Automatic Keyword Extraction Using Linguistic Features

Xinghua Hu, Bin Wu
Baskin School of Engineering
University of California, Santa Cruz
pwp@ucsc.edu, binwu@soe.ucsc.edu

Abstract

This paper describes a novel keyword extraction algorithm Position Weight (PW) that utilizes linguistic features to represent the importance of the word position in a document. Topical terms and their previous-term and next-term co-occurrence collections are extracted. To measure the degree of correlation between a topical term and its co-occurrence terms, three methods are employed including Term Frequency Inverse Term Frequency (TFITF), Position Weight Inverse Position Weight (PWIPW), and CHI-Square (χ^2). The co-occurrence terms that have the highest degree of correlation and exceed a co-occurrence frequency threshold are combined together with the original topical term to form a final keyword. With the linear computational complexity of the algorithm, the vector space of documents in a large corpus or boundless web can be quickly represented by sets of keywords, which makes it possible to retrieve large-scale information fast and effectively.

1. Introduction

With the trends to non-edge development of information, especially for the WWW information retrieval, search engines always return overwhelmed related documents, which makes people hardly decide what to read only from the fuzzy snippets. Another similar bewildering case is the email. Users feel frustrated when they fail to distinguish ham from spam only by the flowery titles of numerous emails. Moreover, in the domain of news browsing, some news with valuable content but unattractive titles will often be ignored, as readers have to find a tradeoff between their limited time and the desire for learning the world (Li, et al, 2003, [11]).

One solution to the above problems is to extract keywords from each document (a document here can be a semi-structured web page or email) which feature the main content of the document. Meaningful keywords will assist users to grasp the essential topic of each document in one sight and help them decide what to read. Here the concept of a keyword is not limited to one word but can be a two or three compound-word phrase.

If each document can be represented by a few of keywords (Ikonomakis, et al, 2005, [9]), these documents can be easily retrieved by calculating the similarity of their keywords to the query terms. This algorithm produces a small linear time complexity since the dimension of the vector which presents the document has been largely diminished to a number of keywords. Furthermore the retrieved documents that include the query terms in their keywords indices will be more relevant than those only mentioning the query terms in their ordinary content.

Some previous work has been done before. Krulwich and Burkey (1996, [7]) extract "semantically significant phrases" from documents based on the structural and superficial features of the documents. The extractor proposed by Turney and Peter (2000, [12]) extracts a small number of keywords from documents by scoring candidate phrases on a number of features.

2. Algorithm

2.1 Position Weight

The word position plays an important role in linguistics. Words in different positions carry different entropy (Troyka and Thweatt, 2003, [8]; Clouse, 2002, [2]). If the same word appears in the introduction and conclusion paragraphs, usually the first and the last paragraphs of the document, the words generally carry more information. Also, words shown in the leading and summarization sentences are often more important than those in other positions of the same paragraph. Here we employ a new method named Position Weight (PW) to record the importance of a word position. A common document consists of paragraphs (title is

Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)
0-7695-2702-7/06 $20.00 © 2006 IEEE

regarded as a special paragraph); a paragraph is formed by sentences; and a sentence is made up of words. The PW of a term must consider these three important elements: paragraph, sentence and word. We define the PW of a term t in one specific position as

$$pw(t_i) = pw(t_i, p_j) \cdot pw(t_i, s_k) \cdot pw(t_i, w_r)$$

Where $pw(t_i, p_j)$ represents the PW of term t in the paragraph j; $pw(t_i, s_k)$ represents the PW of term t in the sentence k; $pw(t_i, w_r)$ represents the PW of term t as a word form r.

The total weight of term t in a document is the sum of the weights of all positions it appears. If term t appears m times in a document d, its PW is

$$PW(t,d) = \sum_{i=1}^{m} pw(t_i)$$

To calculate the PW, we first give two words collections which are the statistic result of all articles in the book "structured reading" (Troyka and Thweatt, 2003, [8]).

Transition phrase collection:

{however, but, yet, nevertheless, nonetheless, although, though, whereas, despite, in spite of, while, therefore, thus, hence, accordingly, consequently, actually, instead, indeed, practically, virtually, essentially}

Summary phrase collection:

{in sum, summary, conclusion, finally, in all, as a whole, in short, in a word, in brief, conclusive, conclusively, consequence, conclude, result, results in, lies in, leads to}

Then each parameter of PW is defined as below, according to our experiment results.

2.1.1 Paragraph Weight ($pw(t_i, p_j)$)

If paragraph j is a main title, it gets a weight of $4\text{Unit}_{\text{paragraph}}$.

If paragraph j is a subtitle, it gets a weight of $3.5\text{Unit}_{\text{paragraph}}$.

If paragraph j is the leading or conclusion paragraph, it gets a weight of $3\text{Unit}_{\text{paragraph}}$. Here the conclusion paragraph is the paragraph which begins with any phrase in the "summary phrase collection".

If paragraph j is the transition paragraph, it gets a weight of $2\text{Unit}_{\text{paragraph}}$. Here a transition paragraph stands for the paragraph which begins with any phrase in the "transition phrase collection".

If paragraph j is other paragraph not mentioned above, it gets a weight of $1\text{Unit}_{\text{paragraph}}$.

$\text{Unit}_{\text{paragraph}}$ is a base value of a common paragraph.

2.1.2 Sentence Weight ($pw(t_i, s_k)$)

If sentence k is a title, it gets a weight of $4\text{Unit}_{\text{sentence}}$.

If sentence k is the leading or conclusion sentence, it gets a weight of $3\text{Unit}_{\text{sentence}}$. Here a conclusion sentence stands for the sentence which contains any phrase in the "summary phrase collection".

If sentence k is the second sentence, last non-conclusion sentence or the special transition sentence, it gets a weight of $2\text{Unit}_{\text{sentence}}$. Here a transition sentence is the sentence which contains any phrase in the "transition phrase collection".

If sentence k is other sentence not mentioned above, it gets a weight of $1\text{Unit}_{\text{sentence}}$.

If sentence k is the example sentence, it gets a weight of 0. Here an example sentence means the sentence which begins with any phrase of "for example, as an example, for instance", which is used as an example to explain something concretely.

$\text{Unit}_{\text{sentence}}$ is a base value of a common paragraph.

2.1.3 Word Weight ($pw(t_i, w_r)$)

If word r is capitalized and combined with digit, then it gets a weight of $3\text{Unit}_{\text{word}}$, as such a word is always a name, a brand or a hallmark of a special object related to the document theme.

If word r is capitalized without digit, then it gets a weight of $2\text{Unit}_{\text{word}}$.

Otherwise a word r gets a weight of $1\text{Unit}_{\text{word}}$.

$\text{Unit}_{\text{word}}$ is a base value of a common word.

In our experiment, we set

$$\text{Unit}_{\text{paragraph}} = \text{Unit}_{\text{sentence}} = \text{Unit}_{\text{word}}$$

2.2 Term Importance

In our experiment, we employ three methods to estimate the importance of a term in the compound collection which will be discussed later.

2.2.1 Term Frequency Inverse Term Frequency (TFITF)

This measure comes from TFIDF. The difference is that TFIDF is used in the corpus while TFITF is used in single document. Term frequency (TF) is the number of term t occurring in the collection. The Inverse Term Frequency (ITF) is used to measure the inverse frequency of term t among the document. It quantifies the dependency of term t on topical terms. Terms that appear more in the collection but show relatively less in the document will get a higher TFITF value.

$$TFITF(t) = \frac{f_{clt}}{\max f_{clt}} \cdot \log \frac{F_d}{F_t}$$

Where f_{clt} is the frequency of term t appearing in the collection; $\max f_{clt}$ is the maximum frequency of terms appearing in the collection; F_d is the total number of terms in the document; F_t is the total frequency of term t showing in the document; $\frac{f_{clt}}{\max f_{clt}} or \frac{F_d}{F_t}$ is normalized frequency of the document.

2.2.2 Position Weight Inverse Position Weight (PWIPW)

Position Weight (PW) is the PW value of term t in the specific place where it is found. The Inverse Position Weight (IPW) is used to measure the inverse Position Weight of term t among the document. It measures the dependency of term t on topical terms. Terms with more importance in the collection but less relative in document will get a higher PWIPW value.

$$PW(t) = \frac{pw_{clt}}{\max pw_{clt}} \cdot \log \frac{PW_d}{PW_t}$$

Where pw_{clt} is the sum PW of term t appearing in the collection; $\max pw_{clt}$ is the maximum PW of terms appearing in the collection; PW_d is the total PW of all terms in the document; PW_t is the total PW of term t showing in the document; $\frac{pw_{clt}}{\max pw_{clt}} or \frac{PW_d}{PW_t}$ is the normalized weight of the document.

2.2.3 CHI-Square (χ^2)

The χ^2 statistic measures the dependence of term t on the collection and can be compared to the χ^2 distribution with one degree of freedom to judge the extremeness. For convenience, we use the contingency table (Yang and Pedersen, 1997, [13]) of term and collection as shown in Table 1.

Table 1. Contingency table of term and collection

Title	In Collection	Not In Collection
Frequency of Term t	a	b
Frequency of Other Terms	c	d

Where a is the frequency of term t in the collection; b is the frequency of term t outside the collection; c is the total frequency of terms in the collection except t; d is the total frequency of terms outside the collection except t; N is the total number of terms in document.

The χ^2 measure is defined as

$$\chi^2(t) = \frac{N(ad-bc)^2}{(a+c)(b+d)(a+b)(c+d)}$$

3. Implementation

3.1 Text Indexing

For pre-processing, we do some common work in information retrieval. First we chunk the text of the document and eliminate the stop words included in the Fox stop list (1992, [3]), and leave those special words which have the transmissible or negative meaning like "however", "nevertheless" and etc. (Riloff, 1995, [4]). Secondly, we stem words using the Krovetz algorithm (1993, [10]) based on the WordNet Dictionary (Miller, 1990, [5]; Gonzalo, et al, 1998, [6]). Thirdly we calculate the PW based on the algorithm described.

3.2 Topical Term Selection

After calculating the PW of all the terms in a document, we do normalization and smoothing according to the maximum PW weight and the length of the document, since some terms have an excessive frequency compared to all others. Then we can extract topical terms by judgment $PW(t) > PW_{threshold}$. Here $PW_{threshold}$ can be approximated as 1/6 PW_{max} according to our statistic experiment result.

3.3 Compound Word Selection

Most of keywords are not single words, and they usually consist of two or three words, which are called compound words. As they are easy to understand, our keyword extraction task mainly target on this kind of compound words. In order to achieve this goal, for each topical term, we go through the document to extract all the terms before and after it. Thus we get two collections, the previous-term collection and the next-term collection.

Not all the words in a compound word will co-occur. They may be represented by a central word for conciseness after mentioned several times in an article. We utilize three term importance algorithms to calculate the correlation between the topical term and each term in the collections. One may think that the most frequent term in the collection will of most importance. In fact, this assumption is not always true. For instance, in an article about "Artificial Intelligence", the compound word "human computer" is more important than the compound word "electronic computer", although the word "electronic" appears more frequent in front of the word "computer".

No matter what kind of term importance algorithms is used, the most important term in previous-term and next-term collection will be picked out by choosing the one that has the highest value. Then if the co-occurrence frequency of the chosen term and the topical term is greater than $Freq_{threshold}$, the term will be compounded together with the topical term, otherwise it will be discarded. $Freq_{threshold}$ in our experiment is set to be 1/6 frequency of the topical term (Yang, 2001, [14]). After all these are done, the compound word will finally be sorted out and restored to its original word form in the document to present as a keyword.

3.4 Synthesizing and Ranking

In this step, we merge the sub sets to a main one in the keyword collection. For example, "discrete state" or "state machine" will be merged into "discrete state machine". We also discard those keywords without any noun or verb. In order to extract the most important keywords, we need to rank the keyword collection. In our experiment, we consider four factors: Uppercase/Digit Existence, Number of Compound Words, Average Word Length, Chosen Order of Topical Word. After synthesizing these parameters to one value with a certain ratio (we use ratio 1:2:2:3.5 according to our statistic experiment result which has the best performance), we can sort the keyword collection by the synthesized value. We finally extract the top 10 from this rank list to be the final output keywords.

4. Evaluation

Since our algorithm focuses on the keyword extraction for a single document, we chose 30 articles from 4 different categories --- academic, news, literature and spam email. For each article, we had 5 people look at it before extraction and let them manually select at least 3 keywords from it. Then we merged their results together to form a keyword answer set for every document.

We used different term importance algorithms to extract the top 10 keywords which is the upper limit for people to seize at one sight. We then asked the 5 people to choose words from the automatic extracted result which they thought were still important besides the answer set and marked it meaningful. We count meaningful words as they are important and effective to help people grasp the theme of the document. Meaningful words can be considered as compensation to the keywords answer set.

We employ two common used measurements in information retrieval domain --- precision and recall --- to evaluate the effectiveness of our algorithms. Precision can be calculated by the ratio of the exact and meaningful keywords to 10 terms derived from each algorithm. Recall can be calculated by counting how many keywords in the 10 automatically extracted keywords are exactly included in the answer set. The formulas are defined as

$$\text{Precision } P = \frac{Exact\ \&\ MeaningfulWords}{Retrived Words}$$

$$\text{Recall } R = \frac{ExactWords}{WordsofAnswerSet}$$

Here we take a very famous paper "Computing Machinery and Intelligence" written by Alan Turing (1950, [1]), the founder of computer science, as an example. In order to show the performance of PW, we also get the keywords of TF for each category, the topical words of which are chosen based on the Term Frequency (TF), as a compare set.

Tables 2, 3 and 4 show the measurements of PWIPW, χ^2, and TFITF, respectively, for the selection of compound words. There are two keyword columns in each table. The keywords listed on the left represent that their topical words were selected by the PW method, whereas the selection process of the topical words on the right is according to the TF method. The symbol [R] attached to a keyword indicates the correct keyword and [M] stands for the meaningful keyword. The rest are meaningless keywords. The precision and recall values are also presented below the tables.

Table 2. Compound Words Selected by PWIPW

[Topical-PW]	[Topical-TF]
Digital Computers [R]	discrete state machine [R]
Imitation Game [R]	interrogator [M]
discrete state machine [R]	digital computer [R]
Professor Jefferson	argument appears
Machine concerned [M]	human computer [R]
7 Learning Machines [R]	more expeditious
satisfactory support	imitation game [R]
scan interrogator [M]	well established
better witness Yes	right identification
human computer [R]	confusion between

Precision: P_{PW}=70%, P_{TF}=50%; Recall: R_{PW}=100%, R_{TF}=80%

Table 3. Compound Words Selected by χ2

[Topical-PW]	[Topical-TF]
Digital Computers [R]	discrete state machine [R]
Imitation Game [R]	interrogator [M]
discrete state machine [R]	digital computer [R]
Professor Jefferson	argument appears
Machine concerned [M]	more expeditious
comparison witness Yes	imitation game [R]
scan interrogator [M]	well established
satisfactory support	right identification
INTELLIGENCE [M]	machine cannot
7 Argument	confusion between

Precision: P_{PW}=60%, P_{TF}=40%; Recall: R_{PW}=60%, R_{TF}=60%

Table 4. Compound Words Selected by TFITF

[Topical-PW]	[Topical-TF]
Digital Computers [R]	discrete state machine [R]
Imitation Game [R]	interrogator [M]
Professor Jefferson	digital computer [R]
discrete state machine [R]	argument appears
Machine concerned [M]	well established
comparison witness Yes	imitation game [R]
7 Learning Machines [R]	right identification
satisfactory support	machine cannot
scan interrogator[M]	confusion between
INTELLIGENCE [M]	nervous system [M]

Precision: P_{PW}=70%, P_{TF}=50%; Recall: R_{PW}=80%, R_{TF}=60%

For the paper "Computing Machinery and Intelligence", the manually selected keywords are: digital computer, discrete state machine, learning machine, imitation game, human computer.

The overall precision and recall with respect to 30 articles are shown in Table 5 and 6.

Table 5. Overall Precision and Recall with Topical Words selected by PW

Topical-PW	PWIPW	TFITF	χ^2
Precision	85.56%	80.00%	81.11%
Recall	77.33%	71.39%	70.89%

Table 6. Overall Precision and Recall with Topical Words selected by TF

Topical-TF	PWIPW	TFITF	χ^2
Precision	73.33%	72.22%	70.00%
Recall	55.50%	47.28%	49.50%

From the experimental result, we can see: for topical word selection, the PW algorithm outperforms the TF algorithm; for compound word selection, the performance of three term importance algorithms shows as

$$PWIPW > TFITF \cong \chi^2$$

5. Conclusion

In this paper, we have developed the Position Weight (PW) algorithm to automatically extract keyword from a single document using linguistic features. The experiment results show that the PW algorithm has a great potential for keywords extraction, as it generates a better result than other existing approaches. With its linear computational complexity, the vector space of documents in large corpus or boundless semi-structured web pages can be quickly represented by keywords sets, which will make it possible to run information retrieval in an affordable time and space. The PW algorithm therefore might be able to play an important role in the information retrieval domain, especially for automatic keyword extraction.

6. Acknowledgements

It is a pleasure and honor to thank many colleagues who have collaborated with us in ISM of UCSC and their valuable comments and suggestions, particularly from Professor Yi Zhang. Comments from anonymous reviewers are also acknowledged.

7. References

[1] A. Turing. Computing machinery and intelligence. Mind, 59:433, 1950.

[2] B. F. Clouse. Progressions With Readings. Pearson Education, New Jersey, 2002.

[3] C. Fox. Lexical analysis and stoplists. Information Retrieval: Data Structures & Algorithms, 102–130. Prentice-Hall, New Jersey, 1992.

[4] E. Riloff. Little words can make a big difference for text classification. Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995.

[5] G. Miller. Wordnet: An on-line lexical database. International Journal of Lexicography, 3(4), 1990.

[6] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with WordNet synsets can improve text retrieval. In Proceedings of COLING-ACL '98 Workshop on Usage of WordNet in Natural Language Processing Systems, Montreal, Canada, August, 1998.

[7] Krulwich, Bruce & C. Burkey. Learning user information interests through the extraction of semantically significant phrases. AAAI 1996 Spring Symposium on Machine Learning in Information Access. AAAI Press, California, 1996.

[8] L. Q. Troyka, J. W. Thweatt. Structured Reading. Prentice Hall, New Jersey, 2003.

[9] M. Ikonomakis, S. Kotsiantis, V. Tampakas. Text Classification Using Machine Learning Techniques. WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, 966-974, 2005.

[10] R. Krovetz. Viewing morphology as an inference process. Proceedings of ACM-SIGIR93, 191-203, 1993.

[11] S. Li, H. Wang, S. Yu, C. Xin. News-Oriented Automatic Chinese Keyword Indexing. Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, 2003.

[12] Turney, D. Peter. Learning Algorithms for Keyphrase Extraction. Information Retrieval, 303-336, 2000.

[13] Y. Yang, J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. Proceedings of the Fourteenth International Conference on Machine Learning, 1997.

[14] Y. Yang. A study of thresholding strategies for text categorization. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001