Evaluating Automatic Term Extraction Methods on Individual Documents

Antonio Šajatović **Maja Buljan** **Jan Šnajder** **Bojana Dalbelo Bašić**
University of Zagreb, Faculty of Electrical Engineering and Computing,
Text Analysis and Knowledge Engineering Lab, Zagreb, Croatia
{antonio.sajatovic,maja.buljan,jan.snajder,bojana.dalbelo}@fer.hr

Abstract

Automatic Term Extraction (ATE) extracts terminology from domain-specific corpora. ATE is used in many NLP tasks, including Computer Assisted Translation, where it is typically applied to individual documents rather than the entire corpus. While corpus-level ATE has been extensively evaluated, it is not obvious how the results transfer to document-level ATE. To fill this gap, we evaluate 16 state-of-the-art ATE methods on full-length documents from three different domains, on both corpus and document levels. Unlike existing studies, our evaluation is more realistic as we take into account all gold terms. We show that no single method is best in corpus-level ATE, but C-Value and KeyConceptRelatendess surpass others in document-level ATE.

1 Introduction

The aim of Automatic Term Extraction (or Recognition) (ATE) is to extract terms – single words or multiword expressions (MWEs) representing domain-specific concepts – from a domain-specific corpus. ATE is widely used in many NLP tasks, such as information retrieval and machine translation. Moreover, Computer Assisted Translation (CAT) tools often use ATE methods to aid translators in finding and extracting translation equivalent terms in the target language (Costa et al., 2016; Oliver, 2017).

While corpus-based approaches to terminology extraction are the norm when building large-scale termbases (Warburton, 2014), a survey we conducted[1] showed that translators are most often interested in ATE from individual documents of various lengths, rather than entire corpora, since they typically translate on document at a time.

[1]Survey results available at http://bit.ly/2LwrTkv.

A task related to ATE is Automatic Keyword and Keyphrase Extraction (AKE), which deals with the extraction of single words and MWEs from a single document. Unlike ATE, which aims to capture domain-specific terminology, keywords and keyphrases extracted by AKE should capture the main topics of a document. Consequently, there will only be a handful of representative keyphrases for a document (Turney, 2000). In spite of these differences, several AKE methods were adapted for ATE (Zhang et al., 2016).

While corpus-level ATE methods, as well as AKE methods, have been extensively evaluated in the literature, it is not obvious how the results transfer to document-level ATE, which is how ATE is typically used for CAT. In this paper, we aim to close this gap and present an evaluation study that considers both corpus- and document-level ATE. We evaluate 16 state-of-the-art ATE methods, including modified AKE methods. Furthermore, addressing another deficiency in existing evaluations, we evaluate the methods using a complete set of gold terms, making the evaluation more realistic.

2 Related Work

Most ATE methods begin with the extraction and filtering of candidate terms, followed by candidate term scoring and ranking. Because of divergent candidate extraction and filtering step implementations, many existing ATE evaluations are not directly comparable. Zhang et al. (2008) were among the first to compare several scoring and ranking methods, using the same candidate extraction and filtering step and the UAP metric on a custom Wikipedia corpus and GENIA (Kim et al., 2003) corpus. In a followup work, they developed JATE 2.0 (Zhang et al., 2016), with 10 ATE methods available out-of-the-box, that were evaluated

on GENIA and ACL RD-TEC (Zadeh and Handschuh, 2014) using the "precision at K" metric. A similar toolkit, ATR4S (Astrakhantsev, 2018), which implements 15 ATE methods, was evaluated on even more datasets using "average precision at K". All abovementioned studies were carried out corpus-level, and rely on exact matching between extracted terms and a subset of gold terms. The latter makes such evaluations unrealistic because it disregards the contribution of the candidate extraction and filtering step. The subset is selected by considering only the gold terms that appear in the output above the cut off of at level K, which is used to discriminate between real terms and non-terms. A general consensus is that there is no single best method (Zhang et al., 2008; Astrakhantsev, 2018; Zhang et al., 2018).

To the best of our knowledge, we are the first to carry out a document-level ATE evaluation, and take into account all gold terms instead of only a subset. To this end, we use a single ATE toolkit, to allow for a direct comparison among different term-ranking methods, by using the same preprocessing and filters. Our toolkit of choice is ATR4S, because it has the most diverse set of methods, many of which are state-of-the-art.

3 Term Extraction Methods

ATE methods may be roughly grouped by the type of information used for scoring the term candidates (Astrakhantsev, 2018). Due to the sheer number of ATE methods, we only describe the main principle behind each group and list the main methods. In the evaluation, we consider a total of 16 methods from ATR4S, covering all groups.

Frequency. Most methods rests on the assumption that a higher term candidate frequency implies a higher likelihood that a candidate is an actual term. Among these are AverageTermFrequency (Zhang et al., 2016), ResidualIDF (Zhang et al., 2016) (adapted from AKE), TotalTF-IDF (Evans and Lefferts, 1995), C-Value (Frantzi et al., 2000), Basic (Buitelaar et al., 2013), ComboBasic (Astrakhantsev et al., 2015). Two notable ATE-adapted AKE methods, not provided in ATR4S, are Chi-Square (Matsuo and Ishizuka, 2004) and Rapid Keyword Extraction (Rose et al., 2010).

Context. A handful of methods adopt the distributional hypothesis (Harris, 1954) and consider the context in which the term candidate appears, such as DomainCoherence (Buitelaar et al., 2013) and NC-Value (Frantzi et al., 2000).

Reference corpora. Several methods compare the domain corpus and reference corpus term frequencies, assuming that the difference between them can be used to distinguish terms from non-terms. Domain pertinence (DomPertinence) (Meijer et al., 2014) is the simplest one, while Relevance (Peñas et al., 2001) and Weirdness (Ahmad et al., 1999) can be considered its modifications.

Topic modeling. Topic information can also be used instead of term frequency information, as in NovelTM (Li et al., 2013).

Wikipedia. Several methods use Wikipedia instead of term frequency to distinguish between candidate and actual terms, such as LinkProbability (Astrakhantsev, 2014) and KeyConceptRelatedness (Astrakhantsev, 2014). In addition to Wikipedia, KeyConceptRelatedness also relies on keyphrase extraction and semantic relatedness.

Re-ranking. Methods from this group use other ATE methods as features, and attempt to learn the importance of each feature in an unsupervised or supervised setting. Glossary Extraction (Park et al., 2002) extends Weirdness, while Term Extraction (Sclano and Velardi, 2007) further extends Glossary Extraction. SemRe-Rank (Zhang et al., 2018) is a generic approach that incorporates semantic relatedness to re-rank terms. Both da Silva Conrado et al. (2013) and Yuan et al. (2017) use a variety of features in a supervised binary term classifier. A weakly supervised bootstrapping approach called fault tolerant learning (Yang et al., 2010) has been extended for deep learning (Wang et al., 2016). The following methods are the only ones from this group available in ATR4S and therefore the only ones evaluated: PostRankDC (Buitelaar et al., 2013) combines DomainCoherence with Basic, while both PU-ATR (supervised) (Astrakhantsev, 2014) and Voting (unsupervised) (Zhang et al., 2008) use the same five features as implemented in ATR4S. In our study, we distinguish between the original Voting_5 and its variant, Voting_3, in which the two Wikipedia-based features are removed to gauge their impact.

Dataset	# Docs	# Terms	% MWEs	Avg terms/doc
Patents	16	1585	86	151
TTCm	37	160	55	51
TTCw	102	190	72	33

Table 1: Full-length document datasets statistics

4 Evaluation

Datasets. There exists a number of ATE datasets compiled using various criteria, comprised of abstracts or full-length documents. As our focus is document-level ATE, our criteria were that the dataset has to consist of full-length documents and be manually annotated. This ruled out the two most popular datasets used in most of previous works, GENIA and ACL RD-TEC, as the former consists of abstracts only and the latter is not manually annotated. Instead, we were able to find only three datasets that meet both of our requirements. One is the Patents dataset (Judea et al., 2014), which has the least number of documents, but most terms. It consists of electrical engineering patents manually annotated by three annotators. The other two datasets were created under the TTC project.[2] Both TTC-wind (TTCw) and TTC-mobile (TTCm) were compiled by crawling the Web, and then manually filtered. These datasets are listed in Table 1. They all cover different domains and have a different number of documents and terms per document. Since most of the gold terms in all three datasets are MWEs, there could be a slight bias toward methods designed to extract only the MWEs, such as Basic or ComboBasic.

Extraction setup. ATR4S collects n-grams up to a specified size (4 by default), which are filtered through the stop words, noise words, and POS-pattern filters (cf. Astrakhantsev (2018) for details). The collected term candidates are then scored and ranked using one of the 16 methods. In order to evaluate each method's output, we lemmatize each term candidate and repeat the same procedure for each gold term. We use the same default settings for both extraction levels.[3]

Metrics. Following Zhang et al. (2018), we differentiate between two types of true positives: (1) Actual True Positives (ATP), which are all the terms contained in the gold set, and (2) Recoverable True Positives (RTP), which are the intersection of the extracted candidate terms after filtering and the gold set terms. To separate real terms from non-terms based on their scores, a cutoff at rank K has to be set. Setting K equal to |RTP| is the default choice in the majority of previous work (Zhang et al., 2016; Astrakhantsev, 2018; Zhang et al., 2018), but any such metric can easily become too optimistic because $|\text{RTP}| \leq |\text{ATP}|$, i.e., evaluation becomes oblivious to the candidate extraction and filtering step.

To obtain a more realistic score, we calculate ATP for both the corpus- and document-level ATE. In the former, ATP is equal to the entire gold set, while in the latter we build the gold set of each document by checking if the lemma of any term from the gold set is a substring of the entire lemmatized document. Following Zhang et al. (2018), we use two measures to evaluate the ATR4S output: F_1 score and average precision (AvP), at levels |RTP| and |ATP|. We define $(\text{retrieved}_i)_{i=1}^{K}$ as the list of ranked extracted terms, up to rank K. The rank-insensitive F_1 score is calculated as the harmonic mean of P@K and R@K:

$$\text{P@K} = \frac{|(\text{retrieved}_i)_{i=1}^{K} \cap \{\text{relevant}\}|}{|(\text{retrieved}_i)_{i=1}^{K}|} \quad (1)$$

$$\text{R@K} = \frac{|(\text{retrieved}_i)_{i=1}^{K} \cap \{\text{relevant}\}|}{|\{\text{relevant}\}|} \quad (2)$$

$$\text{F}_1\text{@K} = 2 \cdot \frac{\text{P@K} \cdot \text{R@K}}{\text{P@K} + \text{R@K}} \quad (3)$$

To evaluate the ranking performance of an ATE method, we use AvP@K, a standard ATE metric:

$$\text{AvP@K} = \frac{1}{\text{K}} \sum_{\text{k}=1}^{\text{K}} \text{P@k} \quad (4)$$

5 Results

Corpus-level extraction. As mentioned above, in corpus-level ATE, the input is a collection of documents. All methods from Section 3 were developed with the aim of extracting terms from a domain-specific corpus. The F_1 and AvP scores for this level are shown in the left half of Table 2.

C-Value most often performs best, compared to both frequency-based and all other methods, and thus may be considered a strong baseline. Voting_3 has negligibly lower scores than its more feature-rich variant, Voting_5. LinkProbability, relying on a normalized frequency of a term being a hyperlink in Wikipedia pages, most often has the lowest score. Our results corroborate earlier findings

[2] http://www.ttc-project.eu/

[3] https://github.com/ispras/atr4s/tree/master/configs

	Corpus-level ATE												Document-level ATE											
	Patents				TTCm				TTCw				Patents				TTCm				TTCw			
	ATP		RTP		ATP		RTP		ATP		RTP		ATP		RTP		ATP		RTP		ATP		RTP	
	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP	F_1	AvP
AvgTermFreq	.36	.46	.29	.53	.15	.16	.11	.17	.06	.10	.07	.11	.34	.44	.26	.53	.19	.21	.11	.22	.22	.36	.20	.50
ResidualIDF	.36	.45	.27	.51	.04	.07	.03	.11	.02	.02	.00	.00	.34	.43	.26	.51	.19	.21	.11	.22	.22	.33	.19	.41
TotalTF-IDF	.35	.45	.28	.53	.27	.34	.28	.34	.15	.18	.13	.20	.27	.26	.16	.28	.09	.11	.05	.11	.10	.15	.07	.24
C-Value	**.42**	**.55**	.33	.63	**.35**	**.39**	**.33**	.40	.26	.42	.23	.51	**.38**	**.53**	**.32**	**.65**	.14	.20	.09	.29	.23	.36	.20	.50
Basic	.37	.47	.29	.53	.20	.33	.21	.35	.26	**.46**	.27	**.59**	.36	.47	.30	.57	.14	.19	.09	.25	.24	.35	.20	.47
ComboBasic	.37	.47	.30	.53	.20	.33	.21	.35	.26	.45	**.27**	.59	.36	.47	.30	.56	.14	.18	.08	.24	.23	.34	.19	.46
Relevance	.39	.47	.29	.54	.18	.34	.15	**.56**	.13	.23	.11	.35	.37	.44	.25	.52	.10	.18	.07	.35	.10	.10	.06	.14
DomPertinence	.39	.47	.29	.54	.18	.32	.16	.52	.12	.19	.09	.28	.37	.44	.25	.52	.10	.18	.07	**.35**	.10	.10	.06	.14
Weirdness	.36	.42	.27	.46	.29	.30	.29	.30	.13	.23	.13	.29	.35	.46	.27	.55	.20	.23	.12	.25	.24	.37	.21	.52
NovelTM	.39	.51	.31	.58	.11	.17	.11	.19	.08	.03	.01	.00	.36	.50	.30	.61	.15	.20	.09	.26	.25	.39	.23	.50
LinkProbability	.30	.40	.24	.50	.03	.01	.02	.00	.02	.00	.00	.00	.31	.41	.26	.50	.22	.23	.12	.23	.25	.30	.19	.32
KeyConceptRel	.28	.40	.21	.53	.27	.39	.25	.43	.21	.38	.23	.51	.30	.45	.26	.58	**.23**	**.29**	**.16**	.33	**.31**	**.46**	**.28**	**.60**
PostRankDC	.35	.44	.27	.49	.26	.31	.25	.32	.15	.33	.16	.44	.35	.46	.28	.55	.16	.19	.09	.22	.23	.34	.19	.47
PU-ATR	.39	.54	**.34**	**.65**	.27	.39	.23	.46	**.28**	.44	.26	.55	.37	.49	.31	.58	.15	.19	.09	.25	.23	.35	.19	.46
Voting_5	.40	.53	.32	.62	.26	.34	.24	.35	.24	.31	.20	.35	.37	.52	.32	.64	.18	.25	.13	.33	.24	.36	.20	.49
Voting_3	.39	.50	.31	.58	.29	.37	.27	.38	.21	.31	.19	.36	.35	.49	.30	.60	.13	.21	.10	.31	.19	.28	.15	.39

Table 2: Scores for corpus-level ATE (left half) and mean scores for document-level ATE (right half).

(Astrakhantsev, 2018; Zhang et al., 2018) that no single ATE method is consistently the best in a corpus-level setting. A notable trend is that most methods have higher F_1 scores in the ATP case and lower AvP scores in the RTP case. Both can be explained by noting that $|\text{ATP}| \geq |\text{RTP}|$ and F_1 is not rank-sensitive, while AvP is. I.e., the larger the gold term set (ATP), the more likely an actual term will be above the fixed cut-off level $K = \text{ATP}$, while the smaller the gold term set is (RTP), the more likely an actual term will be highly ranked, as there are less terms to rank.

Document-level extraction. In document-level extraction, the input to ATE is a single document. Document-level scores are shown in the right half of Table 2. C-Value is not the overall best frequency-based method, as it was on the corpus-level. However, it outperforms all other methods in a highly technical domain (Patents dataset), for which it was originally developed. A clear overall winner is KeyConceptRelatedness. Its good performance may be attributed to its hybrid nature: using semantic relatedness between keyphrases and candidate terms. Voting with Wikipedia-based features is better overall than the variant without them, especially when considering the more optimistic RTP metrics. TotalTF-IDF is by definition ill-equipped for document-level ATE (log term becomes zero), which is why it is the worst performing method.

	Patents	TTCm	TTCw
AvgTermFreq	−.35	−.06	−.42
ResidualIDF	−.33	−.03	−.55
TotalTF-IDF	−.38	−.12	−.33
C-Value	**.01**	−.11	−.25
Basic	−.09	−.25	−.34
ComboBasic	−.06	−.24	−.32
Relevance	−.06	−.19	−.23
DomPertinence	−.06	−.19	−.23
Weirdness	−.39	−.08	−.43
NovelTM	−.21	−.19	−.35
LinkProbability	−.03	−.46	−.56
KeyConceptRel	−.44	−.26	−.25
PostRankDC	−.17	**.02**	−.25
PU-ATR	**.15**	−.17	−.29
Voting_5	−.09	−.26	−.25
Voting_3	−.06	−.09	−.09

Table 3: Correlation between ATP AvP and document length for document-level ATE.

For the ATP case, we statistically compared[4] C-value, KeyConceptRelatedness, and AvgTermFrequency (baseline) methods, for both F1 and AvP, on all three datasets. The comparison confirmed that C-value significantly outperform other two methods on Patents dataset and that KeyConceptRelatedness significantly outperforms other two methods on TTCm and TTCw dataset, and this holds for both metrics.

[4]We used the non-parametric Friedman ANOVA for dependent samples with post-hoc comparison using Wilcoxon matched paired test and Bonferroni-corrected paired t-test, depending on whether normality assumption was met.

	Patents		TTCm		TTCw	
	% MWEs	Recall	% MWEs	Recall	% MWEs	Recall
C-Value	53	.41	77	.30	78	.30
KeyConceptRel	21	.18	31	.28	35	.32

Table 4: Percentage of MWEs and recall for document-level ATE

Given the difference in performance between corpus-level and document-level ATE, document length is another practical consideration when choosing the appropriate ATE method. We calculated the Pearson correlation coefficient between the document lengths and ATP AvP scores for document-level ATE, shown in table 3. Correlation coefficients for individual methods vary across datasets – predominantly, as the document length increases, the ATP AvP score decreases, or there is almost no correlation.

Additionally, we analysed the recall of top-performing document-level ATE methods with regards to MWEs, depending on their share in the gold terms for a given document. The percentage of MWEs in gold terms per dataset is given in Table 1. Table 4 shows the percentage of MWEs in the output of a given ATE method at ATP cut-off, averaged over all documents of a particular dataset, as well as the per-document recall for MWEs, averaged over all documents. The performance varies across datasets, but C-Value – a frequency-based method – modestly outperforms KeyConceptRelatedness in identifying multiword terms.

Taken together, our results clearly show that corpus-level performances do not linearly transfer to document-level performances, the case in point being the KeyConceptRelatedness ATE method.

6 Conclusion

Motivated by the use of ATE in Computer Aided Translation, we evaluated 16 ATE methods in a novel setting: apart from using a corpus as a source of terms, we also consider using individual documents only. Unlike previous ATE work, we use metrics that distinguish between actual and recoverable true positives. Our findings confirm that no single ATE method is consistently the best in corpus-level ATE. We show that for document-level ATE most of the methods perform comparable, with two exceptions: (1) C-Value performs exceptionally well in highly technical domains, and (2) KeyConceptRelatedness outperforms all other methods on two other domains. We thus recommend using C-Value for corpus-level ATE or document-level ATE in a highly technical domain, and KeyConceptRelatedness for document-level ATE in non-technical domains.

Our work opens up a new line of research, namely an investigation into ATE methods more suitable for single-document input, possibly employing related AKE methods. Another research topic is single-document bilingual ATE.

Acknowledgments

The authors would like to thank Maria Pia di Buono for her support.

References

Khurshid Ahmad, Lee Gillam, and Lena Tostevin. 1999. University of Surrey Participation in TREC 8: Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER). In *The Eighth Text REtrieval Conference (TREC 8)*, pages 1–8.

Nikita Astrakhantsev. 2014. Automatic term acquisition from domain-specific text collection by using Wikipedia. *Proceedings of the Institute for System Programming*, 26(4):7–20.

Nikita Astrakhantsev. 2018. ATR4S: toolkit with state-of-the-art automatic terms recognition methods in Scala. *Language Resources and Evaluation*, 52(3):853–872.

Nikita Astrakhantsev, Denis G. Fedorenko, and D. Yu. Turdakov. 2015. Methods for automatic term recognition in domain-specific text collections: A survey. *Programming and Computer Software*, 41(6):336–349.

Paul Buitelaar, Georgeta Bordea, and Tamara Polajnar. 2013. Domain-independent term extraction through domain modelling. In *The 10th International Conference on Terminology and Artificial Intelligence (TIA 2013), Paris, France*.

Hernani Costa, Gloria Corpas Pastor, Míriam Seghiri Domínguez, and Anna Zaretskaya. 2016. Nine Terminology Extraction Tools: Are they useful for translators? *Multilingual*, 27(3).

David A. Evans and Robert G. Lefferts. 1995. CLARIT-TREC experiments. In *Information Processing and Management: an International Journal*, volume 31, pages 385–395. Pergamon Press, Inc.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method. *International Journal on Digital Libraries*, 3(2):115–130.

Zellig S. Harris. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Alex Judea, Hinrich Schütze, and Sören Brügmann. 2014. Unsupervised Training Set Generation for Automatic Acquisition of Technical Terminology in Patents. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical Papers*, pages 290–300.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. GENIA corpus – a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Sujian Li, Jiwei Li, Tao Song, Wenjie Li, and Baobao Chang. 2013. A novel topic model for automatic term extraction. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 885–888. ACM.

Yutaka Matsuo and Mitsuru Ishizuka. 2004. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169.

Kevin Meijer, Flavius Frasincar, and Frederik Hogenboom. 2014. A semantic approach for extracting domain taxonomies from text. *Decision Support Systems*, 62:78–93.

Antoni Oliver. 2017. A system for terminology extraction and translation equivalent detection in real time. *Machine Translation*, 31(3):147–161.

Youngja Park, Roy J. Byrd, and Branimir K. Boguraev. 2002. Automatic Glossary Extraction: Beyond Terminology Identification. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Anselmo Peñas, Felisa Verdejo, Julio Gonzalo, et al. 2001. Corpus-based terminology extraction applied to information access. In *Proceedings of Corpus Linguistics*, volume 2001, page 458. Citeseer.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory*, pages 1–20.

Francesco Sclano and Paola Velardi. 2007. TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities. In *Enterprise Interoperability II*, pages 287–290. Springer.

Merley da Silva Conrado, Thiago A Salgueiro Pardo, and Solange Oliveira Rezende. 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.

Peter D. Turney. 2000. Learning Algorithms for Keyphrase Extraction. *Information retrieval*, 2(4):303–336.

Rui Wang, Wei Liu, and Chris McDonald. 2016. Featureless Domain-Specific Term Extraction with Minimal Labelled Data. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 103–112.

Kara Warburton. 2014. Narrowing the gap between termbases and corpora in commercial environments. In *LREC 2014 Proceedings*, pages 722–727.

Yuhang Yang, Hao Yu, Yao Meng, Yingliang Lu, and Yingju Xia. 2010. Fault-Tolerant Learning for Term Extraction. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*.

Yu Yuan, Jie Gao, and Yue Zhang. 2017. Supervised Learning for Robust Term Extraction. In *2017 International Conference on Asian Language Processing (IALP)*, pages 302–305. IEEE.

Behrang Q. Zadeh and Siegfried Handschuh. 2014. 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 (Computerm)*, pages 52–63.

Ziqi Zhang, Jie Gao, and Fabio Ciravegna. 2016. JATE 2.0: Java Automatic Term Extraction with Apache Solr. In *The Proceedings of the 10th Language Resources and Evaluation Conference*.

Ziqi Zhang, Jie Gao, and Fabio Ciravegna. 2018. SemRe-Rank: Improving Automatic Term Extraction by Incorporating Semantic Relatedness with Personalised PageRank. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5):57.

Ziqi Zhang, José Iria, Christopher Brewster, and Fabio Ciravegna. 2008. A Comparative Evaluation of Term Recognition Algorithms. In *The International Conference on Language Resources and Evaluation (LREC)*, volume 5.