Using Human Attention to Extract Keyphrase from Microblog Post

Yingyi Zhang Chengzhi Zhang*
Nanjing University of Science and Technology
{yingyizhang, zhangcz}@njust.edu.cn

Abstract

This paper studies automatic keyphrase extraction on social media. Previous works have achieved promising results on it, but they neglect human reading behavior during keyphrase annotating. The human attention is a crucial element of human reading behavior. It reveals the relevance of words to the main topics of the target text. Thus, this paper aims to integrate human attention into keyphrase extraction models. First, human attention is represented by the reading duration estimated from eye-tracking corpus. Then, we merge human attention with neural network models by an attention mechanism. In addition, we also integrate human attention into unsupervised models. To the best of our knowledge, we are the first to utilize human attention on keyphrase extraction tasks. The experimental results show that our models have significant improvements on two Twitter datasets.

1 Introduction

Rapidly growth of user-generated content on social media has far outpaced human beings' reading and understanding capacity. Keyphrase extraction is one of the technologies that can organize this massive content. A keyphrase consists of one or more salient words, which represents the main topics of a document. It has a series of downstream applications, e.g., text summarization (Zhao et al., 2011a) and information retrieval (Choi et al., 2012).

Generally, corpus with human annotated keyphrases are needed to train models in supervised keyphrase extraction frameworks. The premise for annotators to annotate keyphrases is to read the corresponding content. Intuitively, features estimated from human reading behavior can be leveraged to assist keyphrase extraction. Previous studies on keyphrase extraction have ignored these features (Zhang et al., 2016, 2018). Thus, this paper aims to integrate the reading behavior into keyphrase extraction frameworks.

When human reading, they do not pay the same attention to all words (Carpenter and Just, 1983). The reading time of per-word is the indicative of textual (as well as lexical, syntactic and semantic) processing (Demberg and Keller, 2008), which reflects human attention on various content. To obtain human attention during reading, this paper estimates eye fixation duration from eye-tracking corpus inspired by Carpenter and Just (1983) and Barrett et al. (2018). The modern-day eye tracking equipment resulting in a very rich and detailed dataset (Cop et al., 2017). Thus, we utilize open-source eye-tracking corpora and do not require eye-tracking information of the target datasets.

To integrate human attention into keyphrase extraction models, this paper constructs a neural network model with attention mechanism. Attention mechanism is a neural module designed to imitate human visual attention when they reading and looking (Bahdanau et al., 2014). To regularize the predicted value of attention mechanism, human attention estimated from eye-tracking corpus is leveraged as the ground truth of it. Quantitative and qualitative analyses demonstrate that our models yield a better performance than state-of-the-art models. In addition, we prove that human attention is also effective on unsupervised keyphrase extraction models. We are, to the best of our knowledge, the first to integrate human attention into keyphrase extraction tasks.

2 Related Work

Recently, keyphrase extraction technologies have been extended to social media (Zhao et al., 2011b; Bellaachia and Al-Dhelaan, 2012), e.g.,

*Corresponding Author.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5867–5872
Florence, Italy, July 28 - August 2, 2019.

Twitter and Sina Weibo. Previous studies extract keyphrases using traditional supervised algorithms (Marujo et al., 2015), which depending on a large set of manually selected features. To overcome this drawback, neural network models, which can learn features from training corpus automatically, are proposed and are proven effective in keyphrase extraction. For instance, Zhang et al. (2016) propose a neural network model to extract keyphrases from Tweets. This model extracts keyphrases from Tweets directly, which suffers from the severe data sparsity problem. External knowledge is utilized to alleviate this problem. Zhang et al. (2018) encode conversation context consisting of Tweet reply in neural models. This model yields a better performance than Zhang et al. (2016) , which prove the effectiveness of external knowledge. Thus, this paper is in the line of integrating external knowledge into neural network models. In this paper, we explore the idea of using human attention estimated from available eye-tracking corpus to assist keyphrase extraction.

The open source eye-tracking corpus of natural reading include the Dundee corpus (Ekbal et al., 2007) and GECO (Cop et al., 2017). The features of eye tracking corpus include first fixation duration (FFD), total reading time (TRT), go-past time (GPT) , et al. TRT is a feature that has been applied to various natural language processing tasks, such as multi word expressions prediction (Rohanian et al., 2017) and sentiment analysis (Barrett et al., 2018). Thus, we select the TRT feature to represent the human attention. Since the GECO corpus is open sourced and is in English, we estimate the TRT feature from it.

3 Keyphrase Extraction Framework

Formally, given a target microblog post x_i formulated as word sequence $< x_{i,1}, x_{i,2}, \cdots, x_{i,|x_i|} >$, where $|x_i|$ denotes the length of x_i, we aim to produce a tag sequence $< y_{i,1}, y_{i,2}, \cdots, y_{i,|x_i|} >$, where $y_{i,w}$ indicates whether $x_{i,w}$ is part of a keyphrase. As shown in Figure 1, our models use the character-level word embedding proposed by Jebbara and Cimiano (2017), but we ignore this part of our architecture in the equations below:

$$y_{i,w} = \sigma(W_y tanh(W_{\widetilde{y}} h_{i,w} + b_{\widetilde{y}}) + b_y) \quad (1)$$

where $h_{i,w}$ is the representation of $x_{i,w}$ after passing through the Bi-directional LSTM (BiLSTM) layer, W_y and b_y are parameters of the function

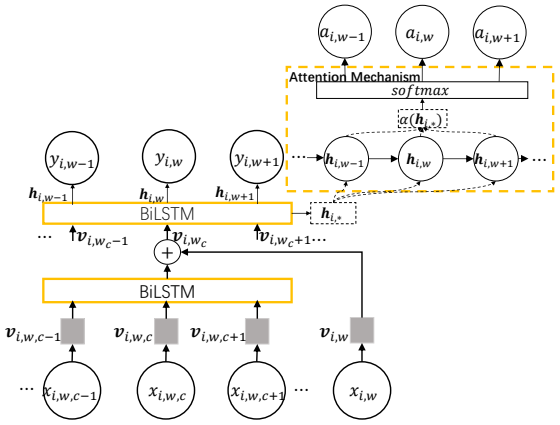

Figure 1: The framework of neural network keyphrase extraction with human attention.

$\sigma(\cdot)$ to be learned. $W_{\widetilde{y}}$ and $b_{\widetilde{y}}$ are parameters of the function $tanh(\cdot)$ to be learned, $\sigma(\cdot)$ is a nonliner function. In detail, $y_{i,w}$ has five possible values following Zhang et al. (2016):

$$y \epsilon \{Single, Begin, Middle, End, Not\} \quad (2)$$

where Single represents that $x_{i,w}$ is a one-word keyword. Begin, Middle and End represent that $x_{i,w}$ is the first word, the middle word and the last word of a keyphrase, respectively. Not represents that $x_{i,w}$ is not a keyword or part of a keyphrase.

From the hidden states, we directly predict word level raw attention scores $a_{i,w}$:

$$a_{i,w} = W_a e_{i,w} + b_a \quad (3)$$

$$e_{i,w} = tanh(W_e h_{i,w} + b_e) \quad (4)$$

where W_e and b_e are parameters of function $tanh(\cdot)$. Then, we normalize these predictions to attention weights $\widetilde{a_{i,w}}$:

$$\widetilde{a_{i,w}} = \frac{a_{i,w}}{\sum_k a_{i,k}} \quad (5)$$

where k is the length of x_i. Inspired by Barrett et al. (2018), we combine above mentioned two objections: word-level and attention-level. The word-level is to minimize the squared error between outputs $y_{i,w}$ and true word labels $\hat{y}_{i,w}$.

$$L_{word} = \sum_i \sum_w (y_{i,w} - \hat{y}_{i,w})^2 \quad (6)$$

The attention-level objective, similarly, is to minimize the squared error between the attention

weights $a_{i,w}$ and real human attention $\hat{a}_{i,w}$ estimated from eye-tracking corpus.

$$L_{att} = \sum_{i} \sum_{w} (a_{i,w} - \hat{a}_{i,w})^2 \quad (7)$$

When combined, λ_{word} and λ_{att} (between 0 and 1) are utilized to trade off loss functions at the word-level and attention-level, respectively.

$$L = \lambda_{word} L_{word} + \lambda_{att} L_{att} \quad (8)$$

In addition to above mentioned single layer models, we also use joint-layer BiLSTM proposed by Zhang et al. (2016). As a multi-task learner, joint-layer BiLSTM tackles two tasks with two types of outputs, $y^1_{i,w}$ and $y^2_{i,w}$. $y^1_{i,w}$ has a binary tagset, which indicates whether the word $x_{i,w}$ is part of a keyphrase or not. $y^2_{i,w}$ employs the 5-value tagset defined in Equation 2. There is an attention module upon each BiLSTM layer with a corresponding prediction. The loss changes with the number of layers in models. The out represents the number of layers in the model.

$$L = \sum_{i=1}^{out} \lambda^i_{word} L^i_{word} + \sum_{i=1}^{out} \lambda^i_{att} L^i_{att} \quad (9)$$

4 Experiment Settings

4.1 Twitter Dataset

Our experiments are conducted on two datasets, i.e., Daily-Life dataset and Election-Trec dataset.

Daily-Life This is collected from January of 2018 to April of 2018 using Twitter's steaming API with a set of daily life keywords.

Election-Trec This is constructed based on opensource dataset TREC2011 track[1] and Election corpus (Zeng et al., 2018)[2].

For keyphrase annotation, we follow Zhang et al. (2016) to use microblog hashtags as gold-standard keyphrases and filtered all microblog posts by two rules: first, there is only one hash tag per post; second, the hashtag is inside a post. Then, we removed all the '#' before keyphrase extraction. For both Twitter datasets, we randomly sample 0.8, 0.1 and 0.1 for training, development and testing. We preprocessed both Twitter datasets with Twitter NLP tool[3] for tokenization. After filtering and preprocessing, Daily-Life dataset and Election-Trec dataset contains 16,047 Tweets and 30,264 Tweets, respectively. Table 1 shows the statistic information of two Twitter datasets

Dataset	# of annot. msgs	mesgs length	Vocab	Cover
Election-Trec				
Train	24,210	19.94	36,018	7.7
Vali	3,027	20.00	9,909	17.8
Test	3,027	19.71	9,973	17.9
Daily-Life				
Train	12,827	28.92	40,628	7.0
Vali	1,610	28.77	9,964	17.4
Test	1,610	29.75	10,355	17.5

Table 1: Statistics of two datasets. Train, Dev, and Test denotes training, development, and test set, respectively. # of annot. Msgs: number of target post with keyphrase annotation. mesgs length: average count of words in the target post. Vocab: vocabulary size. Cover: The percent (%) of words existing in GECO.

Since there are no spaces between words in hashtags, we use some strategies to segment hashtags. There are two kinds of hashtags in the datasets. One is the 'multi-word' that contains both capitals and lowercases, the other are the 'single-word' in all lowercases or capitals. If a hashtag is a 'multi-word', we segment hashtags with two patterns, first is $(capital) * (lowercase)+$, which represents one capital followed by one or more lowercases, second is $(capital)+$, which represents one or more capitals. When doing hashtag segmentation, the first pattern is utilized firstly and then the second pattern is applied. Meanwhile, we do not do any preprocessing if a hashtag is a 'single-word'.

4.2 Eye-tracking Corpus

This paper estimates human attention from GECO corpus (Cop et al., 2017), which is based on normal reading. In GECO, participants read a part of the novel 'The Mysterious Affair at Styles' by Agatha Christie. Six males and seven females whose native language is English participated in and read a total of 5,031 sentences. There are various features in GECO, including First Fixation Duration (FFD) and Total Reading Time (TRT). In this paper, we merely use the TRT feature, which represents total human attention on words during reading. This feature is also used by Carpenter and Just (1983) and Barrett et al. (2018). We then di-

[1] https://trec.nist.gov/data/tweets/

[2] http://www.ccs.neu.edu/home/luwang/datasets/microblog_conversation.zip

[3] http://www.cs.cmu.edu/ ark/TweetNLP/

vide TRT values by the number of participants to get an average TRT (ATRT).

Human attention correlates with word frequency (Rayner and Duffy, 1988). Thus, ATRT is normalized by the word frequency of the British National Corpus (BNC)[4]. Before normalizing, BNC is log-transformed per million and inversed (INV-BNC), such that rare words get a high value. ATRT and INV-BNC are min-max-normalized to a value in the range 0-1. ATRT is multiplied with INV-BNC to get normalized ATRT (N-ATRT). After preprocessing, there are 5,012 unique words in the dataset. In addition, words that are not included in the GECO corpus, which do not have a corresponding N-ATRT value, are given the mean value of N-ATRT. Table 1 shows the percentage of words that can be found in GECO corpus.

4.3 Implementation Details

In the training phrase, we choose BiLSTM (Graves and Schmidhuber, 2005) with 300 dimensions. For single layer models, λ_{word} and λ_{att} are set to 0.7 and 0.3, respectively. For joint layer models, λ^1_{word}, λ^1_{att}, λ^2_{word} and λ^2_{att} are set to 0.4, 0.2, 0.2 and 0.2, respectively. Parameters are set under the best performance. The epoch is set to 5. We initialize target post by embeddings pre-trained on 99M tweets with 27B tokens and 4.6M words in the vocabulary.

4.4 Baseline Models

We compare our models with CRF (Zhang et al., 2008) and two kinds of neural network models: one kind is the neural network model without attention mechanism (BiLSTM model), the other is the neural network model with attention mechanism but is not modified by human attention (A-BiLSTM model). Similar as HA-BiLSTM proposed by this paper, BiLSTM models and A-BiLSTM models employ the single layer pattern and the joint layer pattern. The parameter setting of the joint layer pattern is same with Zhang et al. (2016). We compare the performance of models with the P, R and F1 evaluation metrics.

BiLSTM model This model is merely constructed by the character-level word embedding and the BiLSTM layer.

A-BiLSTM model This model is constructed by the character-level word embedding, BiLSTM layer and attention mechanism. Different with HA-BiLSTM, the attention mechanism in A-BiLSTM is not modified by human attention.

[4]http://www.natcorp.ox.ac.uk/

	Daily-Life	Election-Trec
Baseline		
CRF	64.07	58.34
BiLSTM(Single)	70.37±1.30	66.42±0.97
A-BiLSTM(Single)	70.49±0.50	66.70±0.81
BiLSTM(Joint)	72.48±0.47	67.74±0.47
A-BiLSTM(Joint)	73.23±1.06	69.69±0.37
Our model		
HA-BiLSTM(Single)	71.28±0.33	67.57±0.28
HA-BiLSTM(Joint)	**74.35**±0.17	**70.74**±0.38

Table 2: Comparisons of the average F1 scores (%) and their standard deviations (%) over the results of models on two datasets with 5 sets of parameters for random initialization. BiLSTM (Single) is the BiLSTM model with a single layer pattern. BiLSTM (Joint) is the BiLSTM model with a joint layer model. A-BiLSTM (Single) is the A-BiLSTM model with a single layer pattern. A-BiLSTM (Joint) is the A-BiLSTM model with a joint layer pattern. HA-BiLSTM (Single) is the HA-BiLSTM model with a single layer pattern. HA-BiLSTM (Joint) is the HA-BiLSTM model with a joint layer pattern.

5 Result

5.1 Overall Comparisons

Human attention estimated from eye-tracking corpus is helpful in improving the performance of neural network keyphrase extraction. As shown in Table 2, all the F1 values of models with human attention are higher than those of baseline models. In this paper, human attention is represented by the total reading time of per-word estimated from eye-tracking corpus. Thus, it indicates that the attempt of integrating human reading behavior information into neural network is feasible.

The open-source eye-tracking corpus can improve the performance of models on datasets in different genres. Although the genre of the GECO eye-tracking corpus is fiction, which is different with the genre of the target dataset (Microblog), it has the ability to improve the performance of keyphrase extraction on target datasets.

5.2 Qualitative Analysis

To qualitatively analyze why models with human attention generally perform better in comparison, we conduct a case study on two simple instances in Table 3 and Table 4. In Table 3, the keyphrase of the target post should be 'hillary clinton'. We compare the keyphrase produced by A-BiLSTM

Target Post	*what would a **hillary clinton** supreme court look like?*
Gold-standard	*hillary clinton*
Models	
A-BiLSTM (Single)	*hillary clinton; court*
HA-BiLSTM (Single)	*hillary clinton*

Table 3: The example that the hashtag in the target post is 'hillary clinton'.

Target Post	*I nominate MEN for a shorty award in **entertainment** because she never fails to write awesome smileys! xd URL*
Gold-standard	*entertainment*
Models	
A-BiLSTM (Single)	*NULL*
HA-BiLSTM (Single)	*entertainment*

Table 4: The example that the hashtag in the target post is 'entertainment'.

(Single) and HA-BiLSTM (Single). Interestingly, the A-BiLSTM extracts two phrases 'hillary clinton' and 'court'. It may due to that the attention weight of 'court' is the biggest among all words in the target post in A-BiLSTM. The HA-BiLSTM identifies the correct keyphrase. In this model, the attention weight of 'court' is the 6th biggest among all words in the target post. The reason of this phenomenon is that the 'court' has a low N-ATRT value (0.024). Using the N-ATRT value of 'court' can modify the attention weight of 'court'.

In Table 4, the keyphrase of the target post should be 'entertainment'. As shown in Table 4, the A-BiLSTM model do not extract any phrase, while the HA-BiLSTM model extract the correct keyphrase. It may due to that the attention weight of 'entertainment' in A-BiLSTM is the 13th biggest among all the words in the target post, while it is the third biggest in HA-BiLSTM, which is due to the high N-ATRT value (0.147) of 'entertainment' in GECO eye-tracking dataset modifying the corresponding attention weight.

5.3 Analysis on Unsupervised Models

In this section, we explore the idea of using human attention on TextRank (Mihalcea and Tarau, 2004), which is an unsupervised keyphrase extraction algorithm. As defined in Section 3, a Tweet x_i consist of words $x_{i,1}, x_{i,2}, \cdots, x_{i,n}$. If $x_{i,m}$ is appeared within the window of $x_{i,j}$, there is an edge $e(x_{i,m}, x_{i,j})$ between these two words. Based on the graph composited by word vertices and edges, the importance of each word vertices can be calculated. In TextRank, the value of $x_{i,j}$ and $e(x_{i,m}, x_{i,j})$ are initialized unprivileged.

Num	Daily-Life			Election-Trec		
	P	R	F1	P	R	F1
TextRank						
2	1.7	3.5	2.3	4.0	8.0	5.4
5	2.8	8.6	4.3	4.6	15.3	7.1
10	2.9	8.6	4.3	4.7	15.8	7.2
HATR						
2	2.7	5.5	3.6	6.4	12.9	8.6
5	**4.0**	**12.1**	**6.0**	7.3	24.4	11.3
10	**4.0**	**12.1**	**6.0**	**7.4**	**24.9**	**11.4**

Table 5: The P, R, F1 scores (%) of TextRank and TextRank with human attention (HATR) models on two datasets. Num represents the number of top-Num phrases that are chose to be candidate words.

In our models, we utilize human attention to normalize the initialized value of $x_{i,j}$ and $e(x_{i,m}, x_{i,j})$. The initialized value of $x_{i,j}$ depends on the N-ATRT value of itself. The initialized value of $e(x_{i,m}, x_{i,j})$ depends on the N-ATRT value of $x_{i,m}$ and $x_{i,j}$. After extracting candidate words by HATR, we generate keyphrases by combining candidate words if words are connected together in target posts.

As shown in Table 5, *all the P, R and F1 values of HATR are higher than those of TextRank.* These observations indicate that integrating human attention during reading into TextRank is feasible. Moreover, more candidate keyphrases yield better keyphrase extraction performance.

6 Conclusion

In this paper, we consolidate the neural network keyphrase extraction algorithm with human attention represented by total reading time (TRT) estimated from GECO eye-tracking corpus. The proposed models yield a better performance on two Twitter datasets. Moreover, human attention is also effective on unsupervised models.

In the future, first, we try to utilize more eye-tracking corpus and estimate more features of reading behavior. Then, we will attempt to analyze real human reading behavior on social media and thereby explore more specific human attention features on social media.

Acknowledgments

This work is supported by Major Projects of National Social Science Fund (No. 17ZDA291).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv pre-print*, arXiv/1409.0473.

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL*, pages 302–312.

Abdelghani Bellaachia and Mohammed Al-Dhelaan. 2012. NE-Rank: A Novel Graph-Based Keyphrase Extraction in Twitter. In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence*, pages 372–379.

Patricia A Carpenter and Marcel Adam Just. 1983. What your eyes do while your mind is reading. *Eye movements in reading: Perceptual and language processes*, pages 275–307.

Jaeho Choi, W. Bruce Croft, and Jinyoung Kim. 2012. Quality Models For Microblog Retrieval. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM*, pages 1834–1838.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting geco: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior research methods*, 49(2):602–615.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Asif Ekbal, S Mondal, and Sivaji Bandyopadhyay. 2007. Pos tagging using hmm and rule-based chunking. In *Proceedings of workshop on shallow parsing in South Asian languages, SPSAL*, pages 25–28.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18(5-6):602–610.

Soufian Jebbara and Philipp Cimiano. 2017. Improving opinion-target extraction with character-level word embeddings. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP, SCLeM*, pages 159–167.

Luís Marujo, Wang Ling, Isabel Trancoso, Chris Dyer, Alan W. Black, Anatole Gershman, David Martins de Matos, João Paulo da Silva Neto, and Jaime G. Carbonell. 2015. Automatic Keyword Extraction on Twitter. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL*, pages 637–643.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL*, pages 404–411.

Keith Rayner and Susan A Duffy. 1988. On-line comprehension processes and eye movements in reading. *Reading research: Advances in theory and practice*, 6:13–66.

Omid Rohanian, Shiva Taslimipoor, Victoria Yaneva, and Le An Ha. 2017. Using gaze data to predict multiword expressions. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP*, pages 601–609.

Xingshan Zeng, Jing Li, Lu Wang, Nicholas Beauchamp, Sarah Shugars, and Kam-Fai Wong. 2018. Microblog conversation recommendation via joint modeling of topics and discourse. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 375–385.

Chengzhi Zhang, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, and Bo Wang. 2008. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 836–845.

Yingyi Zhang, Jing Li, Yan Song, and Chengzhi Zhang. 2018. Encoding conversation context for neural keyphrase extraction from microblog posts. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 1676–1686.

Hong Zhao, Chen Sheng Bai, and Song Zhu. 2011a. Automatic keyword extraction algorithm and implementation. *Applied Mechanics and Materials*, 44:4041–4049.

Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011b. Topical Keyphrase Extraction from Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL*, pages 379–388.