Eng. & Tech. Journal, Vol.30, No.9, 2012

Finding the Relevance Degree between an English Text and its Title

Dr. Abdul Monem S. Rahma
Computer Sciences Department, University of Technology/ Baghdad
Email: monemrahma@yahoo.com
Dr. Suhad M. Kadhem
Computer Sciences Department, University of Technology/ Baghdad
Email **:**suhad_malalla@yahoo.com
Dr. Alaa Kadhim Farhan
Computer Sciences Department, University of Technology/ Baghdad
Email: dralaa.student@yahoo.com

Received on: 5/10/2011 & Accepted on: 1/3/2012

ABSTRACT

Keywords are useful tools as they give the shorter summary of the document. Keywords are useful for a variety of purposes including summarizing, indexing, labeling, categorization, clustering, and searching, and in this paper we will use keywords in order to find the relevance degree between an English text and its title.

The proposed system solves this problem through simple statistic (Term frequency) and linguistic approaches by extracting the keywords of the title and keywords of the text (with their frequency that appear in the text) and finding the average of title's keywords frequency across the text that represent the relevance degree that required, with depending on a lexicon of a particular field(in this work we choose computer science field). This lexicon is represented using two different B^+ trees one for non-keywords and the other for candidate keywords, these keywords was stored in a manner that prevent redundancy of these terms or even sub-terms to provide efficient memory usage and to minimize the search time.

The proposed system was implemented using Visual Prolog 5.1 and after testing, it proved to be valuable for finding the degree of relevance between a text and its title (from point of view of accuracy and search time).

Keywords: Keyword extraction, Lexicon, Morphology, B^+ tree.

https://doi.org/10.30684/etj.30.9.14
2412-0758/University of Technology-Iraq, Baghdad, Iraq

إيجاد درجة الترابط بين نص انكليزي وعنوانه

الخلاصة

إن الكلمات المفتاحيه أداة مفيدة لأنها تعطي ملخصا قصيرا عن النص. وهي مفيدة لمجالات عديدة كالتلخيص والفهرسة والعنونة والتصنيف وفي هذا البحث سوف نستخدم الكلمات المفتاحيه من اجل معرفة درجة ترابط نص انكليزي بعنوانه.

لقد حل النظام المقترح هذه المشكلة من خلال طرق إحصائية (تكرار التعبير) ولغوية باستخلاص الكلمات المفتاحية للعنوان، والكلمات المفتاحية للنص الانكليزي (مع تكرارهم داخل النص) ومن ثم إيجاد معدل تكرار الكلمات المفتاحية للعنوان داخل النص والذي يمثل درجة تعلق النص بعنوانه، وبالاعتماد على معجم للكلمات الغير مفتاحيه ومعجم للكلمات المرشحة لان تكون كلمات مفتاحية ولمجال معين (في بحثنا هذا تم اختيار مجال علوم الحاسوب). هذا المعجم ممثل بهيكل شجري للكلمات الغير مفتاحية وهيكل شجري آخر للكلمات المفتاحية المرشحة، ولقد خزنت هذه الكلمات المفتاحية بطريقة تمنع التكرار لهذه التعابير او اجزاء منها لتوفير كفائة الخزن ولتقليل وقت البحث.

تم تنفيذ النظام المقترح باستخدام اللغة البرمجية المرئية Visual Prolog 5.1 ولقد اثبت النظام المقترح بعد اختباره بانه قيم في ايجاد درجة الترابط بين نص وعنوانه (من وجهة نظر الدقة ووقت البحث).

INTRODUCTION

Automatic keyword assignment is a research topic that has received less attention than it deserves. Keyword extraction is an important technique for document retrieval, Web page retrieval, document clustering, summarization, text mining, and so on. By extracting appropriate keywords, we can easily choose which document to read to learn the relationship among documents.[1]

The focus of our work is in enabling an ordinary user know the relevance degree between a text and its title. To make this possible, we need to separate this process in three phases: first there is a need to extract the keywords that describe the title, second there is a need to extract the keywords that describe the text with their frequency, and then finding the average of frequency for title's keywords across the text that represent the required relevance degree.

KEYWORD EXTRACTION

Automatic keyword extraction is the task to identify a small set of words, key phrases, keywords, or key segments from a document that can describe the meaning of the document [2]. It should be done systematically and with either minimal or no human intervention, depending on the model. The goal of automatic extraction is to apply the power and speed of computation to the problems of access and discoverability, adding value to information

organization and retrieval without the significant costs and drawbacks associated with human indexers [3].

EXISTING APPROACHES

The manual extraction of keywords is slow, expensive and bristling with mistakes. Therefore, most algorithms and systems to help people perform automatic keyword extraction have been proposed. Existing methods can be divided into four categories: simple statistics, linguistics, machine learning and mixed approaches [3, 4], in our work we will use a mixed between a some linguistic methods with common statistical measures such as term frequency in order to find the relevance degree between an English text and its title for a specific domain (for evaluation we use computer science field to be our domain).

Simple Statistics Approaches

These methods are simple, have limited requirements and don't need the training data. They tend to focus on non-linguistic features of the text such as term frequency, inverse document frequency, and position of a keyword. The statistics information of the words can be used to identify the keywords in the document. Some statistical methods use N-Gram statistical information to automatic index the document [5]. Other statistics methods include word frequency, TF*IDF, word co-occurrences [1], etc. The benefits of purely statistical methods are their ease of use.

Linguistics Approaches

These approaches use the linguistic features of the words, sentences and document. Methods which pay attention to linguistic features such as part-of-speech, syntactic structure and semantic qualities tend to add value, functioning sometimes as filters for bad keywords. Plas *et al.* [6] use for evaluation two lexical resources: the EDR electronic dictionary, and Princeton University's freely available WordNet. Both provide well-populated lexicons including semantic relationships and linking, such as IS-A and PART-OF relations and concept polysemy. During automatic keyword extraction from multipleparty dialogue episodes, the advantages of using the lexical resources are compared to a pure statistical method and relative frequency ratio. Hulth [2] examines a few different methods of incorporating linguistics into keyword extraction. Terms are vetted as keywords based on three features: document frequency (TF), collection frequency (IDF), relative position of its first occurrence in a document and the term's part of speech tag. The results indicate that the use of linguistic features signify the remarkable improvement of the automatic keyword extraction.

Machine Learning Approaches

Keyword extraction can be seen as supervised learning from the examples. The machine learning mechanism works as follows. First a set of training documents is provided to the system, each of which has a range of human-

chosen keywords as well. Then the gained knowledge is applied to find keywords from new documents. The Keyphrase Extraction Algorithm (KEA) [7] uses the machine learning techniques and naive Bayes formula for domain-based extraction of technical keyphrases. Suzuki *et al.* [8] use spoken language processing techniques to extract keywords from radio news, using an encyclopedia and newspaper articles as a guide for relevance.

Mixed Approaches

Other approaches about keyword extraction mainly combine the methods mentioned above or use some heuristic knowledge in the task of keyword extraction, such as the position, length, layout feature of the words, html tags around of the words, etc [9,5]. The task of automatic keywords extraction using combined methods (AKWE) [10] is to extract keywords from the document abstract. This system have three stages: the entered document is firstly pre-processed to remove noisy data, word tagging, and word stemming. Secondly to give candidate keywords, three extracting approaches presented in the proposed system, N-gram approach to extract uni-gram, bi-grams and tri-grams; part-of-speech approach (POS) that extracts phrases which match a set of patterns, and NP-chunk which extract noun phrases.

MORPHOLOGY

If fact there are two types of morphology, one is used for analyzing a word and the other is used for generation a word, and in this paper we dealing with the morphology that can analyse a word (English word). The morphology of English language deal with the changes that may occur through adding affixes to the English words (such as rot will be rotting and give will be giving by adding "ing").

In English language we have two types of affixes: prefixes and suffixes, table(1) show some spelling rules for adding the suffixes to English words, each row in this table shows the suffix and its spelling rule depending on the end of the word. In this table (C) means non vowel letter while (V) means vowel letter, and the symbol (∅) means the last letter must be deleted [11].

Table (1) Some spelling rules for adding some suffixes to the end of the English words

Spelling rule		suffix
Cy	Cie	s
Co	Coe	
e	∅	ed, en
Cy	Ci	ed, er, est
CVuC	CVuCC	
Ce	C	ing
CVuC	CVuCC	
y	i	ly

B^+ Tree[12,13]

B^+ Tree is a structure of nodes linked by pointers is anchored by a special node called the root, and bounded by leaves has a unique path to each leaf, and all paths are equal length stores keys only at leaves, and stores reference values in other, internal, nodes guides key search, via the reference values, from the root to the leaves.

B^+ tree is called an index to database, such that each record will be stored in the database, the reference number (and the key) of that record will be stored in the B^+ tree. So when we want to reach a certain record, we need to know its key to get its reference number from the B^+ tree. When we get the reference number of that record we can retrieve the required record directly. B^+ tree is an arranged and balanced tree (see figure 1), and this is why it is so fast in retrieving the required data.

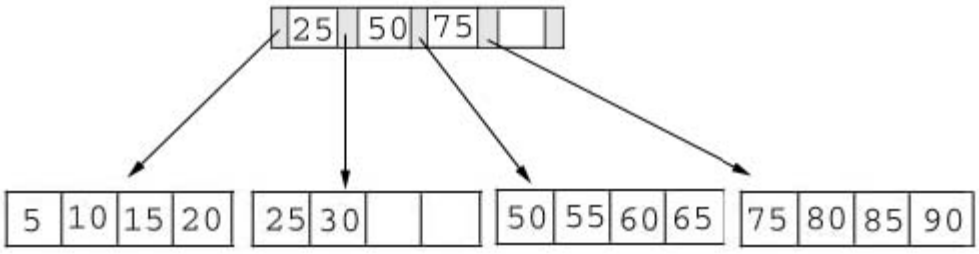

Figure (1): B^+ tree

DESCRIPTION FOR THE PROPOSED METHOD

Basically the proposed method includes three stages:extract keywords of the title, extract keywords from the text (with interaction with the lexicon and morphology) and then finding the average of keyword frequency of the title (after filtering process) across the text to be the relevance degree between the text and its title (see figure 2).

The input to the proposed system will be a title and a text consists of sentences (a sentence is considered to be a set of words separated by a stop mark ".","?" or "!"), and the sentence cutter is responsible on producing these sentences.

The user interface responsible on interaction between the proposed system and the user in ease form (since we use a visual programming language), also the user can update the contents of the lexicon through user interface by removing or adding a new english keyword (or non-keyword) with its information (its suffix, prefix, synonyms and its abbreviation).

Tokenization part of the proposed system is used for converting the title or a sentence (of the text) to a list of tokens.

English morphology is responsible on extract the stem for English word by removing its suffix or prefix and removing the changes that occur during adding these affixes according to the apelling rules of English language.

The other parts of the proposed system will be discussed with more details in the following sections.

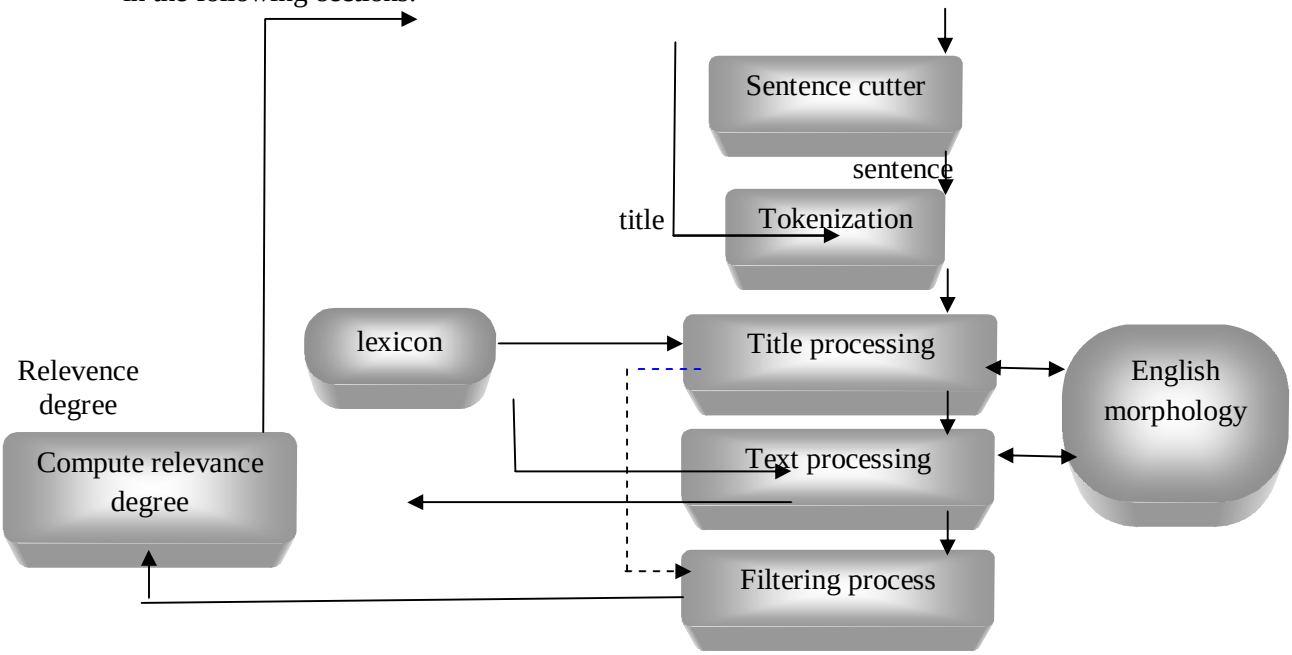

Figure (2): the architecture of the proposed method

Lexicon of the Proposed Method

Lexicon is an important part in any linguistic system, and it is responsible for providing the system with its required information. The lexicon of the proposed method is represented using one database (dbase1) with its index tree (Bt1) for keywords and another database (dbase2) with its index tree (Bt2) for non-keywords.

Non-keywords that stored in the lexicon are: Articles , Conjunctions, Demonstratives , Prepositions, Pronouns , qualities, Main verbs, Auxiliary verbs and Modals. The key of "Bt2" is the stem of the non-keyword.

Keywords that stored in the lexicon are all candidate keywords in a particular domain (in our work we choose the computer science field). The keyword may be one word or a sequence of words, we will store with each keyword its Synonyms and its abbreviation if any. The first word of keyword will be the key for the index tree "Bt1".

The keywords is stored in the lexicon in manner that prevent redundancy to provide efficient memory usage and to minimize search time, in general if one keyword consists of [$word_1$, $word_2$, $word_3$, $word_4$] with logical term P1 (the first word will be the key of Bt1) and another keyword consists of [$word_1$, $word_2$, $word_3$] with logical term P2 then there is no need to restore the second keyword, only P2 need to be added to the dbase1(see figure 3-a), and if another keyword consists of [$word_1$, $word_2$, $word_3$,$word_5$,$word_6$] with logical term P3 that content its required information then only $word_5$ and $word_6$ will be added to the dbase1 with P3(see figure 3- b) .

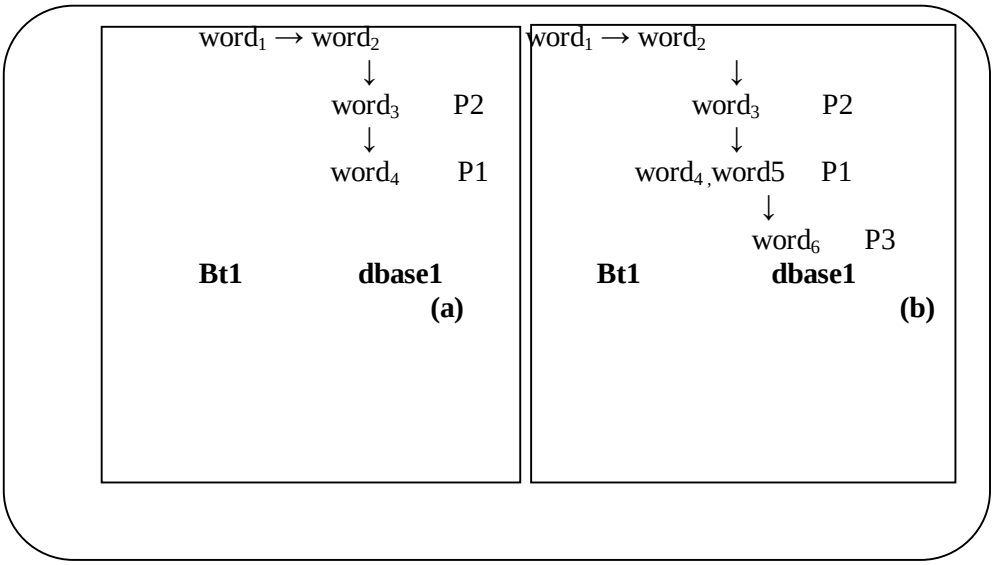

Figure (3) show how redundancy is preventing in the lexicon of the proposed method.

Title processing of the proposed method

In this work, each word in the title will be one of the following (see figure 4):

non-keywords and will be discarded.

- keyword and will be stored in a keyword list.
- Candidate keyword and will be stored in a candidate keyword list.

To check if the current word is not keyword we have either the current word is found in the index tree of non-keywords (Bt2), or the current word is found in the index tree of non-keywords (Bt2) after processing by the morphology to extract the stem of non-keyword , and the same thing happened for keyword if it is consist of one word. If the keyword consists of more than word then we must found the first word of the keyword in the index tree of the keywords (Bt1), and found the sequence of next words in the first term of the logical predicates of the keyword in dbase1, in other words we search for an item with length equal to the length of the keyword, and if not found we need to return back to search for a keyword with length less by one.

If the current word of the title is not found in keyword database and not found in non keyword database then it will be stored in a candidate keyword list, and will be depend on its frequency in the text in order to determine if it is keyword or not.

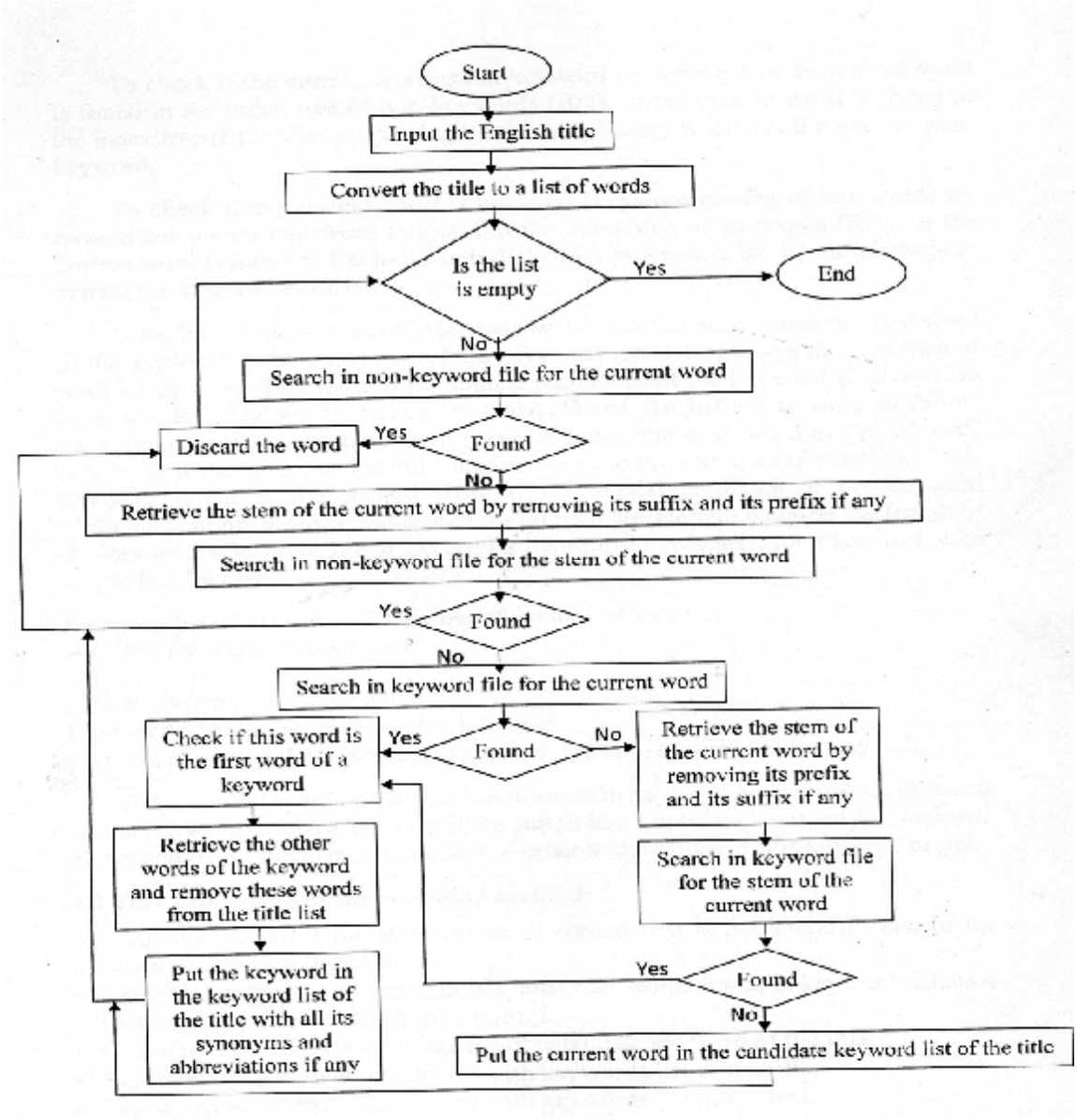

Figure (4) flowchart for title processing of the proposed method

Text processing of the proposed method

All the words of the text will be discarded if it is not found in one of the following cases (see figure 5):

- Keyword, its first word found in Bt1 and other words found in keyword database (before or after morphology processing).
- Candidate keyword found in the list of candidate keyword of the title.
- Synonyms or abbreviations found with keywords list of the title.
- Synonyms or abbreviations found with keywords list of the text.

Only the frequency of above cases will be computed and put it in the list of keywords of the text**.**

Filtering process of the proposed method

In this paper we use a filter function that can remove any un-useful terms from the candidate keywords (filters for bad keywords) depending on their frequency in the text as in algorithm1.

Algorithm1: " filtering "

Input: List0: list of keywords.
List1:list of candidate keywords.
List2: list of keywords found in the text with their frequency.

Output: List: list of keywords.

Process:

Begin

1. **List**=**List0**;
2. Find the average frequency for each keyword in **List0** by using their frequency found in **List2**;
3. Get the minimum average frequency to be **Min**;
4. While **List1** is not empty do
   begin
   4.1. Remove first item from **List1** to be **Term1**;
   4.2. Compute the average frequency (**AV**) for **Term1** by using its frequency found in **List2**;
   4.3. If **AV**>=**Min** then Add **Term1** to **List**;

End; /*while*/

5. Return(**List**);

End.

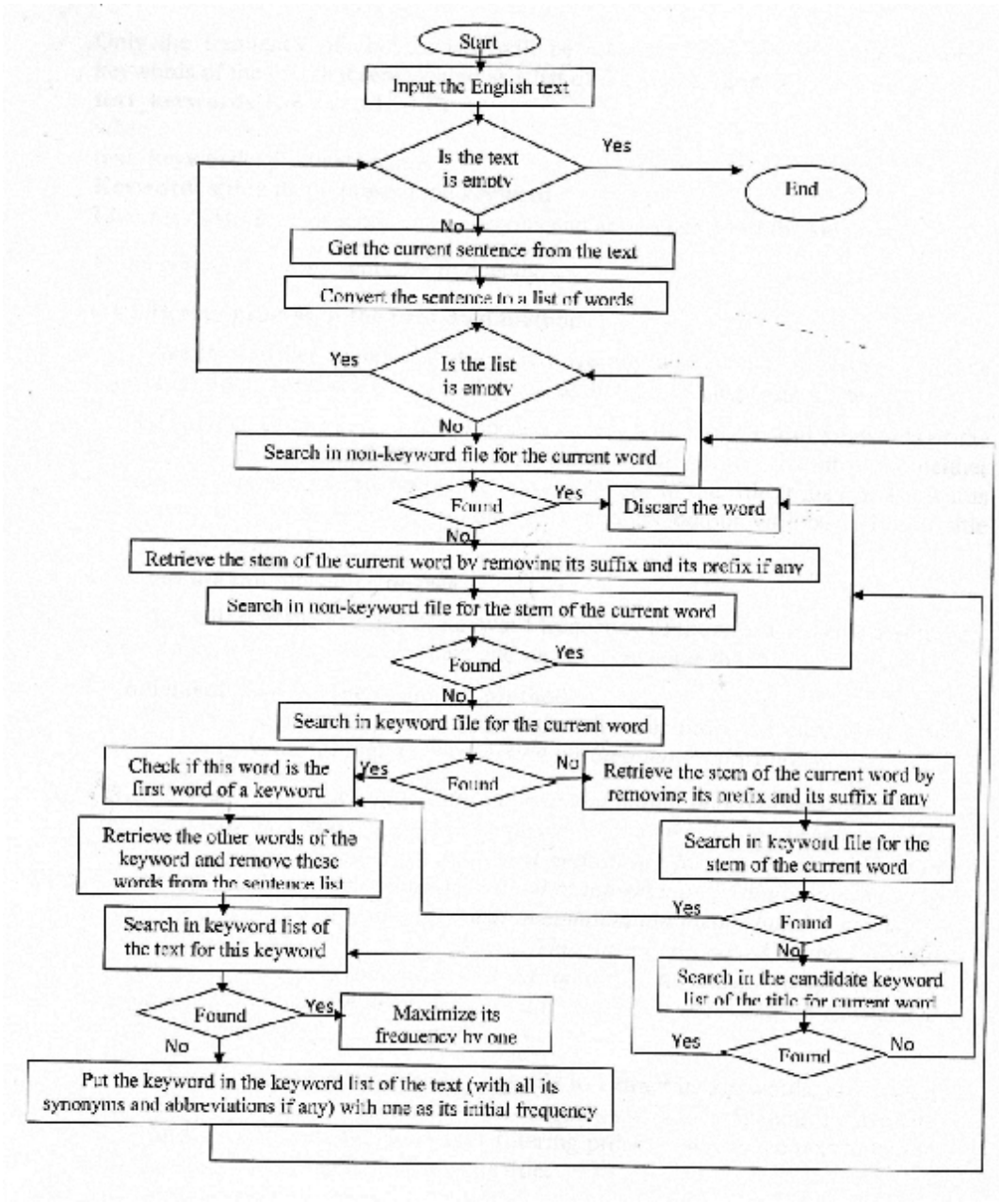

Figure (5) flowchart for text processing of the proposed method

Compute the relevance degree

We will find the average of keywords frequency of the title across the text to be the relevance degree between the text and its title as in algorithm 2.

Algorithm2: "find_relevance_degree"
Input: Title;
Text.
Output: Relevance degree between the text and its title.

Process:
Begin

1. Call *process_title* function that take the **title** as input and return list of its keywords (**List0**) and another list for its candidate keywords(**List1**); /*see figure4*/
2. Call *process_text* function that take the **text** and **List1** as input and return list of its keywords (**List2**) with their frequency; /*see figure 5*/
3. Call *filtering* function that take **List0** and **List1** and return **List** as output; /*see algorithm1*/
4. Compute the summation(**Total**) for the frequency of all keywords appear in **List**2;
5. Compute the summation(**Sum**) for the frequency of **List** keywords that appear in **List**2;
6. Let **N1 =Sum / Total**;
7. Let **N**=**N1** * **100**;
8. Print (**N**);

End.

IMPLEMENTATION FOR THE PROPOSED METHOD

We will take an example in order to describe our proposed method: let the title be: "*Using Natural language processing for Steganography purpose*".
Let the text be: "*Steganography is the technique of hiding information within some format in a way that makes it difficult to detect by one who doesn't know it's there. Steganography has become quite advanced and allows for information hiding in all types of data files. One important method of information hiding is constructing the context free grammar that may found in computation theory but this method is not suitable when the CFG is ambiguous. Thus in our proposed method we will use natural language processing instead of CFG to avoid the problem of ambiguous and unambiguous grammar. NLP is used as abbreviation for natural language processing*".

We have three stages: processing the title to extract its keywords, processing the text to extract its keywords with their frequency and then finding the average of keyword frequency of the title (after filtering process) across the text to be the relevance degree between the text and its title.

Stage1: Title processing

- "using" will be not found in index tree of non-keywords, but after extracting its stem "use" by the morphology and the suffix "ing" the system will found "use" in the non-keyword index tree (Bt2) and after retrieving its logical term the system will found the suffix "ing" with its affixes, so "using" will be discarded as non-keyword.
- "natural" will be found in keyword index tree (Bt1), and after check its stored information and follow the reference for the next words we will have the keyword "natural language processing", and retrieve its synonyms "linquistics" and its appreviation "NLP".
- "for" will be found in the index tree of non_keywords directly and will be discarded.
- "steganography" will be found in the index tree of keywords "Bt1" and will be treat it as keyword, and retrieve its synonyms "information hiding".
- "purpose" will be not found neither in index tree of non-keywords "Bt1" nor in the index tree of keywords"Bt1" and will be regarded as candidate keyword.

Stage2: Text processing

In this case the proposed system will use the same process that used with the title in oreder to find the keywords list of the text with their frequency and discard the other words, so the keywords of the text will be:

keyword	Synonyms and abbreviations	frequency
steganography	Information hiding	5
Context free grammar	CFG	3
Computation theory	Computer theory	1
ambiguous	ambiguation	3
Natural language processing	Linguistic, NLP	3

Since the word of candidate keywords of the title "purpose" is not appears in the text then the output keywords of the title will be the output keywords of the filtering process.

Stage3: compute the relevance degree:

- Compute the summation (Total) for the frequency of all keywords of the text, Total=15.
- Compute the summation (Sum) for the frequency of title keywords that appear in the text, Sum=8.

- Av=8/15.
- Relevance degree=Av*100.

So the relevance degree between the text and its title will be 53%(see figure 6).

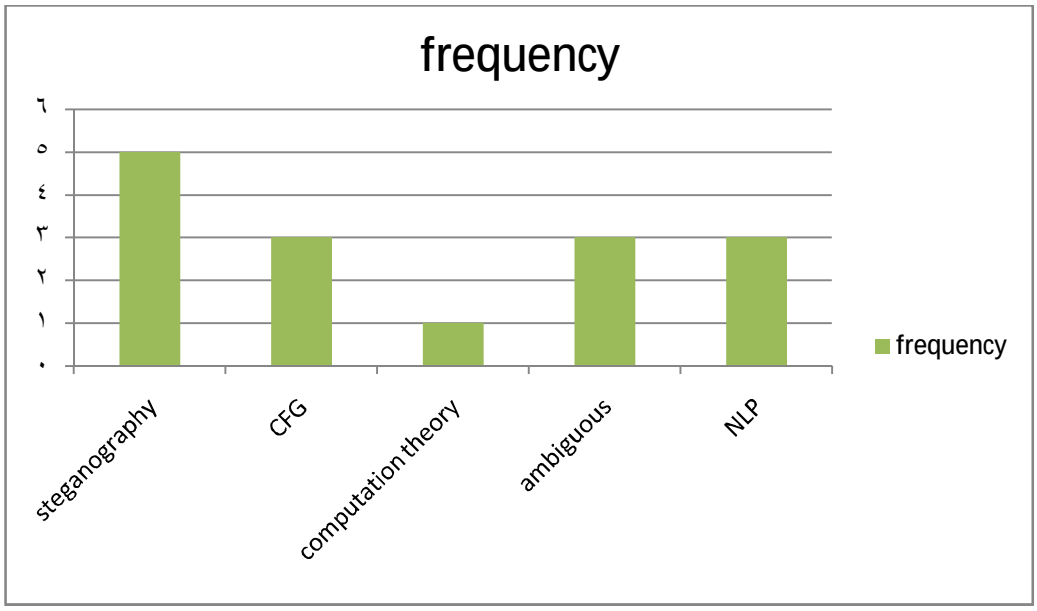

Figure (6) the frequency of keywords of the text

DISCUSSION

For the purpose of evaluation, we compare our approach with frequency-based keyword extraction that can be used to find the relevance degree between a text and its title (see table 2).

Proposed method	Frequency-based keyword extraction
Can deal with any changes occur to the word from adding any affix because of existing of morphology.	Cannot deal with them effectively.
Synonyms (and abbreviations and all the words with same meaning) problem was solved.	Cannot Deal with this problem.
Provide an efficient search time for gating the relevance degree because of using two B^+ tree one for keywords and other for non keyword.	Search time for gating the relevance degree between a text and its title will depend on text length.
Using a particular domain will help us take all the special situations in our account.	Cannot deal with any special situations because it is independent on a special domain.
our method can extract keywords even if they do not appear frequently	Depend on term frequency to extract the keyword
our method can detects keywords consisting of one, two or more words.	Can deal with keywords with one word only.
dependent on the language and the domain.	completely independent of language or domain.
High accuracy sine it depend on a lexicon and analyzing	More easy to use since it do not need any expensive resources but with less accuracy

Table (2) compare between Frequency-based keyword extraction and the proposed method for finding the degree of relevance between text and its title.

CONCLUSIONS

In this paper the following points can be concluded:

- Using a mix between the simple statistic approach(such as finding the average of title's keywords frequency across the text) and linguistic approach (such as morphology and the lexicon) will provide a high accuracy for relevance degree between a text and its title.
- Computing the average for title's keywords frequency across the text will provide a high accuracy for relevance degree between a text and its title.
- Prevent the redundancy of the keywords in the lexicon or even the sub keywords will provide efficient memory usage.
- Using B^+ tree for representing the lexicon for keywords and non keywords that may found in a particular field will provide an efficient search time in finding the relevance degree between a text and its title.
- We solved the problem of Synonyms and abbreviations or keywords with the same meaning by storing with each keyword all its Synonyms in the lexicon.
- Using a lexicon depend on stems of English words will provide efficient memory usage.

- Determining a special domain and a special language make the system take the special situations in its account.
- As more electronic documents become available, we believe our method will be useful in many applications, especially for domain-dependent keyword extraction.

REFERENCES

[1] Matsuo, Y. M. Ishizuka, "keyword extraction from a single document using word co-occurrence statistical information", international journal on artificial intelligence tools, vol. 13, no. 1, world scientific publishing company, 157-169,2004.

[2] Hulth, A. "Improved automatic keyword extraction given more linguistic knowledge", In Proceedings of the 2003 Conference on Emprical Methods in Natural Language Processing, Sapporo, Japan, 2003.

[3] Michael J. Giarlo, "A comparative analysis of keyword extraction techniques". Rutgers, The State University of New Jersey, 2005.

[4] Chengzhi Zhang, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, Bo Wang, "Automatic Keyword Extraction from Documents Using Conditional Random Fields", Journal of Computational Information Systems, 2008.

[5] Iryna Oelze, "Automatic Keyword Extraction for Database Search",2009.

[6] L. Plas, V.Pallotta, M.Rajman, H.Ghorbel, "Automatic keyword extraction from spoken text", A comparison of two lexical resources: the EDR and WordNet. Proceedings of the 4th International Language Resources and Evaluation, European Language Resource Association, 2004.

[7] Witten, I. G. Paynte, E. Frank, C. Gutwin, C. Nevill-Manning. KEA: practical automatic keyphrase extraction. In Proceedings of the 4th ACM Conference on Digital Library, 1999.

[8] Suzuki, Y. F. Fukumoto, Y. Sekiguchi, "Keyword extraction of radio news using term weighting with an encyclopedia and newspaper articles", SIGIR, 1999.

[9] Keith، J. B. Humphreys, Phraserate:An HTML keyphrase extractor, Technical Report, 2002.

[10] R. Abdul-Rahman, "Automatic Keywords Extraction Using Combined Methods", Ph.D. thesis, Technology University, 2006.

[11] Winograd,T.,"language as A Cognitive Process: v1, Syntax", Addison-Wisely, 1983.

[12] Goetz Graefe ,"B-tree indexes, interpolation search, and skew", Chicago Illinois, USA, 2006.

[13] Anderson, Susan, "B+ Trees", Freed, 2005.