The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents

IOP Conf. Series: Journal of Physics: Conf. Series **978** (2018) 012120 doi:10.1088/1742-6596/978/1/012120

The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents

D Gunawan[1*], C A Sembiring[1], M A Budiman[2]

[1] Department of Information Technology, Universitas Sumatera Utara, Jl. dr. Mansur No. 9 Kampus USU Medan 20155
[2] Department of Computer Science, Universitas Sumatera Utara, Jl. dr. Mansur No. 9 Kampus USU Medan 20155

*Email: danigunawan@usu.ac.id

Abstract. Rapidly increasing number of web pages or documents leads to topic specific filtering in order to find web pages or documents efficiently. This is a preliminary research that uses cosine similarity to implement text relevance in order to find topic specific document. This research is divided into three parts. The first part is text-preprocessing. In this part, the punctuation in a document will be removed, then convert the document to lower case, implement stop word removal and then extracting the root word by using Porter Stemming algorithm. The second part is keywords weighting. Keyword weighting will be used by the next part, the text relevance calculation. Text relevance calculation will result the value between 0 and 1. The closer value to 1, then both documents are more related, vice versa.

1. Introduction

The rapid stream of information emerges a new challenge to fetch specific information. As an abundance of web pages and documents are still increasing, focused crawler [1] is an alternative to gather the topic specific information. According to previously conducted research, filtering specific information can be done by using text relevance, classification or clustering algorithm. Besides focused crawling, text relevance can be used to find related web pages or documents that can be used to automatic text summarization [2], especially multi-documents summarization. Text relevance refers to the match between the goal related to the reader's goal [3].

Previous research implements web pages similarity in focused crawler [4][5]. This research utilizes a classification algorithm called Naïve Bayes to provide web pages' similarity. It does not calculate text relevance between web pages. This research classifies the web pages in the same category by providing bag of words.

In addition, a research about the efficient focused crawling strategy has been conducted [6]. This research proposes the combination of link structure and content similarity to crawl pages in certain topic. This research automatically extracted the topic feature vector and judging the topic similarity by using the combination of link structure and page content. Their experiment proves that their strategy has successfully increate the precision of fetching topic pages.

Another research utilizes clustering algorithm called self-organizing map to gather the similar web pages [7] which also can be used to recognize character [8]. This approach is a bit similar with [4] and [5] because it uses keywords to determine the web pages similarity. However, the keywords are not pre-

determined. The keywords are generated based on term frequency – inverse document frequency (TF-IDF). It also considers multiword expression (MWE) candidates [9] to determine the keywords.

This research is a preliminary research to obtain suitable text relevance algorithm to find related documents. It demonstrates the implementation of text relevance calculation between two documents by using cosine similarity. Cosine similarity is useful to measure the similarity between two documents based on terms of their subject matter [10]. The organization of this paper is as follow: the first section is the introduction about the preliminary research. Section two discusses the calculation of text relevance. Section three discusses the implementation of text relevance. The last section is conclusion of the research.

2. Methodology

This research is intended to demonstrate the implementation of text relevance calculation between a reference document with the second documents. Reference document is a document that is assigned as a reference to obtain the similarity between two documents. Second document is the document that will be compared to reference document to obtain its relevancy value with the reference document. We conduct several process to obtain text relevance between two documents. First process is text pre-processing. The result of this process is keywords of each document. The result will be used to calculate keywords weighting. The next process is text relevance calculation. In this process we calculate the relevance between two document by conducting cosine similarity.

Text pre-processing consists of several steps. The first step is punctuation removal. This punctuation will not influence the calculation as we do not include semantic. The next step is applying case folding that will turn the sentences into lower case. This action is required because there is no difference in meaning between "cell" and "CELL". This process is continued with tokenizing. After tokenizing, every token will be compared to the words in stop word list. All matched words will be removed and the rest will be the keywords to the related document. By implementing the stop word removal, we might reduce the time to the next process. As we do not consider semantic, then the word which has prefix will be turned into its root word. This process is called stemming. In this research we use Porter Stemming algorithm [11].

The calculation of text relevance requires keyword weighting. To obtain keyword weighting, we count the number of the same words. However, as there is possibility that the same words come in many forms, we should transform the words to their root word. This issue has been covered in text pre-processing. Formula (1) shows the calculation of keyword weighting, where w is weight of the keyword, w_i is the weight of the keyword i, and w_{max} is the weight of the keywords which has the maximum weight.

$$w = \frac{w_i}{w_{max}} \tag{1}$$

The text relevance is calculated by using cosine similarity. Formula (2) shows the calculation of text relevance between two documents, where $wkre$ is the weight of the reference document, which is obtained from the same keywords between reference document and second document, $wkse$ is the weight of the second document, which is obtained from the same keywords between reference document and the second document, wkr is the weight of the keywords in reference document and wks is the weight of the keywords in second document. As shown in formula (3), $wkre$ is obtained by dividing the sum of the weight of the same keywords between reference document and second document ($\sum kre$) with the maximum keyword weight in the reference document (wkr_{max}). Meanwhile, $wkse$ is obtained by dividing the sum of the weight of the same keywords between second document and reference document ($\sum kse$) with the maximum keyword weight in the second document (wks_{max}). This calculation is shown in formula (4). As shown in formula (5), wkr is obtained by dividing the sum of the keywords' weight in reference document ($\sum kr$) with the maximum keyword weight in the reference

document (wkr_{max}). Meanwhile, wks is obtained by dividing the sum of the keywords' weight in second document ($\sum ks$) with the maximum keyword weight in the second document (wks_{max}). This calculation is shown in formula (6).

$$relevance = \frac{wkre\ .wkse}{\sqrt{wkr^2}\ .\sqrt{wks^2}} \tag{2}$$

$$wkre = \frac{\sum kre}{wkr_{max}} \tag{3}$$

$$wkse = \frac{\sum kse}{wks_{max}} \tag{4}$$

$$wkr = \frac{\sum kr}{wkr_{max}} \tag{5}$$

$$wks = \frac{\sum ks}{wks_{max}} \tag{6}$$

3. The Implementation

This section will discuss the implementation of text relevance approach to obtain related document. The implementation begins with text pre-processing. In the text pre-processing we will discuss every process involved. Next, we will discuss keywords extraction after the text has been pre-processed. Then we will discuss about normalization of the keywords. The last, we will discuss about text relevance calculation.

Text pre-processing is begun with punctuation removal. Then we apply case folding to set all the letters to lower case. After applying case folding, we tokenize the text into separated tokens. Some of these token will be removed if the term is listed in stop word list. After the tokens are clean without any meaningless term, then the next step is stemming. Stemming process is done to obtain root word that is used to calculate text relevance between two documents. In this research we use Porter Stemming algorithm to stem the tokens.

Table 1. Keywords extraction

Document	Keywords	Category
Reference document	Adenocarcinoma:9; cell:9; scc:8; xenograft:8; region:7; distribut:6; cto:5; fdb:4; cuatsm:4; necrot:4; origin:3; intratumor:3; high:3; compar:3; overlap:3; accumul:3; live: 3; deriv:2; adduct:2; colon:2; cancer: 2; cucuatsm:2; characterist:2; pimonidazol:2; tumor:2; occur: 2; studi:2; observ:2; lung:2; dualtrac:1;	Biology and medical
Second document	Ffphpa:12; uptak:10; amino:8; radiotrac:5; ffet:4; imag:4; studi:4; prepar:3; cancer:3; pet:3; emt:3; compar:3; min:2; crosscoupl:2; asc:2; pdmediat:2; suvmin:2; maximum:2; transport:2; reach:2; cell:2; evalu:2; small:2; anim:2; reaction:2; high:2; radiochem:2; inhibit:2; cellular:2;	Biology and medical

The result after stemming process is keywords extraction from a document. For example, the first record in table 1 is the topic document that is used as reference by the second document (second record in table 1) to show the relevancy between both documents. The keywords are written in root word and its number of appearance in a document (separated with colon). One keyword and another is separated

with semicolon (";"). For example, "cell:9; intratumor:3; cancer:2;" means the term "cell", "intratumor", and "cancer" has been appeared in document as much as 9, 3 and 2 times respectively. These documents are grouped in to the same category namely "Biology and medical".

The next process is calculating weight of the keywords. For example, we will use the data in table 1 for weighting calculation. The weight of the term "cell" and "intratumor" from the reference document is 9 and 3 respectively. This weight is obtained from the number of term appearance in the text. Thus, the term "cell" is known as the maximum weight (w_{max}). According to formula (1), the keyword weight of the term "intratumor" (w_i) is:

$$w = \frac{3}{9}$$

$$w = 0.33$$

The weight of "intratumor" keyword is 0.33. By using the same formula, we can obtain the keyword weight of the reference document in table 1. The snippet result is shown in table 2.

Table 2. Keyword weighting

Keyword	w_i	w_{max}	w
adenocarcinoma	9	9	1
cell	9	9	1
scc	8	9	0.89
xenograft	8	9	0.89
region	7	9	0.78
distribut	6	9	0.67
intratumor	3	9	0.33
compar	3	9	0.33
pimonidazol	2	9	0.22
tumor	2	9	0.22

Text relevance is calculated by utilizing cosine similarity function as shown in formula (2). The calculation involves the same keywords from the both documents (reference document and second document) to obtain the weight. For example, as shown in table 3, the same keywords from reference document and second document are: cell, high, compar, cancer and study.

Table 3. The same keywords between reference and second document

Keywords	**Weight**	
	Reference document	**Second document**
Cell	9	2
High	3	2
Compar	3	3
Cancer	2	3
Study	2	4

According to table 1, the maximum weight of the reference document is 9 and the maximum weight of the second document is 12. Then, according to formula (3), weight of the reference document ($wkre$) is calculated below:

$$wkre = \frac{9 + 3 + 3 + 2 + 2}{9} = 2.11$$

Then, the weight of the second document ($wkse$) is calculated according to formula (4). The calculation is shown below:

$$wkse = \frac{2 + 2 + 3 + 3 + 4}{12} = 1.67$$

Next, the keywords weight in reference document (wkr) is calculated according to formula (5). The calculation is shown below:

$$wkr = \frac{110}{9} = 12.22$$

After that, the keywords weight in reference document (wks) is calculated according to formula (6). The calculation is shown below:

$$wks = \frac{104}{12} = 8.67$$

After we obtain all the parameters ($wkre$, $wkse$, wkr, and wks), then we calculate the text relevance according to formula (2). The calculation is shown below:

$$relevance = \frac{2.11 \,.\, 1.67}{\sqrt{12.11^2} \,.\, \sqrt{8.67^2}}$$

$$relevance = \frac{2.46}{105.97}$$

$$relevance = 0.023$$

In the case of information retrieval, the result of cosine similarity will be between 0 and 1. If the value is closer to 1 (one), then it means the second document is very relevant with the reference document. In this result, the relevance value is 0.023. This means the value is closer to 0 (zero). Therefore, we can conclude that the second document is not relevant to the reference document.

4. Conclusion

Focused crawler can be used to filter web pages and documents into topic specific information. Topic specific information can be filtered by using classification, clustering or text relevance method. This research is a preliminary research to demonstrate the implementation of text relevance between two documents. In this research, text relevance is calculated by using cosine similarity, where the value which is closer to 1 is considered as more relevant, vice versa. In the future, research should conduct the text relevance implementation in Bahasa Indonesia.

References

[1] Chakrabarti S, van den Berg M, Dom B 1999 Focused crawling: a new approach to topic-specific Web resource discovery *Comput. Networks* **31** 11–16 pp 1623–1640
[2] Gunawan D, Pasaribu A, Rahmat R F, and Budiarto R 2017 Automatic Text Summarization for Indonesian Language Using TextTeaser *IOP Conf. Ser. Mater. Sci. Eng.* **190** 1 p 12048
[3] Hager P J *et al*. 2012 Text Relevance in *Encyclopedia of the Sciences of Learning* Springer US pp 3307–3310
[4] Amalia A, Gunawan D, Nazwan A, and Meirina F 2016 Focused crawler for the acquisition of health articles *2016 Int. Conf. on Data and Software Engineering (ICoDSE)* pp 1–6
[5] Gunawan D, Amalia, and Najwan A 2017 Improving Data Collection on Article Clustering by Using Distributed Focused Crawler *J. Comput. Appl. Informatics* **8** 1 pp 39–50
[6] Cheng Q, Beizhan W, and Pianpian W 2008 Efficient focused crawling strategy using combination of link structure and content similarity *Proc. 2008 IEEE Int. Symp. IT Med. Educ. ITME 2008*
[7] Gunawan D, Amalia A, and Charisma I 2017 Clustering Articles in Bahasa Indonesia Using Self-Organizing Map *2017 Int. Conf. on Electrical Engineering and Informatics*
[8] Gunawan D, Arisandi D, Ginting F M, Rahmat R F, and Amalia A 2017 Russian Character Recognition using Self-Organizing Map *J. Phys. Conf. Ser.* **801** 1 p 12040
[9] Gunawan D, Amalia A, and Charisma I 2016 Automatic extraction of multiword expression candidates for Indonesian language *2016 6th IEEE Int. Conf. on Control System, Computing and Engineering (ICCSCE)* pp 304–309
[10] Singhal A 2001 Modern Information Retrieval: A Brief Overview *Bulletin of the Technical Committee on Data Engineering* pp 35–43
[11] Porter M 2006 “The Porter Stemming Algorithm" Available: https://tartarus.org/martin/PorterStemmer/