Keyphrase Extraction from User Generated Contents on Medical Domain

ABSTRACT

The tremendous amount of user-generated contents, such as the questions about symptoms and diseases being asked by a user in online forums, or blog posts about dietary and fitness, are potential sources for enriching knowledge in medical document retrieval and health-related text mining applications. Understanding those various contents is simply not a trivial task. Therefore, reducing the complex characteristics of the contents into structured information could be very beneficial for the text interpretation process. We argue that there are many missing pieces of information when we rely merely on medical entities (i.e., using Medical Entity Recognizer) as our main information. In our work, we use keyphrases extraction techniques to extract more important information about the concept of the document. Previous works on keyphrases extraction mostly focused on a general domain, long and formal written documents, which make them perform poorly when being applied to user-generated contents and medical documents. Moreover, we propose several deep learning architectures, such as bi-directional long-short term memory networks, convolutional neural networks, and their combinations as the models for extracting keyphrases. Furthermore, we leverage word embeddings, medical concepts from a knowledge base, and linguistic components as our features. The proposed algorithm achieved F1 score of 82.41%. This result outperforms state of the art system for extracting keyphrases using a statistical model.

KEYWORDS

keyphrase extraction, health-related text mining, deep learning, recurrent neural networks, convolutional neural networks

ACM Reference format:
. 2017. Keyphrase Extraction from User Generated Contents on Medical Domain. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 4 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The growth of internet access facilitates users to share and obtain contents about healthcare topic. These user-generated contents are actually potential sources for enriching medical-related knowledge, for which it can be very valuable for several tasks in IR and NLP.

However, understanding those various health-related contents, especially from an online forum, is simply not a trivial task since they are written in free text format (i.e. unstructured format), and often prone to grammatical and typographical glitches. Therefore, reducing the complex characteristics of the textual contents, including but not limited to linguistic variations, into structured information could be very beneficial for the text interpretation process.

One possible solution is harnessing medical entity recognizer to extract medical terms that represent the document they belong. However, we argue that there are many missing pieces of information when we merely rely on medical entities or terminologies as our main information for representing a particular document. Hence, we leverage keyphrase to cover more valuable information. Keyphrases are usually selected phrases or clauses that can capture the main topic of a given document [13]. They can provide readers with highly valuable and representative information, such that looking at keyphrases is sufficient to understand the whole body of a document.

In this paper, we only focus the task of keyphrase extraction on Indonesian medical questions from online forums, such as alodokter[1], health detik [2], dokter.id [3], klikdokter [4]. We present a new keyphrase extraction techniques to extract more important information about the patient intentions and needs, which expressed inside Indonesian medical questions. Furthermore, we also propose a new definition of keyphrases, which is based on our observation of health-related and user-generated questions. Actually, our final goal is to develop medical question answering systems to assist physicians or people obtaining health information. The work presented in this paper is absolutely in line with our goal since important keyphrases of a medical question can be used to formulate a query for the passage or question retrieval subsystem [3]. The following example shows the results of keyphrase extraction in a medical user-generated contents:
Dok , saya sering sekali merasa **kram pada perut kanan bagian atas** *disertai* ***detak jantung yang cepat*** *serta* ***keringat dingin*** *dan* ***pusing****. Dan saat kambuh seperti itu,* ***nafas sangat sakit****. apakah ada hubungannya dengan* ***diet mayo****? terima kasih sblumnya dok.* `(Doc, i frequently feel a` **`cramp on upper right stomach`** `with` **`fasten heart beat`** `also` **`cold sweat`** `and` **`headache`**`. And when the symptom occur it's` **`very hurt when breathing`**`. is there any connection with` **`mayo diet`**`? thanks before)`

It can be seen on the example that keyphrases can be a word, phrase, or even clause. Furthermore, there are many grammatical errors and abbreviation, such as `"sy"` for the replacement of the word `"saya (I)"`. It is also worth to note that keyphrase `"diet mayo (mayo diet)"` is not part of any medical entity categories. Thus, we argue that extracting keyphrase from user-generated content especially on medical domain is not a trivial task.

Conference'17, July 2017, Washington, DC, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] www.alodokter.com
[2] health.detik.com
[3] www.dokter.id
[4] www.klikdokter.com

In our work, we use similar approach as in Cao et al. [1], which treats keyphrase extraction as a sequence labeling task. The difference is that, in our experiment, we employed and combined various deep learning architectures, such as convolutional neural networks and 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 handcrafted features that can handle our keyphrase extraction problems in medical user-generated content, such as word importance and word stickiness features. The word importance feature is enable to rank words in a document by their importance value. The more important a word is to the content of document, the higher its "keywordness" value would seem to be. In addition, we also propose word stickiness feature to make our model constructs keyphrases

2 RELATED WORKS

In general, keyphrase extraction methods can be divided into two groups: unsupervised ranking (statistical) approach and supervised machine learning approach. For the unsupervised line of research, keyphrase extraction is formulated as a ranking problem, in which each candidate keyphrase is assigned a score that represents its keywordness value. The well-known research in unsupervised approach are RAKE [9] which use the ratio of word degree and frequency to rank terms. There is also Mihalcea and Tarau [6] who developed a graph-based approach which treats words as vertices and constructs edge between words using co-occurrence.

On the other hand, supervised machine learning approach requires training data that contains a collection of documents with their labeled keywords which is often very difficult to obtain. Using this approach, keyphrase extraction is usually treated as a classification or sequence labeling task in the level of words or phrases. The first step of this approach generates candidate keyphrases from a particular document. Finally, every candidate keyphrase in the document will be classified as either a keyphrase or non-keyphrase. The well-known method for this approach is KEA [14], they use machine-learning (i.e., Naive Bayes) for classifying candidate keyphrases. The utilization of neural networks in classifying candidate keyphrases also conducted by [11]. Recently, a well-known supervised approach for keyphrase extraction is based on sequence labeling problem ([15], [1], [16]). 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.

Sarkar and Kamal [10] use a hybrid medical knowledge base and statistical approach to extract keyphrases from medical articles. Stopwords was used by them to split candidate keyphrases, then candidates are ranked with two aspects: PF-IDF (Phrase Frequency * Inverse Document Frequency) and domain knowledge which is extracted from medical article.

As far as our knowledge, there are limited works regarding the task of keyphrase extraction from user-generated contents, especially in medical question. Cao et al. [1] use Conditional Random Fields (CRFs) for extracting keywords from medical questions in online health forum. They use information, such as word location and length as features in their experiments.

3 METHODOLOGY

In this work, we view keyphrase extraction problem as a sequence labeling task. That is, given a medical question 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 $P(y_i | w_{i-l}, ..., w_{i+l}, y_{i-l}, ..., y_{i+l})$ where l is a small number.

Because we have a small dataset, our model would have a small learning material and an end-to-end learning approach will not work well. As a result, we leveraged nine handcrafted features that can help our model to characterize the sequence of keyphrases.

Word Embedding (We). In the experiments, we use word embedding as input to the neural network. A skip-gram model from Mikolov [7] research was used to generate these 128-dimensional vectors for documents in evaluation data. The word embedding we used in this work were trained using a document that we gathered from Indonesia online forum, medical article, and medical question answering forum. With this feature, slank words can be handled better.

Medical Dictionary (Md). In our experiment, we leverage a medical knowledge base feature using a medical dictionary. The dictionary was built by combining disease, symptom, treatment, and drug dictionaries from Konsil Kedokteran Indonesia[5]. The rationale behind this feature is intentions and needs of patients is more focused on their disease, symptom, treatment, and drug. Based on this feature, we classified words by their appearance in the medical dictionary. We represent this feature as one-hot-vector.

Word Length (Wl). This feature represents the length of each word (i.e., the number of characters in every word). This feature is based on Cao et al. [1] research on keyphrase extraction in medical question. This feature is added because domain-specific words (e.g. "tuberculosis") tend to be lengthy when compared to common English words, and there is a correlation between the length of a word and its IDF value [1].

Word Position (Wp). We also adapted word position as feature based on Cao et al. [1] research. Based on their observation, an important term sometimes appears toward the end of a clinical question. For example, "medicine for bell's palsy" appear towards the end of a question "what is the best medicine for bell's palsy?".

POS-tag (Pt). POS-tag of words was also added as our feature. POS-tag may give model grammatical information and a better understanding of ambiguous words. By our observation, many keyphrases have a common POS pattern. We use Stanford Part-Of-Speech Tagger and a model which pre-trained by Dinakaramani [2]. We represent our POS tag feature as one-hot-vector.

Medical Entity (Me). We use medical entities which was annotated manually using four categories: drug, treatment, symptom, disease. By our observation, medical entity are sometimes part of a keyphrase. Furthermore, medical entities can provide more information about drug or disease which not available in training data or a medical dictionary. We also use one-hot-vector to represent this feature.

Abbreviation and Acronym (Wa). We also identified words by their appearance in abbreviation or acronym dictionary which was gathered manually. We observed that important word is rarely shortened by the users, such as `""cancer", "flu", "bell's`

[5] www. kki.go.id/assets/data/arsip/SKDI_Perkonsil,_11_maret_13.pdf.

palsy". One-hot-vector is used to classify whether a word is an abbreviation/acronym or not.

Word Importance (Wi). The purpose of this feature is to rank words in a document by their importance. To extract this feature, we adapted TextRank [6] algorithm for ranking words in a document. They represent words as vertices in the graph and the distance between words as edges. However, in building an undirected graph, we use a word similarity as a weight for edges by using our pre-trained word embedding model to calculate cosine similarity between two word vectors. Two words have a connected edge if their similarity is not negative. Moreover, we use a modified PageRank [8] algorithm that consider edges weight in the 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 point to it (predecessors) and let $Out(V_i)$ be the set of vertices that vertex V_i points to (successors). The modified PageRank formula that proposed by [6] can be seen in Formula 1.

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

Word Stickiness (Ws). As a keyphrase, a sequence must be in a valid order. There is some noise such as misused punctuation by users. For example, "*Saya mengalami pusing pada jidat sakit perut dan pandangan kabur* (I am having a headache on forehead stomachache and blurred vision)", there is no comma between "jidat (forehead)", "sakit perut (stomachache)" and "dan (and)". Our model may mistaken the sequence "jidat sakit perut (forehead stomachache)" as a keyphrase. Based on that problem, we propose a feature that may capture how likely of a given word is occurred together with the word before and after. We leveraged a language model which use Pointwise Mutual Information (PMI) of a bigram probability to capture the problem. The language model was trained using documents from health-related online forum. Formula 2 is the formula for calculating the PMI value. In that formula, $p(x)$ 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.

$$PMI(x, y) = log(\frac{P(x, y)}{P(x).P(y)}) \qquad (2)$$

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

3.1 Proposed Model

We process the sequences word by word. All Features are extracted and concatenated into one vector from every word as the input representation of the model. The output of our model are labels which describe whether a word is either part of keyphrase or not.

We introduce Convolutional Neural Networks (CNNs) to exploit high-level features between neighboring word positions. Based on our observation, a keyphrase is. For example, "Doc, I have a frequent back pain. What happen?". The keyphrase for the previous example is "frequent back pain". In order to know that "back" is part of keyphrase, we need information from the word "frequent" which give an intensity of a symptom and "pain" which refer to "back pain" as the main term. With CNNs, we argue that our model can capture the surrounding information of a word. The output of the CNNs layer will be fed into our next layer which is B-LSTM.

Table 1: Statistical information of dataset. W, K, $\bar{N_w}$, $\bar{N_k}$ are the total of words, number of keyphrases, average number of words, and average number of keyphrases in each question, respectively.

#questions	W	K	$\bar{N_w}$	$\bar{N_k}$
416	26747	64.76	1861	4.49

To get a better inference, information from the past and future of sequences can be integrated. This approach is proven effective in sequence labeling task such as semantic role labeling [17] and named entity recognition [5]. Therefore, we utilized bi-directional LSTM (B-LSTM) for extracting structural knowledge by processing sequences both forward and backward.Then, we concatenate the output from both LSTM. We build up to two layers of B-LSTM.

Finally, the locally normalized distribution over output labels is computed via a softmax layer.

We also experimented on customizing the way we input the features underneath our aforementioned model. Instead of concatenating all features into one vector, which can be seen in Formula 3, we tried to assign weights to every feature we have before the input fed into CNNs layer.

$$Z = concatenate(F_1, F_2, ..F_9) \qquad (3)$$

We argue that each feature has different contribution to the model. In order to do so, we create a new layer underneath our model to do the weighting scenario. Formally, let Z is the input representation and F_i are the extracted feature in vector form, $W_i \in \mathbb{R}^{\mathrm{a_i \times b}}$ where the length of a is equal to F_i and b is the input representation vector. The following equation to weight all features can be seen in below:

$$Z = tanh(W_1.F_1 + W_2.F_2 + .. + W_n.F_9) \qquad (4)$$

4 EVALUATIONS AND RESULTS

4.1 Data Collection

To develop training data, we collected question-answer pairs which previously crawled from Indonesian question-answering medical forum. Then, we removed the answer section for every pair because it is not relevant for obtaining patient intentions and needs. Moreover, we also observed our data and defined the definition of keyphrase on user-generated medical domain so that the annotation process can be consistent. Finally, we manually annotated 416 questions. The statistical information of our data can be seen in Table 1.

Table 2: Model Scenario Results

Models	Precision	Recall	F-Measure
RAKE	43.24%	68.19%	52.90%
CRF	63.44%	64.94%	62.52%
B-LSTM	81.88%	83.05%	82.37%
CNN-B-LSTM	82.00%	82.99%	82.41%
W-CNN-B-LSTM	**82.52%**	**83.82%**	**83.06%**

4.2 Evaluation

For our experiments, we employed 10-cross-validation on our dataset. We use partial evaluation which was proposed by Seki et al. [12] to get precision, recall, F1-measure for evaluation metrics. This evaluation is used because we treat keyphrases as sequences labeling task.

4.3 Results and Analyisis

For comparison, we implemented RAKE [9] and CRF [1]. For CRF, we use the same features as proposed above. Moreover, We also conducted an experiment to proof the effectiveness of our CNNs layer by comparing B-LSTM with (CNN-B-LSTM) or without CNNs layer. We also compared our model with (W-CNN-B-LSTM) or without weighting layer.

As we can see in Table 2, RAKE performed the worst on Indonesian user-generated medical question since RAKE were specially devised for formal text. Of the other methods, CRF performed the worst. For CRF, based on our observation from the result provided by CRF, the method failed to predict long sequences as a keyphrase. Our CNNs layer also shows improvement in precision and F1 score when compared to standard B-LSTM. We suggest that CNNs layer can extract more features implicitly from neighboring words. Moreover, our weighting layer shows improvement compared to CNN-B-LSTM. Hence, the best result was obtained by W-CNN-LSTM. This is indicated that the feature weighting process worked well and could to some degree demonstrate the effectiveness of our model in keyphrase extraction for user-generated medical domain.

We performed an analysis to the aforementioned model, the result shown there are several errors. Some of the error are uncompleted keyphrases. For example, "`pusing pada (headache on)`", "`obat untuk (medicine for)`". Our model also failed to capture complex medicine name and disease. This is due to our small data set and not all medicine and disease are captured in our medical knowledge base or medical entities. Moreover, there are also special case that our model failed to handle. For example, "`penyembuhan untuk sakit kepala dan tenggorokan (treatment for headache and sore throat)`", the keyphrase need to be captured as a whole because "`sore throat`" have a connection with the term "treatment for". But, our model failed to capture that connection and extracted "treatment for headache" and "sore throat" separately instead.

5 CONCLUSION

We have proposed a model to address the task of keyphrase extraction on Indonesian user-generated contents in medical domain. Extracting information patient intentions and needs on user-generated medical contents is not a trivial task due to the fact that the content is usually short, contain many medical terms, 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 convolutional neural networks and bi-directional lstm. We also proposed a new layer for weighting our features. Furthermore, Several handcrafted features were proposed, including word importance to detect important word in a document and word stickiness for handling unstructured writing. Although our model outperforms baseline methods for keyphrase extraction, further improvements are needed.

REFERENCES

[1] 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 (2010), 962–971.

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

[3] Zhiguo Gong and Qian Liu. 2009. Improving keyword based web image search with visual feature distribution and term expansion. *Knowledge and Information Systems* 21, 1 (2009), 113–132.

[4] Raditya Herwando. 2016. Pemrosesan Pertanyaan Pada Sistem Tanya Jawab Bidang Kesehatan Dengan Pendekatan Pembelajaran Mesin. (6 2016).

[5] 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*. ACM, 1064–1074.

[6] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. Association for Computational Linguistics.

[7] 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*. 3111–3119.

[8] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

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

[10] Kamal Sarkar. A hybrid approach to extract keyphrases from medical documents. *International Journal of Computer Applications* 63 (????), 14–19.

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

[12] Kazuhiro Seki and Javed Mostafa. 2003. A probabilistic model for identifying protein names and their name boundaries. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*. IEEE, 251–258.

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

[14] 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*. ACM, 254–255.

[15] Chengzhi Zhang. 2008. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems* 4, 3 (2008), 1169–1180.

[16] 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*. 836–845.

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