Keyphrases Extraction from User-Generated Contents in Healthcare Domain Using Long Short-Term Memory Networks

Ilham Fathy Saputra, Rahmad Mahendra, Alfan Farizki Wicaksono
Faculty of Computer Science, Universitas Indonesia
Depok 16424, West Java, Indonesia
ilham.fathy@ui.ac.id, {rahmad.mahendra, alfan}@cs.ui.ac.id

Abstract

We propose keyphrases extraction technique to extract important terms from the healthcare user-generated contents. We employ deep learning architecture, i.e. Long Short-Term Memory, and leverage word embeddings, medical concepts from a knowledge base, and linguistic components as our features. The proposed model achieves 61.37% F-1 score. Experimental results indicate that our proposed approach outperforms the baseline methods, i.e. RAKE and CRF, on the task of extracting keyphrases from Indonesian health forum posts.

1 Introduction

The growth of Internet access facilitates users to share and obtain contents related to healthcare topic. There has been growing interest in using Internet to find information related to healthcare concerns, including symptoms management, medication side effects, alternative treatment, and fitness plan. The tremendous amounts of user-generated health contents are available on the web pages, online forums, blogs, and social networks. These user-generated contents are actually potential sources for enriching medical-related knowledge. It is highly desirable if the knowledge contained in the user generated contents can be extracted and reused for Natural Language Processing and text mining application.

Keyphrases, which is a concise representation of document, describe important information contained in that document. The number of text processing tasks can take advantage of keyphrases, e.g. document indexing, text summarization, text classification, topic detection and tracking, information visualization.

We believe that extracting keyphrases from documents in healthcare domain can be beneficial. A medical question answering system is expected to provide concise answers in response to clinical questions. The keyphrases extracted from the question can be used to formulate a query to retrieve the answer passage from a collection of medical documents. On the other hand, the health-related web forums usually contain very large archives of forum threads and posts. To make use of those archives, it is critical to have functionality facilitating users to search previous forum contents. Keyphrases identification is an important step to tackle this kind of document retrieval problem.

Most previous works on keyphrases extraction task focused on long documents, e.g. scientific articles and web pages, while few works attempt to identify keyphrases from user-generated contents, e.g. e-mail messages (Dredze et al., 2008), chats (Kim and Baldwin, 2012; Habibi and Popescu-Belis, 2013), and tweets (Li et al., 2010; Zhao et al., 2011; Zhang et al., 2016). Extracting keyphrases from an online forum is simply not a trivial task, since the contents are written in free text format (i.e. unstructured format), and often prone to grammatical and typographical glitches.

In this paper, we address the task of keyphrases extraction from user-generated posts in online healthcare forums. We present the technique that treats keyphrase extraction as a sequence labeling task. In our experiment, we employ and combine deep learning architectures, i.e. bi-directional Long Short-Term Memory networks, to exploit high level features between neighboring word positions. To improve the quality of our model, we leverage several new hand-crafted features that can handle our keyphrase extraction problems in medical user-generated contents.

Proceedings of the BioNLP 2018 workshop, pages 28–34
Melbourne, Australia, July 19, 2018.

2 Related Work

Keyphrases are usually selected phrases or clauses that can capture the main topic of a given document (Turney, 2000). Keyphrases are expected to provide readers with highly valuable and representative information, such that looking at keyphrases is sufficient to understand the whole body of a document.

In general, keyphrases extraction methods can be categorized into unsupervised and supervised approach (Hasan and Ng, 2014). For the unsupervised line of research, keyphrases extraction can be formulated as a ranking problem, in which each candidate keyphrase is assigned the score. RAKE (Rose et al., 2010) uses ratio of word degree and frequency to rank terms. Mihalcea and Tarau (2004) studies the graph-based approach that treats words as vertices and constructs edge between words using co-occurrence.

On the other hand, supervised machine learning approach requires training data that contain a collection of documents with their labeled keywords which are often very difficult to obtain. Using this approach, keyphrase extraction is formulated as a classification or sequence labeling task in the level of words or phrases. The supervised learning approach starts from generating candidate keyphrases from a particular document. Then, each candidate is classified as either a keyphrase or non-keyphrase. The well-known method for this approach is KEA (Witten et al., 1999), which applied machine-learning (i.e. Naive Bayes) for classifying candidate keyphrases. Sarkar et al. (2010) utilizes neural network algorithm in classifying candidate phrase as a keyphrase.

Other supervised approach for keyphrases extraction is based on sequence labeling problem (Zhang, 2008; Cao et al., 2010; Zhang et al., 2016). The assumption behind this model is that the decision on whether a particular word serves as a keyword is affected by the information from its neighboring word positions. Zhang (2008) apply Conditional Random Fields (CRF) algorithm to find keyphrases from the Chinese documents. Zhang et al. (2016) proposes a joint-layer recurrent neural network model to extract keyphrases from tweets, which is an application of deep neural networks in the context of keyphrase extraction.

As far as our knowledge, there are limited works regarding the task of keyphrase extraction from user-generated contents, especially for healthcare domain. Sarkar (2013) applies hybrid statistical and knowledge-base approach to extract keyphrases from medical articles. Stopwords are used to split candidate keyphrases, then candidates were ranked based on two aspects: Phrase Frequency * Inverse Document Frequency (PF-IDF) and domain knowledge which is extracted from medical articles. Cao et al. (2010) uses CRF for extracting keywords from medical questions in online health forum. They harness information about word location and length as features in their experiments.

3 Methodology

In this work, we see the problem of keyphrase extraction as a sequence labeling, in which each word w_i is associated with a hidden label $y_i \in \{\text{keyword, non-keyword}\}$. Formally, given a medical forum posts containing N words $W = (w_1, w_2, ..., w_N)$, we want to find the best sequence of labels $Y = (y_1, y_2, ..., y_N)$, in which each label is determined using probabilities $P(y_i|w_{i-l}, ..., w_{i+l}, y_{i-l}, ..., y_{i+l})$; and l is a small integer.

3.1 Proposed Model

To cope with our problem, we employ a deep neural network-based approach specially designed for sequence labeling problem, such as Long Short-Term Memory (LSTM) Networks and its variants, to extract high-level sequential features, before they are feeded into the last layer that determines the most probable label for each word or timestep. Since we employ neural networks, we can also view our model as a function $F : R^{N \times M} \rightarrow R^{N \times 2}$:

$$[z_1, z_2, ..., z_N] = F([x_1, x_2, ..., x_N])$$

where, M is the size of input vector in each timestep, N is the number of timestep, $x_i \in R^M$ is the vector representation of word w_i, and $z_i \in R^2$ is the output vector in each timestep ($\sum_j z_{i,j} = 1$). Furthermore, the vector representation of word can be obtained using state-of-the-art technique, such as the one proposed by (Mikolov et al., 2013). Finally, y_i can be determined as follows.

$$y_i = \begin{cases} \text{keyphrase} & \text{if } z_{i,0} > z_{i,1} \\ \text{non-keyphrase} & \text{otherwise} \end{cases}$$

In our work, the operational definition of keyphrase is actually extractive, in the sense that keyphrase is explicitly extracted from a sequence of words found in the document. For example, from the following sentence: "*Doc, I have a frequent back pain. What happen?*", we can extract "*frequent back pain*" as our keyphrase.

In order to know that "*back*" and "*frequent*" are part of the keyphrase along with "*pain*", we need to consider such phrasal structure information, i.e. the word "*back*" that serves as the modifier of symptom "*pain*", as well the word "*frequent*" that informs its intensity level. Therefore, we argue that sequential-based neural networks, such as Long Short-Term Memory (LSTM) networks and their variants can better fit our problem since they can naturally leverage neighboring information.

To get a better inference process, the information from the past and future of current position in the sequence can be integrated. This approach has been proven effective in the number of sequence labeling tasks, such as semantic role labeling (Zhou and Xu, 2015) and named-entity recognition (Ma and Hovy, 2016). Based on those previous studies, we utilize a bi-directional LSTM (B-LSTM) in our work to extract structural knowledge by doing forward and backward processing in the sequence. In order to do that, we build two LSTMs with different parameters and subsequently concatenate the outputs from both LSTMs. Moreover, we build our layers for up to two layers of B-LSTM. Finally, the locally normalized distribution over output labels is computed via a softmax layer. In the other scenario, we also employ Conditional Random Fields (CRFs) (Lafferty et al., 2001) to produce label predictions in the last layer. Following Rei et al. (2016), we used Viterbi algorithm to efficiently find the sequence of labels $[y_1, y_2, ..., y_M]$ with the largest score $s(Y)$. As can be seen in the following equation, $s(Y)$ computes CRF score of a sequence, which means the likelihood of the output labels.

$$s(Y) = \sum_{t=1}^{M} A_{t,y_t} + \sum_{t=0}^{M} B_{y_t,y_{t+1}}$$

where A_{t,y_t} shows the confident score of of label y_t at timestep t, $B_{y_t,y_{t+1}}$ show the likelihood of transitioning from label y_t to label y_{t+1}. It is worth to note that all these parameters are trainable.

The spirit of deep learning is basically to automatically extract features from the input without the need of expensive feature engineering. However, this idea works well when we have a significant amount of training samples, which is not applicable in our case since the size of our data is small enough. As a result, to cope with this problem, we combine deep learning technique with several feature engineering steps. The idea is that several hand-crafted features are leveraged to help deep learning architectures understand the main characteristics of the data before they actually learn more high-level features from those hand-crafted features. Suppose, $F_{i,j}$ represents a type of feature vector extracted from an input x_i in one timestep and K is the number of feature types. A feature vector for x_i is defined as follows.

$$x_i = concatenate(F_{i,1}, F_{i,2}, ..., F_{i,K})$$

The detail of our proposed feature types is explained in the next subsection. Moreover, we also argue that each feature has different contribution to the model. Instead of concatenating all features into one vector, we try to assign weights to every feature type before we pass it to the next layer. In order to do so, we create a new layer underneath our model to do the weighting scenario. Suppose, $W_i \in \mathbb{R}^{a \times b_i}$ is a trainable weight for feature vector F_i, where a is the size of input size in each timestep and b_i is the size of feature vector F_i. The following equation presents our idea for weighting the feature vectors.

$$x_i = tanh(W_1.F_{i,1} + W_2.F_{i,2} + .. + W_K.F_{i,K})$$

3.2 Proposed Features

We perform automatic representation learning in the input layer, in which vector representation of a particular word is automatically learned. However, we argue that the end-to-end learning approach alone will have not effectively worked in our case since the tiny size of dataset. So, we leverage nine hand-crafted features that can help our model to characterize the sequence of keyphrases.

WORD EMBEDDING. We use pre-trained word embedding that fills several slots in our feature vector. A skip-gram model from Mikolov et al. (2013) was used to generate a 128-dimensional

vector of a particular word. The word embedding we used in this work is trained using documents that are collected from online forums, medical articles, and medical question answering forums. By using embedding feature, slank words and lexical variants can be naturally handled since all variants should have similar vector representation.

MEDICAL DICTIONARY. We also devise feature vector using the knowledge derived from a dictionary containing medical terminologies. Technically, we generate a one-hot feature vector for each word in the sentence by checking whether the word is listed in the dictionary.

WORD LENGTH. This feature represents the length of each word (i.e., the number of characters in every word) in the sentence. The rationale behind proposing this kind of feature is that the medical domain-specific words (e.g. "influenza", "tuberculosis") tend to be lengthy compared to general words. Cao et al. (2010) found that there is a correlation between the length of a word and its informativeness value (inverse document frequency value).

WORD POSITION. The important term most likely appears in either beginning or end of document. The first sentence of document typically contains phrasal topic, while a few last sentences usually emphasize the content discussed in the document. In a medical consultation forum, a user often starts her post with a statement explaining the problem, then gives more explanation in form of several narrative sentences. In the end, she asks one or two questions. Keyphrases are potentially extracted from a problem statement and the questions in the forum posts.

POS-TAG. Part-of-Speech category of word can be also exploited as a feature, since it may feed our model with grammatical information and a better understanding of ambiguous words. Based on our observation, many keyphrases have a common POS pattern, e.g. a verb followed by the sequence of nouns. The POS-tag feature is represented as a one-hot-vector, whose length is the number of tags.

MEDICAL ENTITY. We extract four types of medical entity from the text, i.e. *drug*, *treatment*, *symptom*, and *disease*. The medical entity often become part of a keyphrase of the sentence or document. Furthermore, this feature complements the Medical Dictionary feature. While a drug or disease name is not available in the training data or a medical dictionary, it is still possible to learn it using medical entity recognizer.

ABBREVIATION AND ACRONYM. We also identify whether a word is an abbreviation or acronym. We compile an acronym dictionary and then check whether a word in the forum post is found in the dictionary. We have observed that important words are rarely shortened by the users.

WORD CENTRALITY. The role of this feature is to rank words in a document by their importance. To extract this feature, we adapt TextRank algorithm (Mihalcea and Tarau, 2004). We build undirected graph, in which the word is represented as the vertice and the distance between words as the edge. We use word similarity score as weights for the edge. Pre-trained word embedding model is used to calculate the cosine similarity between two word vectors. In our work, two words (vertices) are adjacent (having an edge between them) if their similarity are not negative. Moreover, we use a modified PageRank algorithm (Page et al., 1998) that consider weight of edge in calculation.

Formally, let $G = (V, E)$ be an undirected graph with the set of vertices V and set of edges E, where E is a subset of $V \times V$. For a given vertex V_i, let $In(V_i)$ be the set of vertices that points to it (predecessors) and $Out(V_i)$ be the set of vertices that vertex V_i points to (successors), the modified PageRank equation proposed by (**?**) can be seen in the following formula.

$$WS(V_i) = (1-d)+d* \sum_{V_i \in In(V_i)} \frac{w_{ji}}{\sum_{V_i \in Out(V_j)}} w_{jk}$$

WORD STICKINESS. There are typical noises found in the user-generated contents, such as lack of proper punctuation usage. For example, in the sentence "*I have a pain on forehead stomachache and blurred vision*", there is no comma between the words "*forehead*" and "*stomachache*", as well between "*stomachache*" and "*and*"; while it should be required. The model may mistakenly select the sequence "*forehead stomachache*" as a single keyphrase.

To address this problem, we propose a feature that is able to capture how likely a given word is occurred together with the preceding and succeeding words. We compute Pointwise Mutual Information (PMI) of all bigrams to capture such information using documents from health-related online forum.

Table 1: Statistical data of Indonesian healthcare user-generated posts dataset

Number of posts	416
Total number of words	26,747
Number of keyphrases	1,861
Avg number of words per posts	64
Avg number of keyphrases per posts	4

The PMI formula can be seen as follows.

$$PMI(x, y) = log(\frac{P(x, y)}{P(x).P(y)})$$

where $p(x)$ is the occurrence probability of word x, $p(y)$ is the occurrence probability of word y, and $p(x, y)$ is the probability of word x and y co-occur together.

The feature function is formally described as $f_s(w) = [x, y]$, where w is a word in a particular document, x is the stickiness value between w and its preceding word, and y is the stickiness value between w and its succeeding word. For example, given the word "`cancer`" in the sentence "`How to prevent cancer doc?`", the feature value is $f_s(cancer) = [0.56, 0.1]$, where f_s is feature function for stickiness value. It is worth to note that the word "`cancer`" rarely co-occur with the word "`doc`". It is reflected that the stickiness value of the word "`cancer`" relative to the word "`doc`" is smaller than the stickiness value to the word "`prevent`".

4 Evaluation Result

4.1 Data and Resources

The data for the experiment is taken from the collection of consumer-health questions crawled through Indonesian healthcare consultation forums (Hakim et al., 2017). Due to resources limitation, we only manually annotate 416 sample of user-generated posts. The description of the dataset for experiment can be seen in Table 1

We use the dictionary from The Medical Council of Indonesia[1] to extract MEDICAL DICTIONARY feature. On the other hand, POS-Tag feature is learned by model that is trained using data from Dinakaramani et al. (2014).

[1] (www.kki.go.id/assets/data/arsip/SKDI_Perkonsil,_11_maret_13.pdf)

4.2 Experiment

There are two main scenarios for the experiment. First, feature ablation study aims to determine feature's contribution to the model performance. Second, model selection finds the model that outputs best result. The performance of a model is measured by precision, recall, and F1 metric. Precision is the number of keyphrases that are correctly extracted, divided by the total number of keyphrases labeled by our system. Recall is the number of keyphrases that are correctly extracted, divided by the total number of keyphrases in the gold-standard.

4.2.1 Feature Ablation Study

Ablation study is done by systematically removing feature sets to identify the most important features. We adopt leave one out (LOO) technique for feature ablation study. First, the model that uses all proposed features is evaluated. After that, 9 other different models are constructed, each of which uses combination of 8 features (another one feature is ablated in each model). The difference of F1-score between original model using all features and model with one missing feature indicate the contribution of (missing) feature to model performance. For ablation study, we split the data into 80% training set and 20% testing set. In this work, feature ablation is conducted using LSTM model.

Negative delta percentage score, as shown in Table 2, means that our proposed features contribute positively to improve model performance. The WORD EMBEDDING, which is most basic feature in our model, contributes the most. By removing word embedding feature, precision and recall decrease by 13.40% and 23.48% respectively. WORD STICKINESS is the second most important feature, indicated by change of 8.12% F-1 score. Based on this ablation study, WORD POSITION is not part of best feature combination.

We re-evaluate ablation study result using partial match score. In this scheme, suppose that the expected keyphrase consists of two words or more and the predicted contains only one word of it, partial match will still count it as a true positive. We find that removing WORD POSITION feature causes partial-match precision of model drops. So, our decision is to include WORD POSITION feature along with all other features in the final model.

Table 2: Feature Ablation Evaluation (in %)

Removed Features	Precision	Recall	F-1
Word Embedding	-13.40	-23.48	-18.91
Medical Dictionary	-3.44	-5.28	-4.30
Word Length	-4.92	-5.78	-5.33
Word Position	+2.91	+0.52	+1.70
POS-Tag	-5.41	-6.25	-5.80
Medical Entity	-6.15	-6.73	-6.40
Abbreviation and Acronym	-4.85	-5.99	-5.35
Word Centrality	-3.92	-5.44	-4.61
Word Stickiness	-7.35	-9.02	-8.12

Table 3: Model Evaluation (in %)

Models	Precision	Recall	F-1
RAKE	11.60	12.54	12.05
CRF	20.98	18.60	19.23
LSTMs	52.03	55.04	53.43
B-LSTM	55.12	58.07	56.48
Stacked-B-LSTMs	56.19	59.84	57.93
Stacked-B-LSTMs-CRF	59.12	60.11	59.22
Weight-Stacked-B-LSTMs	**60.06**	**63.08**	**61.37**

4.2.2 Model Selection

We test our model using 10-cross-validation scenario. As the baselines, we implement RAKE (Rose et al., 2010) and CRF (Cao et al., 2010). For CRF model, we apply the similar features with used in LSTM. The summary of various model evaluation is presented in Table 3.

RAKE performs the worst on extracting keyphrases from user-generated healthcare forum posts, since it is actually devoted for formal text. Performance of CRF model is also not good. Based on our observation from the predicted output by CRF, this method fails to predict the long sequences as the keyphrases. LSTM outperforms the baselines by achieving 53.43% F-1 score, 35% higher than CRF.

Using bidirectional concept, LSTM is able to integrate information from previous and after timestep, so that B-LSTM deliver better result compared to LSTM. Stacked-B-LSTMs using two layers performs better than B-LSTM for this task. Moreover, the weighting layer, which learns the weight for each feature, improves model performance. Hence, the best result was obtained by Weight-Stacked-B-LSTMs, whose Precision, Recall, and F-1 are respectively 60.06%, 63.06%, and 61.37%. It indicates that the feature weighting process worked well and, on some degree, demonstrate the reliability of our model in keyphrases extraction for user-generated contents in healthcare domain.

5 Conclusion

We proposed the model to address the task of keyphrases extraction from user-generated contents in medical domain. Extracting information about health-related concerns from user-generated forum post is not a trivial task, due to the fact that the content is usually short and written in an unstructured format, as opposed to formal text. Our model is based on sequence labeling task that employs deep learning approach using Long Short-Term Memory networks. Furthermore, several handcrafted features are proposed, including word centrality to detect important word in a document and word stickiness to obtain complete sequence of words as a keyphrase. We also propose a new layer in the neural network architecture for weighting the features. Our model successfully outperforms baseline methods for keyphrase extraction.

References

Yong-gang Cao, James J Cimino, John Ely, and Hong Yu. 2010. Automatically extracting information needs from complex clinical questions. *Journal of biomedical informatics*, 43(6):962–971.

Arawinda Dinakaramani, Fam Rashel, Andry Luthfi, and Ruli Manurung. 2014. Designing an indonesian part of speech tagset and manually tagged indonesian corpus. In *IALP*, pages 66–69.

Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. 2008. Generating summary keywords for emails using topics. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 199–206, New York, NY, USA. ACM.

Maryam Habibi and Andrei Popescu-Belis. 2013. Diverse keyword extraction from conversations. In *Proceedings of the ACL 2013 (51th Annual Meeting of the Association for Computational Linguistics), Short Papers*, pages 651–657. ACL.

Abid Nurul Hakim, Rahmad Mahendra, Mirna Adriani, and Adrianus Saga Ekakristi. 2017. Corpus development for indonesian consumer-health question answering system. In *Proceedings of the 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 221–226.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273, Baltimore, Maryland. Association for Computational Linguistics.

Su Nam Kim and Timothy Baldwin. 2012. Extracting keywords from multi-party live chats. In *Proceedings of the 26th Pacific Asia Conference on Language, Information and Computation, PACLIC 26, Bali, Indonesia, December 16-18, 2012*, pages 199–208.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1064–1074. ACM.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*.

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.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia.

Marek Rei, Gamal K. O. Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. *CoRR*, abs/1611.04361.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20.

Kamal Sarkar. 2013. A hybrid approach to extract keyphrases from medical documents. *International Journal of Computer Applications*, 63:14–19.

Kamal Sarkar, Mita Nasipuri, and Suranjan Ghose. 2010. A new approach to keyphrase extraction using neural networks. *arXiv preprint arXiv:1004.3274*.

Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information retrieval*, 2(4):303–336.

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 fourth ACM conference on Digital libraries*, pages 254–255. ACM.

Chengzhi Zhang. 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 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845.

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.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *ACL (1)*, pages 1127–1137.