

# Automatic Keyword Extraction on Twitter

Luís Marujo<sup>1,2,3</sup>, Wang Ling<sup>1,2,3</sup>, Isabel Trancoso<sup>2,3</sup>, Chris Dyer<sup>1</sup>, Alan W. Black<sup>1</sup>,  
Anatole Gershman<sup>1</sup>, David Martins de Matos<sup>2,3</sup>, João P. Neto<sup>2,3</sup>, and Jaime Carbonell<sup>1</sup>

<sup>1</sup> Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

<sup>2</sup> Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal;

<sup>3</sup> INESC-ID, Lisbon, Portugal

{luis.marujo, wang.ling, isabel.trancoso, david.matos, joao.neto}@inesc-id.pt  
{cdyer, awb, anatoleg, jgc}@cs.cmu.edu,

## Abstract

In this paper, we build a corpus of tweets from Twitter annotated with keywords using crowdsourcing methods. We identify key differences between this domain and the work performed on other domains, such as news, which makes existing approaches for automatic keyword extraction not generalize well on Twitter datasets. These datasets include the small amount of content in each tweet, the frequent usage of lexical variants and the high variance of the cardinality of keywords present in each tweet. We propose methods for addressing these issues, which leads to solid improvements on this dataset for this task.

## 1 Introduction

Keywords are frequently used in many occasions as indicators of important information contained in documents. These can be used by human readers to search for their desired documents, but also in many Natural Language Processing (NLP) applications, such as Text Summarization (Pal et al., 2013), Text Categorization (Özgür et al., 2005), Information Retrieval (Marujo et al., 2011a; Yang and Nyberg, 2015) and Question Answering (Liu and Nyberg, 2013). Many automatic frameworks for extracting keywords have been proposed (Riloff and Lehnert, 1994; Witten et al., 1999; Turney, 2000; Medelyan et al., 2010; Litvak and Last, 2008). These systems were built for more formal domains, such as news data or Web data, where the content is still produced in a controlled fashion.

The emergence of social media environments, such as Twitter and Facebook, has created a framework for more casual data to be posted online.

These messages tend to be shorter than web pages, especially on Twitter, where the content has to be limited to 140 characters. The language is also more casual with many messages containing orthographical errors, slang (e.g., *cday*), abbreviations among domain specific artifacts. In many applications, that existing datasets and models tend to perform significantly worse on these domains, namely in Part-of-Speech (POS) Tagging (Gimpel et al., 2011), Machine Translation (Jelh et al., 2012; Ling et al., 2013), Named Entity Recognition (Ritter et al., 2011; Liu et al., 2013), Information Retrieval (Efron, 2011) and Summarization (Duan et al., 2012; Chang et al., 2013).

As automatic keyword extraction plays an important role in many NLP tasks, building an accurate extractor for the Twitter domain is a valuable asset in many of these applications. In this paper, we propose an automatic keyword extraction system for this end and our contributions are the following ones:

1. Provide a annotated keyword annotated dataset consisting of 1827 tweets. These tweets are obtained from (Gimpel et al., 2011), and also contain POS annotations.
2. Improve a state-of-the-art keyword extraction system (Marujo et al., 2011b; Marujo et al., 2013) for this domain by learning additional features in an unsupervised fashion.

The paper is organized as follows: Section 2 describes the related work; Section 3 presents the annotation process; Section 4 details the architecture of our keyword extraction system; Section 5 presents experiments using our models and we conclude in Section 6.

## 2 Related Work

Both supervised and unsupervised approaches have been explored to perform key word extraction. Most of the automatic keyword/keyphrase extraction methods proposed for social media data, such as tweets, are unsupervised methods (Wu et al., 2010; Zhao et al., 2011; Bellaachia and Al-Dhelaan, 2012). However, the TF-IDF across different methods remains a strong unsupervised baseline (Hasan and Ng, 2010). These methods include adaptations to the PageRank method (Brin and Page, 1998) including TextRank (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004), and Topic PageRank (Liu et al., 2010).

Supervised keyword extraction methods formalize this problem as a binary classification problem of two steps (Riloff and Lehnert, 1994; Witten et al., 1999; Turney, 2000; Medelyan et al., 2010; Wang and Li, 2011): candidate generation and filtering of the phrases selected before. MAUI toolkit-indexer (Medelyan et al., 2010), an improved version of the KEA (Witten et al., 1999) toolkit including new set of features and more robust classifier, remains the state-of-the-art system in the news domain (Marujo et al., 2012).

To the best of our knowledge, only (Li et al., 2010) used a supervised keyword extraction framework (based on KEA) with additional features, such as POS tags to performed keyword extraction on Facebook posts. However, at that time Facebook status updates or posts did not contained either hashtags or user mentions. The size of Facebook posts is frequently longer than tweets and has less abbreviations since it is not limited by number of character as in tweets.

## 3 Dataset

The dataset <sup>1</sup> contains 1827 tweets, which are POS tagged in (Gimpel et al., 2011). We used Amazon Mechanical Turk, an crowdsourcing market, to recruit eleven annotators to identify keywords in each tweet. Each annotator highlighted words that he would consider a keyword. No specific instructions about what words can be keywords (e.g., “urls are not keywords”), as we wish to learn what users find important in a tweet. It is also acceptable for tweets to not contain keywords, as some tweets simply do not contain important in-

<sup>1</sup>The corpus is submitted as supplementary material.

formation (e.g., retweet). The annotations of each annotator are combined by selecting keywords that are chosen by at least three annotators. We also divided the 1827 tweets into 1000 training samples, 327 development samples and 500 test samples, using the splits as in (Gimpel et al., 2011).

## 4 Automatic Keyword Extraction

There are many methods that have been proposed for keyword extraction. TF-IDF is one of the simplest approaches for this end (Salton et al., 1975). The  $k$  words with the highest TF-IDF value are chosen as keywords, where  $k$  is optimized on the development set. This works quite well in text documents, such as news articles, as we wish to find terms that occur frequently within that document, but are not common in the other documents in that domain. However, we found that this approach does not work well in Twitter as tweets tend to be short and generally most terms occur only once, including their keywords. This means that the term frequency component is not very informative as the TF-IDF measure will simply benefit words that rarely occur, as these have a very low inverse document frequency component.

A strong baseline for Automatic Keyword Extraction is the MAUI toolkit-indexer toolkit (Medelyan et al., 2010). The system extracts a list of candidate keywords from a document and trains a decision tree over a large set of hand engineered features, also including TF-IDF, in order to predict the correct keywords on the training set. Once trained, the toolkit extracts a list of keyword candidates from a tweet and returns a ranked list of candidates. The top  $k$  keywords are selected as answers. The parameter  $k$  is maximized on the development set.

From this point, we present two extensions to the MAUI system to address many challenges found in this domain.

### 4.1 Unsupervised Feature Extraction

The first problem is the existence of many lexical variants in Twitter (e.g., “cats vs. catz”). While variants tend to have the same meaning as their standardized form, the proposed model does not have this information and will not be able to generalize properly. For instance, if the term “John” is labelled as keyword in the training set, the model would not be able to extract “Jooohn” as keyword as it is in a different word form. One way to ad-

dress this would be using a normalization system either built using hand engineered rules (Gouws et al., 2011) or trained using labelled data (Han and Baldwin, 2011; Chrupała, 2014). However, these systems are generally limited as these need supervision and cannot scale to new data or data in other languages. Instead, we will use unsupervised methods that leverage large amounts of unannotated data. We used two popular methods for this purpose: Brown Clustering and Continuous Word Vectors.

#### 4.1.1 Brown Clustering

It has been shown in (Owoputi et al., 2013) that Brown clusters are effective for clustering lexical variants. The algorithm attempts to find a clusters distribution to maximize the likelihood of each cluster predicting the next one, under the HMM assumption. Thus, words "yes", "yep" and "yesss" are generally inserted into the same cluster as these tend occur in similar contexts. It also builds an hierarchical structure of clusters. For instance, the clusters 11001 and 11010, share the first three nodes in the hierarchically 110. Sharing more tree nodes tends to translate into better similarity between words within the clusters. Thus, a word a 11001 cluster is simultaneously in clusters 1, 11, 110, 1100 and 11001, and a feature can be extracted for each cluster. In our experiments, we used the dataset with 1,000 Brown clusters made available by Owoputi et al. (Owoputi et al., 2013)<sup>2</sup>.

#### 4.1.2 Continuous Word Vectors

Word representations learned from neural language models are another way to learn more generalizable features for words (Collobert et al., 2011; Huang et al., 2012). In these models, a hidden layer is defined that maps words into a continuous vector. The parameters of this hidden layer are estimated by maximizing a goal function, such as the likelihood of each word predicting surrounding words (Mikolov et al., 2013; Ling et al., 2015). In our work, we used the structured skip-ngram goal function proposed in (Ling et al., 2015) and for each word we extracted its respective word vector as features.

## 4.2 Keyword Length Prediction

The second problem is the high variance in terms of number of keywords per tweet. In larger doc-

uments, such as a news article, contain approximately 3-5 keywords, so extracting 3 keywords per document is a reasonable option. However, this would not work in Twitter, since the number of keywords can be arbitrary small. In fact, many tweets contain less than three words, in which case the extractor would simply extract all words as keywords, which would be incorrect. One alternative is to choose a ratio between the number of words and number of keywords. That is, we define the number of keywords in a tweet as the ratio between number of words in the tweet and  $k$ , which is maximized on the development set. That is, if we set  $k = 3$ , then we extract one keyword for every three words.

Finally, a better approach is to learn a model to predict the number of keywords using the training set. Thus, we introduced a model that attempts to predict the number of keywords in each tweet based on a set of features. This is done using linear regression, which extracts a feature set from an input tweet  $f_1, \dots, f_n$  and returns  $y$ , the expected number of keywords in the tweet. As features we selected the number of words in the input tweet with the intuition that the number of keywords tends to depend on the size of the tweet. Furthermore, (2) we count the number of function words and non-function words in the tweet, emphasizing the fact that some types of words tend to contribute more to the number of keywords in the tweet. The same is done for (3) hashtags and at mentions. Finally, (4) we also count the number of words in each cluster using the trained Brown clusters.

## 5 Experiments

Experiments are performed on the annotated dataset using the train, development and test splits defined in Section 3. As baselines, we reported results using a TF-IDF, the default MAUI toolkit, and our own implementation of (Li et al., 2010) framework. In all cases the IDF component was computed over a collection of 52 million tweets. Results are reported on rows 1 and 2 in Table 1, respectively. The parameter  $k$  (column Nr. Keywords) defines the number of keywords extracted for each tweet and is maximized on the development set. Evaluation is performed using F-measure (column F1), where the precision (column P) is defined as the ratio of extracted keywords that are correct and the number of extracted keywords, and the recall (column R) is de-

<sup>2</sup><http://www.ark.cs.cmu.edu/TweetNLP/clusters/50mpaths2>

	System	Nr. Keywords	Dev			Test		
			P	R	F1	P	R	F1
1	TF-IDF	15	19.31	83.58	29.97	20.21	85.17	31.16
2	(Li et al., 2010)	4	48.81	50.05	49.42	51.78	50.92	51.35
3	MAUI (Default)	4	51.31	52.47	51.88	53.97	53.15	53.56
4	MAUI (Word Vectors)	4	52.70	53.50	53.10	55.80	54.45	55.12
5	MAUI (Brown)	4	68.08	74.11	70.97	71.95	75.01	73.45
6	MAUI (Brown+Word Vectors)	4	<b>68.46</b>	<b>75.05</b>	<b>71.61</b>	<b>72.05</b>	<b>75.16</b>	<b>73.57</b>
7	MAUI (Trained on News)	4	49.12	49.71	49.41	52.40	51.19	51.79

Table 1: F-measure, precision and recall results on the Twitter keyword dataset using different feature sets.

	Selection	Nr. Keywords	Dev			Test		
			P	R	F1	P	R	F1
1	Fixed	4	68.46	75.05	71.61	72.05	75.16	73.57
2	Ratio	$N//3$	65.70	<b>82.69</b>	73.22	69.48	<b>83.8</b>	75.97
3	Regression	$y + k$	<b>67.55</b>	80.9	<b>73.62</b>	71.81	82.55	<b>76.81</b>

Table 2: F-measure, precision and recall results on the Twitter keyword dataset using different keyword selection methods.

defined as the ratio between the number of keywords correctly extracted and the total number of keywords in the dataset. We can see that the TF-IDF, which tends to be a strong baseline for keyword/keyphrase extraction (Hasan and Ng, 2010), yields poor results. In fact, the best value for  $k$  is 15, which means that the system simply retrieves all words as keywords in order to maximize recall. This is because most keywords only occur once<sup>3</sup>, which makes the TF component not very informative. On the other hand, the MAUI baseline performs significantly better, this is because of the usage of many hand engineered features using lists of words and Wikipedia, rather than simply relying on word counts.

Next, we introduce features learnt using an unsupervised setup, namely, word vectors and brown clusters in rows 3 and 4, respectively. These were trained on the same 52 million tweets used for computing the IDF component. Due to the large size of the vocabulary, word types with less than 40 occurrences were removed. We observe that while both features yield improvements over the baseline model in row 2, the improvements obtained using Brown clustering are far more significant. Combining both features yields slightly higher results, reported on row 5. Finally, we also test training the system with all features on an out-

of-domain keyword extraction corpus composed by news documents (Marujo et al., 2012). Results are reported on row 6, where we can observe a significant domain mismatch problem between these two domains as results drop significantly.

We explored different methods for choosing the number of keywords to be extracted in Table 2. The simplest way is choosing a fixed number of keywords  $k$  and tune this value in the development set. Next, we can also define the number of keywords as the ratio  $\frac{N}{k}$ , where  $N$  is the number of words in the tweet, and  $k$  is the parameter that we wish to optimize. Finally, the number of keywords can also be estimated using a linear regressor as  $y = f_1 w_1, \dots, f_n w_n$ , where  $f_1, \dots, f_n$  denote the feature set and  $w_1, \dots, w_n$  are the parameters of the model trained on the training set. Once the model is trained, the number of keywords selected for each tweet is defined as  $y + k$ , where  $k$  is inserted to adjust  $y$  to maximize F-measure on the development set. Results using the best system using Brown clusters and word vectors are described in Table 2. We can observe that defining the number of keywords as a fraction of the number of words in the tweet, yields better results (row 2) yields better overall results than fixing the number of extracted keywords (row 1). Finally, training a predictor for the number of keywords yields further improvements (row 3) over a simple ratio of the

<sup>3</sup>6856 out of 7045 keywords are singletons

number of input words.

## 6 Conclusions

In this work, we built a corpus of tweets annotated with keywords, which was used to build and evaluate a system to automatically extract keywords on Twitter. A baseline system is defined using existing methods applied to our dataset and improved significantly using unsupervised feature extraction methods. Furthermore, an additional component to predict the number of keywords in a tweet is also built. In future work, we plan to use the keyword extraction to perform numerous NLP tasks on the Twitter domain, such as Document Summarization.

## Acknowledgements

This work was partially supported by Fundação para a Ciência e Tecnologia (FCT) through the grants CMUP-EPB/TIC/0026/2013, SFRH/BD/51157/2010, UID/CEC/50021/2013, and the Carnegie Mellon Portugal Program. The authors also wish to thank the anonymous reviewers for their helpful comments.

## References

- Abdelghani Bellaachia and Mohammed Al-Dhelaan. 2012. Ne-rank: A novel graph-based keyphrase extraction in twitter. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '12*, pages 372–379, Washington, DC, USA. IEEE Computer Society.
- Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30:107–117.
- Yi Chang, Xuanhui Wang, Qiaozhu Mei, and Yan Liu. 2013. Towards twitter context summarization with user influence models. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM '13*, pages 527–536, New York, NY, USA. ACM.
- Grzegorz Chrupała. 2014. Normalizing tweets with edit scripts and recurrent neural embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 680–686, Baltimore, Maryland, June. Association for Computational Linguistics.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12.
- Yajuan Duan, Zhumin Chen, Furu Wei, Ming Zhou, and Heung-Yeung Shum. 2012. Twitter topic summarization by ranking tweets using social influence and content quality. In *Proceedings of COLING 2012*, pages 763–780. The COLING 2012 Organizing Committee.
- Miles Efron. 2011. Information search and retrieval in microblogs. *J. Am. Soc. Inf. Sci. Technol.*, 62(6):996–1008, June.
- Güneş Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.
- Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11*, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Stephan Gouws, Dirk Hovy, and Donald Metzler. 2011. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First Workshop on Unsupervised Learning in NLP, EMNLP '11*, pages 82–90, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bo Han and Timothy Baldwin. 2011. Lexical normalization of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11*, pages 368–378, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Kazi Saidul Hasan and Vincent Ng. 2010. V.: Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In *In: COLING*, pages 365–373.
- Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.
- Laura Jelh, Felix Hiebel, and Stefan Riezler. 2012. Twitter translation using translation-based cross-lingual retrieval. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Montréal, Canada, June. Association for Computational Linguistics.

- Zhenhui Li, Ding Zhou, Yun-Fang Juan, and Jiawei Han. 2010. Keyword extraction for social snippets. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 1143–1144, New York, NY, USA. ACM.
- Wang Ling, Guang Xiang, Chris Dyer, Alan Black, and Isabel Trancoso. 2013. Microblogs as parallel corpora. In *Proceedings of the 51st Annual Meeting on Association for Computational Linguistics, ACL '13*. Association for Computational Linguistics.
- Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.
- Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multisource Multilingual Information Extraction and Summarization, MMIES '08*, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Rui Liu and Eric Nyberg. 2013. A phased ranking model for question answering. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM '13*, pages 79–88, New York, NY, USA. ACM.
- Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Xiaohua Liu, Furu Wei, Shaodian Zhang, and Ming Zhou. 2013. Named entity recognition for tweets. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):3.
- Luís Marujo, Miguel Bugalho, João P. Neto, Anatole Gershman, and Jaime Carbonell. 2011a. Hourly traffic prediction of news stories. In *Proceedings of the 3rd International Workshop on Context-Aware Recommender Systems held as part of the 5th ACM RecSys Conference*, October.
- Luís Marujo, Márcio Viveiros, and João P. Neto. 2011b. Keyphrase Cloud Generation of Broadcast News. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011)*. ISCA, September.
- Luís Marujo, Anatole Gershman, Jaime Carbonell, Robert Frederking, and João P. Neto. 2012. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*.
- Luís Marujo, Ricardo Ribeiro, David Martins de Matos, João Paulo Neto, Anatole Gershman, and Jaime G. Carbonell. 2013. Key phrase extraction of lightly filtered broadcast news. In *Proceedings of the 15th International Conference on Text, Speech and Dialogue (TSD)*.
- Olena Medelyan, Vye Perrone, and Ian H. Witten. 2010. Subject metadata support powered by maui. In Jane Hunter, Carl Lagoze, C. Lee Giles, and Yuan-Fang Li, editors, *JCDL*, pages 407–408. ACM.
- Rada Mihalcea and Paul Tarau. 2004. Texttrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
- Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390.
- Arzucan Özgür, Levent Özgür, and Tunga Güngör. 2005. Text categorization with class-based and corpus-based keyword selection. In *Proceedings of the 20th International Conference on Computer and Information Sciences, ISCIS'05*, pages 606–615, Berlin, Heidelberg. Springer-Verlag.
- Alok Ranjan Pal, Projjwal Kumar Maiti, and Diganta Saha. 2013. An approach to automatic text summarization using simplified lesk algorithm and wordnet. *International Journal of Control Theory & Computer Modeling*, 3.
- Ellen Riloff and Wendy Lehnert. 1994. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems (TOIS)*, 12(3):296–333, July.
- Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.
- G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620.
- Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

- Chen Wang and Sujian Li. 2011. Corankbayes: Bayesian learning to rank under the co-training framework and its application in keyphrase extraction. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, pages 2241–2244, New York, NY, USA. ACM.
- Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: practical automatic keyphrase extraction. In *Proceedings of the 4<sup>th</sup> ACM conference on Digital libraries, DL '99*, pages 254–255, New York, NY, USA. ACM.
- Wei Wu, Bin Zhang, and Mari Ostendorf. 2010. Automatic generation of personalized annotation tags for twitter users. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 689–692, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zi Yang and Eric Nyberg. 2015. Leveraging procedural knowledge base for task-oriented search. In *Proceedings of the 38th international ACM SIGIR conference on Research & development in information retrieval*. ACM.
- Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11*, pages 379–388, Stroudsburg, PA, USA. Association for Computational Linguistics.