ACSIJ Advances in Computer Science: an International Journal, Vol. 2, Issue 3, No.4, July 2013
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

A New Approach To Focused Crawling: Combination of Text summarizing With Neural Networks and Vector Space Model

Fahim Mohammadi,

Department of Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran
fahim.mohammadi88@gmail.com

Abstract

Focused crawlers are programs designed to browse the Web and download pages on a specific topic. They are used for answering user queries or for building digital libraries on a topic specified by the user. In this article we will show how summarizing of web pages is needed for improving performance of a crawler which uses vector space model to rank the web pages. A neural network is trained to learn the relevant characteristics of sentences that should be included in the summary of a web page. Then the neural network will be used as a filter to summarize web pages. Finally, the crawler will use vector space model to rank summaries instead of web pages.

Keywords: *Focused Crawlers, Search Engine, Neural Network, Vector Space Mode, Text summarizing.*

1. Introduction

The World Wide Web is a huge information source with billions of web pages. General purpose search engines such as Google, Yahoo, MSN and Ask have appeared in order to help users to find information on the Web. These search engines are complex but not propagated over the Web in real time [1, 2]. Instead they index, analyze and categorize Web information stored locally in data repositories to be used for answering user queries.

Crawlers are programs for gathering locally information from the Web [3]. Focused crawlers in particular are used to satisfy the need of individuals or organizations to create and maintain locally digital libraries on a subject or for answering queries for which were not satisfied by results of a general purposed search engine [4].

With the explosion of the World Wide Web, text summarizing has become an important and timely tool for assisting and interpreting text information. The Internet provides more information than is usually needed. Searching for relevant documents through an overwhelming number of documents available becomes an important problem. Summarizing is a useful tool for selecting relevant texts, and for extracting the key points of each text [5].

The crawler takes as input a user query that describes the topic and a set of starting URLs. The crawling starts from the seed URLs. The crawler assigns a priority value to visited pages according to their relevance to the query. Then the web pages are ordered by relevance. The crawler will visit the most relevant web page first. The criterion for relevance estimation between a retrieved web page and a user query is defined as the similarity between the text of the visited web page with query. This is computed using a text similarity model such as Boolean or the Vector Space Model [6].

In the next part we show the use of VSM. In part *Pruning* we discuss about why VSM doesn't satisfy similarity estimation by itself and we define a function to prune irrelevant parts of a web page. In part *Text summarizing process* we show why web page summarizing process is needed, then we propose a machine learning approach that uses artificial neural networks to produce summary of text of a web page. In part *Sentence Selection*, we combine summarizing process with Vector Space Model to make performance of crawler better. Finally, in part *Results and Analysis* we show that this combination is good and this contribute crawler to achieve reliable results.

2. Vector Space Model

In the statistically based vector-space model, a document is conceptually represented by a vector of keywords extracted from the document, with associated weights representing the importance of the keywords in the document and within the whole document collection; likewise, a query is modeled as a list of keywords and each keyword has a weight representing the importance of the keyword in the query.

In this model the weight of a term in a document vector can be determined in many ways. A common approach uses the so called $tf \times idf$ method, in which the weight of a term is determined by two factors: how often the term j occurs in the document i (the term frequency $tf_{i,j}$) and how often it occurs in the whole document collection (the document frequency df_j). Precisely, the weight of a term j in document i is $w_{i,j} = tf_{i,j} \times idf_j = tf_{i,j} \times \log N/df_j$ where N is the number of documents in the document collection

and idf stands for the inverse document frequency. Once the term weights are determined, we need a ranking function to measure similarity between the query and document vectors. A common similarity measure, known as the cosine measure, determines the angle between the document vectors and the query vector when they are represented in a V-dimensional Euclidean space, where V is the vocabulary size. Precisely, the similarity between a document Di and a query Q is defined as Eq.(1), where $w_{Q,j}$ is the weight of term j in the query, and is defined in a similar way as $w_{i,j}$.

$$sim(Q,D_i) = \frac{\sum_{j=1}^{V} w_{Q,j} \times w_{i,j}}{\sqrt{\sum_{j=1}^{V} w_{Q,j}^2 \times \sum_{j=1}^{V} w_{i,j}^2}} \quad (1)$$

Unfortunately, the computation of the normalization factor is extremely expensive because the term in the normalization factor requires access to every document term, not just the terms specified in the query. Nor can the normalization factor be pre-computed under the $tf \times idf$ method, because every insertion and deletion on the document collection would change idf and thus the pre-computed normalization factors.
Because the exact vector space model is very expensive to implement, in this article we used method 4 of VSM in [7] as Eq.(2). This method only makes use of term frequencies in the calculation and ignores idf . It simplifies the computation as well as saving the file structure needed for storing the df values.

$$Sim(Q,D_i) = \sum_{j=1}^{v} w_{Q,j} \times tf_{i,j} \quad (2)$$

3. Pruning

The vector space model, in some cases doesn't satisfy determining relevance between a web page and a query by itself. The vector space model will not consider cheats in estimation of similarity of web pages to query. For example, some web pages have hidden parts like a series of keywords with same color of background. So users cannot see them but crawlers will. The influences of these cheats to estimation of similarity are not negligible. In addition words with very small font size in web pages are not very important because the writer does not think they are important and if they were, the writer would write them with bigger font. Sometimes it is a way to cheat the crawlers and search engines. In order to increase page importance the writer makes series of keywords with same color of the background in the end of web pages. So we need a function to eliminate the influences of the cheats and non-important parts of a web page and make estimation easier. We determined 4 kinds of sentences to be pruned so they cannot bother estimation of similarity.

3.1 Sentences which their color is same with the background

There may be a large group of sentences which are have color of the background. The purpose is to increase rank of the web page by cheating the crawlers and search engines. In order to eliminate the influence of cheating, the crawler discards this kind of sentences.

3.2 Very short sentences

Generally in a text, information of the short sentences is less than information of longer ones. The short ones are not likely to be selected for making a summary. So the crawler discards the sentences have less than 5 words to decrease the process time and improve the performance of neural network.

3.3 Sentences with very small fonts

Usually, sentences with very small fonts are used as footnotes, copy right or additional comments witch are not directly relevant to the topic. They also may be used for cheating by repeating keywords over and over and usually at the end of web pages. So it is better and safer to not bring them in the summary of the web page. In implementations we discarded sentences with xx-small font size in HTML code.

3.4 Sentences in which a word is repeated so many times

There are lots of web pages with some unmeaning sentences made by repeating keywords over and over to cheat the crawlers and search engines and increase their importance. They are also more likely to be selected by neural network for making summary because of large number of keywords they have. So we need to make sure that they will not exist in text. We set maximum number of repetition of words in a sentence to 3.
The web pages are full of irrelevant information such advertisings and news. In addition, the anchor texts of their links are often not sufficient to convey their important points. Therefore, a summarizing tool would be extremely

useful for extracting the key points of each text. The summarizing has benefits. Redundant information is such a noise and it distracts the process of similarity determination. Summarizing process will discard redundant information, so estimation of similarity will be easier and more reliable.
We propose a machine learning approach that uses artificial neural networks to produce summaries of text of a web page. A neural network is trained on a corpus of web pages. The network discovers the importance (and unimportance) of various features used to determine the summary-worthiness of each sentence.

4. Features

Each document is converted into a list of sentences. Each sentence is represented as a vector $[f_1, f_2, f_3, \ldots, f_6]$, composed of 6 features as shown in table 1.

Features *f1* to *f3* represent the location of the sentence within its paragraph. It is expected that these features would contribute to selecting summary sentences. In [8] have shown that summaries consisting of leading sentences outperform most other methods in this domain, and Baxendale in [9] demonstrated that sentences located at the beginning and end of paragraphs are likely to be good summary sentences. Feature f1 indicates whether any

f1	Paragraph follows anchor text
f2	Sentence location in paragraph
f3	First sentence in paragraph
f4	Sentence length
f5	Number of anchor text words in the sentence
f6	Font size of sentence

Table 1: Six Features of a Web Page

word of anchor text of the link that refers to this web page, is in the following paragraph or not. So the anchor text plays an important role in selecting sentences and making summaries.
Feature *f4*, sentence length, is useful for filtering out short sentences such as datelines and author names commonly found in web pages. We also anticipate that short sentences are unlikely to be included in summaries [10]. Feature *f5* indicates the number of title words in the sentence, relative to the maximum possible. It is obtained by counting the number of matches between the content words in a sentence, and the words in the title. This value is then normalized by the maximum number of matches. Finally, feature *f6* refers to font size of a sentence. Font size of sentences in web pages is not fixed. The font size of a sentence is completely up to the writer's purpose. Sentences with bigger font size are expected to be important and sentences with small font size could be not important. The selection of features plays an important role in determining the type of sentences that will be selected as part of the summary and, therefore, would influence the performance of the neural network.

5. Text Summarizing Process

There are two phases in summarizing process: neural network training, and sentence selection. The first step involves training a neural network to recognize the type of sentences that should be included in the summary. The second step, sentence selection, uses the neural network to filter the text and to select only the highly ranked sentences. This step controls the selection of the summary sentences in terms of their importance. These two steps are explained in detail in the next two sections.

6. Neural Network Training

The first phase of the process involves training the neural networks to learn the types of sentences that should be included in the summary. This is accomplished by training the network with sentences in several test paragraphs where each sentence is identified as to whether it should be included in the summary or not. This is done by an expert human reader. We use a three layered feed forward neural network, which has been proven to be a universal function approximator [11]. It can discover the patterns and approximate the inherent function of any data to an accuracy of 100%, as long as there are no contradictions in the data set. The neural network consists of six input-layer neurons, five hidden-layer neurons, and one output layer neuron.

7. Sentence Selection

Once the network has been trained, it can be used as a tool to filter sentences in any paragraph and determine whether each sentence should be included in the summary or not.
Then web pages will be ordered by the similarity of their summaries to query. It means the crawler will assign priority value to the summary instead of whole web page. In this case relevance of a web page is up to relevance of its summary. Then the crawler proceeds by visiting the most relevant web page first.

There may be a problem in this way. What if a web page has just a paragraph that is much related to the anchor text and several irrelevant paragraphs? The neural network will make most of summary with selecting the sentences of the relevant paragraph. In this way the result summary does not cover the whole web page and the crawler may give a high similarity value to the summary but the whole web page is not relevant to the query. So we need to normalize the similarity value of summaries. The crawler will give similarity value s_{S_i} to summary of web page i as Eq.(4). We define D_i as the division of the number of sentences of the summery of web page i by the number of sentences of whole page as Eq.(3). In fact D_i indicates the density of sentences of page i that are relevant to anchor text. Finally, the similarity value of web page s_i will be achieved by multiplying of D_i in s_{S_i}.

$$D_i = \frac{N_{s_i}}{N_i} \tag{3}$$

$$S_i = D_i * S_{S_i} \tag{4}$$

The order of functions of the suggested crawler is shown in figure 1.

8. Results and Analysis

We used 450 web pages from the Internet with various topics such as technology and sports to train the network with an average of 185 sentences. The entire set consists of 83250 sentences. Every sentence is labeled as either a summary sentence or an unimportant sentence by a human reader. The accuracy of the network ranged from 96% to 100% with an average accuracy of 98.2% when compared to the summaries of the human reader. The network was able to select all sentences that were labeled as summary sentence in most of the web pages. The performance of the text summarizing process depends heavily on the culture and style of the human reader and to what the human reader deems to be important to be included in the summary. So if we are attempting to build a digital library for French sport web pages it is a good idea to train a neural network with a set of sport web pages and also in French and because of cultural issues it is better to label each sentence by a French human reader.

In order to describe the performance of the crawler clearly, 20 web pages selected from the whole set which ranked by a human reader are shown in table 2. The web pages also are ranked by VSM as shown in table 3. Two web pages had some hidden sentences. Two web pages had Sentences in which a word is repeated very much. One page had some meaningful sentences with xx-small font size but the sentences were duplicated from main text sentences.

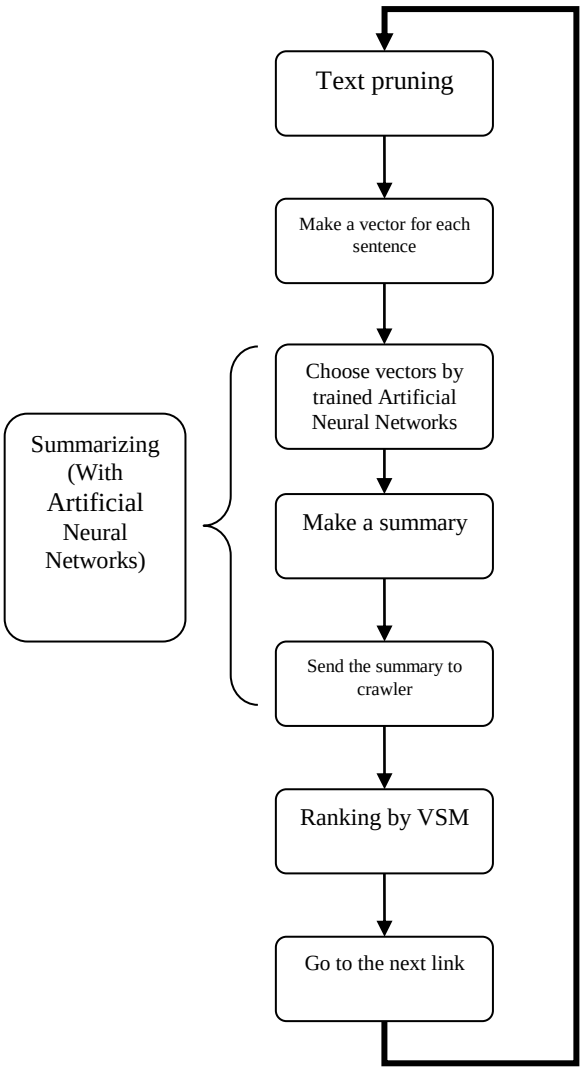

Fig.1 Order of the crawler's tasks

Three of the web pages were completely irrelevant to query. Four pages had low relevance. One page had just one relevant paragraph to its anchor text and accidentally very relevant to crawler's query. The relevance of the rest of the web pages to their anchor texts and also to crawler's query was medium.

After pruning, all sentences of each web page mapped to vectors by a human reader. Then the trained neural network made a summary of each web page by selecting the summary-worth sentences. Because of the equation 3 the length of the summaries is not necessary to be limited. The summaries are ranked by VSM as shown in table 4. As

you see, ranking after pruning and summarizing is much closer to ranking made by a human reader. 11 web pages are exactly on their positions in the ranking table made by a human reader and the other web pages are very close to their actual positions. But none of the web pages in ranking made by just VSM, table 3, are in their actual positions. So it means that pruning and summarizing made a good effect to ranking process of a crawler which uses vector space model.

Page	Comment	Rank
P100		1
P101		2
P103		3
P102		4
P105		5
P108		6
P106	hidden	7
P107		8
P104	hidden	9
P109	repeated	10
P111	average	11
P110	1 very relevant paragraph	12
P112	average	13
P113	repeated	14
P114	Very small font	15
P116	average	16
P115	average	17
P119	irrelevant	18
P118	irrelevant	19
P117	irrelevant	20

Table 2: 20 sample web pages are ranked by a human.

Page	Comment	Rank
P106	hidden	1
P109	repeated	2
P105	hidden	3
P100		4
P102		5
P113	repeated	6
P101		7
P114	Very small font	8
P103		9
P108		10
P105		11
P107		12
P115	average	13
P110	1 very relevant paragraph	14
P112	average	15
P116	average	16
P111	average	17
P117	irrelevant	18
P119	irrelevant	19
P118	irrelevant	20

Table 3: 20 sample web pages are ranked by VSM.

Page	Comment	Rank
P100		1
P101		2
P102		3
P103		4
P104	hidden	5
P105		6
P106	hidden	7
P107		8
P108		9
P109	repeated	10
P110	1 very relevant paragraph	11
P111	average	12
P112	average	13
P113	repeated	14
P114	Very small font	15
P115	average	16
P116	average	17
P117	irrelevant	18
P118	irrelevant	19
P119	irrelevant	20

Table 4: 20 sample web pages are ranked by proposed Crawler

9. Conclusion

Focused crawlers are programs designed to browse the Web and download pages on a specific topic. In this article we showed how summarizing of web pages is needed for improving performance of a Focused crawler which uses vector space model to rank the web pages. A neural network was trained to learn the relevant characteristics of sentences that should be included in the summary of a web page. The neural network used as a filter to summarize web pages. Finally, the crawler used vector space model to rank summaries instead of web pages. As we saw in part Results and Analysis, the combination is good and this contribute crawler to achieve reliable results. The performance of the text summarizing process heavily depends on culture and Native language of the human reader. So we extremely recommend training the artificial neural network for a specific language and with a native human reader. Further works may be improving the pruning method more effectively and customize them for specific types of web pages, for example news web pages. The number of features for training the neural network may experimentally or typically increase or decrease to improve performance of the sentence selection process.
This work followed the idea of combining VSM and text summarizing with artificial neural networks and is one of the pioneers in this field. So it has a long way to destination and certainly will grow up by time.

References

[1] "Web Search for a Planet The Google Cluster Architecture" LA Barroso, J Dean, U Holzle -Micro, IEEE, 2003.

[2] "Very Large Scale Retrieval and Web Search" D Hawking, N Craswell, In E. Voorheesand D. Harman, editors, TREC: Experiment and Evaluation in Information Retrieval. MIT Press, 2005.

[3] "Design and Evaluation of Topic Driven Focused Crawlers" for The World Wide Web By Batsakis Sotirios a Thesis submitted in partial fulfillment of the requirements for the degree of Master of Computer Engineering , Chania, November 2007.

[4] "A Survey of Focused Web Crawling Algorithms", B. Novak, Ljubljana, Slovenia: SIKDD at multi conference IS, October 2004.

[5] "Text Summarization Using Neural Networks", Khosrow Kaikhah, Texas State University-San Marcos, Dept. of Computer Science, 2004.

[6] "A Vector Space Model for Automatic Indexing" G Salton, A Wong, CS Yang–Communications of the ACM, 1975.

[7] " Document Ranking and the Vector-Space Model", DIK L. LEE, Hong Kong University of Science and Technology, HUEI CHUANG, Information Dimensions, KENT SEAMONS, 1997.

[8] R. Brandow, K. Mitze and L. Rau, "Automatic Condensation of Electronic Publications by Sentence Selection", Information Processing and Management, vol. 31(5), pp. 675-685, 1995.

[9] P.B. Baxendale, "Machine-Made Index for Technical Literature: An Experiment" IBM Journal of Research and Development, vol. 2(4), pp. 354-361, 1958.

[10] J. Kupiec, J. Pederson and F. Chen, "A Trainable Document Summarizer", Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, pp. 68-73, 1995.

[11] M.R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems", Journal of Research of the National Bureau of Standards, vol. 49, pp. 409-436, 1952.