\mathrm{i}}}}{\sqrt{\vec{w_{\mathrm{i}}}^2}\ \sqrt{\vec{c_{\mathrm{i}}}^2}}$$

We used pre-trained fastText[6] vectors for German as basis.

5.3. Shift Values

The shift values were used to bias the network by forming personalization vector. The personalization vector was created as:

$$PV[v] = \begin{cases} float(abs(1 - float(SD[v]))), & \text{if } v\ in\ SD. \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where *PV* is the personalization vector of the type dictionary, and *SD* is the dictionary of a word and its shift value.

5.4. Degree of Association

We further distinguish between two degrees of word association resulting from how matrix M is created.

First-order Association: The matrix M is constructed between the nearby words within a *n*-sized window, as described above.

Second-order Association The matrix M is constructed between context words of nearby words within a *n*-sized window.
The idea for second-order co-occurrence vectors was first introduced by Schütze (1998) for word sense discrimination and has since then been extended and applied to a variety of tasks (Rapp, 2002; Sahlgren, 2006; Schulte im Walde, 2010; Zhuang et al., 2018; Schlechtweg et al., 2019). The basic idea is to represent a word *w* not by a vector of the counts of context words it directly co-occurs with, but instead by a count vector of the context words of the context words. The second-order co-occurrence provides the list of words which are contextually similar but not directly related within the corpus, i.e. context words sharing common list of words within a given threshold, therefore allowing to measure contextual similarity between higher order co-occurrences within the corpus. The second-order co-occurrence vectors are considered less sparse and more robust than first-order vectors (Schütze, 1998).

6. Results and Discussion

6.1. A Comparative Study

We perform a comparative study for different edge-weighting techniques using the PageRank algorithm.

[5] www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm

[6] https://fasttext.cc/docs/en/crawl-vectors.html

6.2. Baselines

Table 2 shows our baseline results for each domain which was computed by shuffling the target words and using the resulting random ranking to compute average precision (AP) and the ROC score. We did three runs and averaged. We can see that the results are much better for COOK in contrast to DIY and ACL for average precision. We relate this to the fact that the COOK gold standard is more balanced for terms and non-terms than DIY and ACL.

6.3. Edge-Weighting Experiments

For these experiments, we computed the PageRank values by using different edge weights in order to find out which weight performs best. These first experiments were conducted without biasing the network by employing a personalization vector. Table 3 shows the resulting scores for all datasets with different edge weights. As a first observation, all results clearly outperform the respective baselines. In general, we find that co-occurrence and cosine behave similarly in their results, as do PMI and LMI. For cooking and DIY co-occurrence and cosine perform better than PMI and LMI, while for ACL it is the other way round. The preferred weighting method does not seem to be language-dependent.

As it was the case for the baselines, we also see a performance drop from COOK to DIY and ACL for the PageRank results. We attribute these differences to imbalance regarding terms vs. non-terms in the datasets. To analyse this assumption for the decrease in performance, we calculated P@k ('precision at k') where k denotes the number of terms. By computing P@k, we want to reduce the impact of length and imbalance of the gold standards. In Table 5 we can see that as the value of k increases for the corpora, the performance of the model decreases, which is expected. But also none of the results for the top k for DIY and ACL reach the results for COOK. Thus, the worse results for DIY and ACL cannot solely be attributed to imbalance. Our best guess is that the drop in performance then might be due to a difference between English and German. Since German contains a lot of closed compounds, i.e. one-word expressions that are composed of several simpler words, a lot of more specific terms are contained in the gold standard and the corpus. These specific terms might be less ambiguous or generic, and they might have less diverse contexts and are thus easier to identify for PageRank.

6.4. Shift-Value Experiments

In the last experiment, we used meaning shift predictions as values for the personalization vector in order to bias Page Rank towards less ambiguous terms. Table 4 compares the results for the COOK dataset with and without shift values for all the edge-weighting methods. We find an overall beneficial effect, all results improve when using the meaning shift values. We conclude that the meaning shifts are effective in biasing the network. Applying the meaning shift personalization vector has the highest effect on LMI weight, which achieved lowest scores without shift values. Cosine and co-occurrence still are the best-performing measures.

	Baseline	
Dataset	**AP**	**ROC**
COOK	0.50	0.52
DIY	0.15	0.52
ACL	0.14	0.51

Table 2: Average baseline results for all three datasets.

6.5. First-Order vs. Second-Order Association

The results from Table 6 illustrate the performance measures for different edge-weighting methods using second-order co-occurrence vectors, in comparison to Table 3 relying on first-order co-occurrence vectors. We observe a slight performance drop compared to first-order co-occurrence.

We also compared the second-order co-occurrence vectors to the first-order vectors regarding meaning shift predictions: Table 7 compares the results for the COOK dataset with and without shift values for all the edge-weighting methods using second-order co-occurrence vectors, in contrast to the respective results with first-order co-occurrence vectors in Table 4. We see an improvement in performance for the COOK dataset using second-order co-occurrence vectors, so we conclude that integrating the meaning shifts is effective with both first-order and second-order associations.

7. Conclusion

We presented a systematic comparison of term extraction variants using the PageRank model with different edge-weighting approaches. Results on the COOK corpus were found to be far better in constrast to the DIY and ACL corpora, which at first we assumed to happen because the COOK gold standard is more balanced regarding the term/non-term proportions than the two English gold standards.

In a first set of experiments, co-occurrence and cosine performed better for COOK and DIY in comparison to LMI and PMI. The imbalance in the gold standard was evaluated using P@K ('precision at k'). The results were worse for DIY and ACL across k, so we concluded that the drop in performance could be related to differences between English and German, e.g., because German contains many closed compounds that are potentially less ambiguous than simplex words and thus easier to identify for PageRank.

In a second set of experiments, we used meaning shift predictions as values for personalization vectors to bias PageRank over less ambiguous terms. We achieved an uplift in the performance measure demonstrating that the use of meaning shifts can effectively bias the network.

We further investigated the influence of association on performance by comparing first-order co-occurrence and second-order co-occurrence vectors, and saw that the use of meaning shift predictions can effectively improve the performance measure for both association types.

	Co-occurrence		Cosine		PMI		LMI	
	AP	ROC	AP	ROC	AP	ROC	AP	ROC
COOK	0.65	0.68	0.65	0.68	0.57	0.59	0.57	0.59
DIY	0.22	0.57	0.22	0.57	0.15	0.52	0.15	0.52
ACL	0.16	0.54	0.16	0.55	0.22	0.59	0.22	0.59

Table 3: Evaluation scores for all datasets using different edge-weighting methods and first-order co-occurrence.

COOK	Co-occurrence		Cosine		PMI		LMI	
	AP	ROC	AP	ROC	AP	ROC	AP	ROC
Without shift values	0.65	0.68	0.65	0.68	0.57	0.59	0.57	0.59
With shift values	0.67	0.69	0.67	0.69	0.63	0.65	0.70	0.69

Table 4: Performance evaluation with and without shift values for COOK and first-order co-occurrence.

	COOK	DIY	ACL
P@50	0.82	0.26	0.32
P@100	0.79	0.24	0.27
P@300	0.66	0.20	0.18
P@500	0.60	0.16	0.14
P@1000	0.50	0.14	0.14

Table 5: Precision at k (P@k) for five segments (k=50,100,300,500,1000) on all datasets.

	Co-occurrence		Cosine		PMI		LMI	
	AP	ROC	AP	ROC	AP	ROC	AP	ROC
COOK	0.64	0.68	0.63	0.67	0.56	0.59	0.54	0.57
DIY	0.21	0.60	0.21	0.60	0.20	0.60	0.20	0.60
ACL	0.15	0.58	0.15	0.59	0.24	0.59	0.24	0.59

Table 6: Evaluation scores for all datasets with different edge-weighting methods using second-order co-occurrence.

COOK	Co-occurrence		Cosine		PMI		LMI	
	AP	ROC	AP	ROC	AP	ROC	AP	ROC
Without shift values	0.64	0.68	0.63	0.67	0.56	0.59	0.54	0.57
With shift values	0.72	0.72	0.72	0.72	0.71	0.69	0.69	0.67

Table 7: Performance evaluation with and without shift values for COOK and second-order co-occurrence.

8. Bibliographical References

Amjadian, E., Inkpen, D., Paribakht, T. S., and Faez, F. (2016). Local-global vectors to improve unigram terminology extraction. In *Proceedings of the 5th International Workshop on Computational Terminology*, pages 2–11, Osaka, Japan.

Basili, R., Rossi, G. D., and Pazienza, M. T. (1997). Inducing terminology for lexical acquisition. In *Proceedings of the 1997 Conference on Empirical Methods in Natural Language Processing*, Providence, RI, USA.

da Silva Conrado, M., Salgueiro Pardo, T. A., and Oliveira Rezende, S. (2013). A machine learning approach to automatic term extraction using a rich feature set. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 16–23.

De Groc, C., Tannier, X., and Couto, J. (2011). GrawlTCQ: Terminology and corpora building by ranking simultaneously terms, queries and documents using graph random walks. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, pages 37–41.

Evert, S. (2005). *The Statistics of Word Co-Occurrences: Word Pairs and Collocations*. Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Florescu, C. and Caragea, C. (2017). A position-biased PageRank algorithm for keyphrase extraction. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Frantzi, K., Ananiadou, S., and Tsujii, J. (1998). The c-value/nc-value method of automatic recognition for multi-word terms. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries*, pages 585–604.

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.

Hätty, A., Schlechtweg, D., and Schulte im Walde, S. (2019). SURel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*, pages 1–8, Minneapolis, MN, USA.

Justeson, J. S. and Katz, S. M. (1995). Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Khan, M. T., Ma, Y., and Kim, J.-j. (2016). Term ranker: A graph-based re-ranking approach. In *Proceedings of the 29th International Flairs Conference*.

Maynard, D. and Ananiadou, S. (1999). Identifying contextual information for multi-word term extraction. In *Proceedings of 5th International Congress on Terminology and Knowledge Engineering*, pages 212–221.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Rapp, R. (2002). The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Sahlgren, M. (2006). *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces*. Ph.D. thesis, Stockholm University.

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

Schäfer, J., Rösiger, I., Heid, U., and Dorna, M. (2015). Evaluating noise reduction strategies for terminology extraction. In *Proceedings of the Conference on Terminology and Artificial Intelligence*, pages 123–131, Granada, Spain.

Schlechtweg, D., Oğuz, C., and Schulte im Walde, S. (2019). Second-order co-occurrence sensitivity of skip-gram with negative sampling. In *Proceedings of the 2nd Workshop on the Analysis and Interpretation of Neural Networks for Natural Language Processing*, pages 24–30, Florence, Italy.

Schulte im Walde, S. (2010). Comparing computational approaches to selectional preferences: Second-order co-occurrence vs. latent semantic clusters. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 1381–1388, Valletta, Malta.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Wang, R., Liu, W., and McDonald, C. (2016). Featureless domain-specific term extraction with minimal labelled data. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 103–112.

Zadeh, B. Q. and Handschuh, S. (2014a). The ACL RD-TEC: A dataset for benchmarking terminology extraction and classification in computational linguistics. In *Proceedings of the 4th International Workshop on Computational Terminology*, pages 52–63, Dublin, Ireland.

Zadeh, B. Q. and Handschuh, S. (2014b). Evaluation of technology term recognition with random indexing. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.

Zhanga, Z., Gaob, J., and Ciravegnab, F. (2017). SemRe-Rank: Incorporating semantic relatedness to improve automatic term extraction using personalized pagerank. *arXiv:1711.03373*.

Zhuang, Y., Xie, J., Zheng, Y., and Zhu, X. (2018). Quantifying context overlap for training word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 587–593, Brussels, Belgium.