社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ＦＩ－90 (7)
2008－ＤＤ－65 (7)
2008／3／28

Using English for Queries: An Approach to Implementing an Intelligent Web Search

Vitaly Klyuev
University of Aizu, Aizu-Wakamatsu

The retrieval efficiency of the presently used search tools cannot be significantly improved: A "bag of words" interpretation causes loosing semantics of texts. The functional approach to present English texts in the memory of computers makes it possible to keep semantic relations between words and use ordinary English sentences as queries. The prototype of the system utilizing this approach is presented.

質問のための英語使用:
インテリジエントウエブ検索エンジン実装へのアプローチ

ヴイタリー　クリユーエフ
会津大学ソフトウエア工学研究室

現在の検索ツールが使用している手法では、これ以上劇的な検索効率の改善を期待することはできない。つまり、「単語のつまった袋」的な解釈による現在の検索ツールの手法は、文章が本来持っている意味を失う原因となっている。しかしながら、コンピューターのメモリーに存在する英文章に対して、機能的な観点からアプローチすれば、文章の意味部分を失うことなく、元来のキーワード検索と組み合わせて検索できる。すなわち、質問の意味を理解した検索エンジンが実装可能ということである。この手法を利用したシステムのプロトタイプを今回発表する。

1. Introduction

Nowadays, the Internet is the major source of information for millions of people all round the world. For many users, the Internet has supplanted TV and newspapers. Searching is the most common task performed on the Web by the end user. There are many general purpose and topic-oriented search tools available on the net but finding appropriate information is still difficult and searching remains the most frustrating task.

The key reasons for such a situation can be characterized as follows. Texts are considered as "bags of words" by search engines. Such a view resulted in two main outcomes:

- Search engines do not take into account semantics of text documents when perform indexing and searching. In other words, they simply loose semantics of documents.
- The language to express queries is artificial: a set of key words. Its semantics is very far from the semantics of any natural language.

Users have been taught to utilize keywords as a query language since the beginning of the search engines era. Many on-line instructions suggest how to choose keywords for queries but selecting correct words to specify the information needs is not an easy task for ordinary users: Statistical analysis of user behaviors showed that queries are short (2 to 3 terms on average) and ambiguous, and users rarely look beyond the first 10 – 20 links retrieved [1].

The logical outcome from the aforementioned representation of text documents is: Statistical methods dominate in the area. A large portion of research to develop new techniques is oriented towards testing different heuristics. The most popular techniques utilized by search engines' developers include a) vector space model, b) Boolean model, and c) probabilistic model. Their comprehensive description can be found in [2]. To improve the quality of the search different techniques were proposed. Among techniques utilized by search engines to solve the polysemy problem, we point out statistical analysis of the user profile data and query expansion utilizing dictionaries of synonyms. These techniques are in common use [3, 4]. The Google approach adopted the idea of the citation index widely used in the scientific world to detect importance of publications. This solution helped to improve the quality of search results and made Google the most successful search engine [5].

The retrieval efficiency of the presently used systems cannot be significantly improved: "Bag of words" interpretation causes loosing semantics of texts.

One of the promising solutions to make the Web search more intelligent is to implement the functional interpretation of texts in natural languages. Such a view helps to preserve semantic relations between words. These relations can be taken into account when indexing documents and when performing searching. Utilizing this approach, it is possible to use a natural language to express user queries. In many cases, this way is more usual for users to describe their information needs compared to the keyword style.

In this study, we discuss an improved model of Web service and the architecture of the prototype of the retrieval system based on the functional approach.

2. A Functional Model

The functional approach was initially proposed for Russian language [6]. Its main feature is that it uses a functional representation of the sentence meaning. The key point of this view is as follows: Each word of the natural language (in our case English) is the name of a function $f(x_1, x_2, ..., x_n)$ connected with this word and called its semantics. Most of the words in the natural language have several meanings. This fact is well known. The word obtains one of its particular meanings after assigning values to the particular arguments.

This concept has an analogy in programming languages: Overloading involves providing several methods with the same name, but having different parameter lists. The meaning of the word is calculated when the function f is running. The sentence represents a single complete superposition of word functions. The same situation can be seen in programming languages: The meaning of the program is calculated when the program is running. There is a direct analogy between sentences in a natural language and programs in programming languages such as Java, C, etc.

3. Improved Web Search Service

Nowadays, general purpose search engines such as Google, Yahoo!, Ask, etc. have huge databases of crawled documents. Usually, relevant documents can be found in the returned lists in response to user queries. Precision is relatively low. The simple and practical approach is to use the power of popular search engines implementing intelligent Web service as a front end part and general purpose search engines as a back end part. It aim is to filter irrelevant documents.

The query expressed in a natural language (English in our case) is submitted to the general purpose search engine in the usual style: as a set of keywords. Software responsible for intelligent retrieval can be installed on the client computer or on the computer of the Internet Service Provider. Functions of this part are as follows:

- Get the ranked list of returned URLs from the search engine
- Download up to 100 documents referenced by URLs from the beginning of the list

- Index them utilizing the functional approach
- Submit the initial query as a sentence (sentences, questions in English) to the dataset created at the previous step
- Retrieve the documents from it utilizing the functional approach
- Present the results to the end user
- Destroy the dataset of locally indexed documents

Figure 1 gives an illustration of the schema described. This solution will increase the response time of the computer to the user query. On the other hand, the quality of the search will be improved.

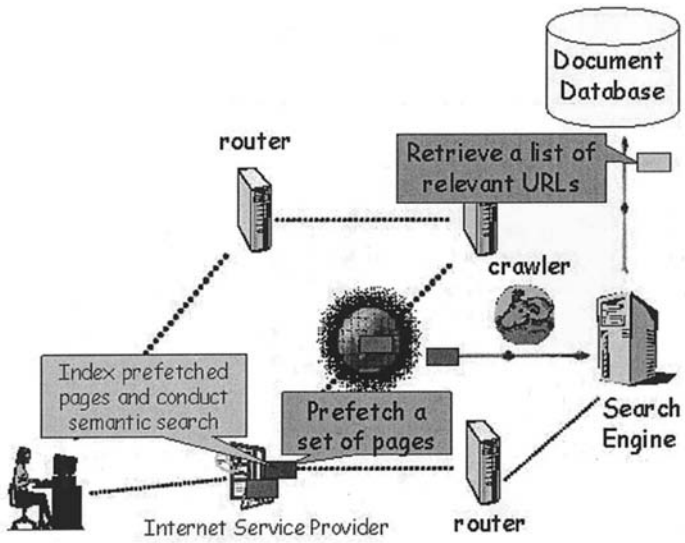

Fig. 1 Architecture of intelligent Web search service

4. System Architecture

The architecture of the prototype is presented in Figure 2. The prototype consists of three major parts: a) a morphological analyzer, b) a parser, and c) a semantic database. A role of the morphological analyzer is to define the part of speech of each word in the sentences and some characteristics such as the tense for verb groups, the plural or singular form for nouns. These characteristics are the key for correct parsing utilizing the semantic database. Records of this database include the semantic definitions of the words.

The example of a dictionary entry in Figure 2 is presented in the simplified form to illustrate how meanings of words are determined when a sentence is parsing. From Table 1 you can see that parsing the first and the second sentences is done in accordance for the first alternative of the definition of the verb "steal". The functional representation for the third sentence is done in accordance with the second alternative of the verb definition. The key elements in this selection are the type of words and the number of words (arguments). For the first sentence the word "from" put the parsing to select the first alternative. The second sentence failed to be parsed successfully for all alternatives except for the first one. For the third sentence, the presence of the adverb "up" moved parsing to the second alternative.

Tab. 1 Parsing sentences

Sentence	Part of speech tagging	Functional representation
Leroy stole the money from clients' account.	Leroy(NNP) steal(verb, past) the(DT) money(NN) from(IN) clients(JJ) account(NN)	Steal (verb, past; Leroy(noun, sbd), money(noun, sth; the(det)), from, account(noun, sth; client(noun, plural, sb))
Leroy stole the money.	Leroy(NNP) steal(verb, past) the(DT) money(NN)	Steal (verb, past; Leroy(noun, sbd), money(noun, sth; the(det)))
Leroy stole up the hall.	Leroy(NNP) steal(verb, past) up(adv) the(DT) hall(NN)	Steal (verb, past; Leroy(noun, sbd), up(adv), hall(noun, sth; the(det)))

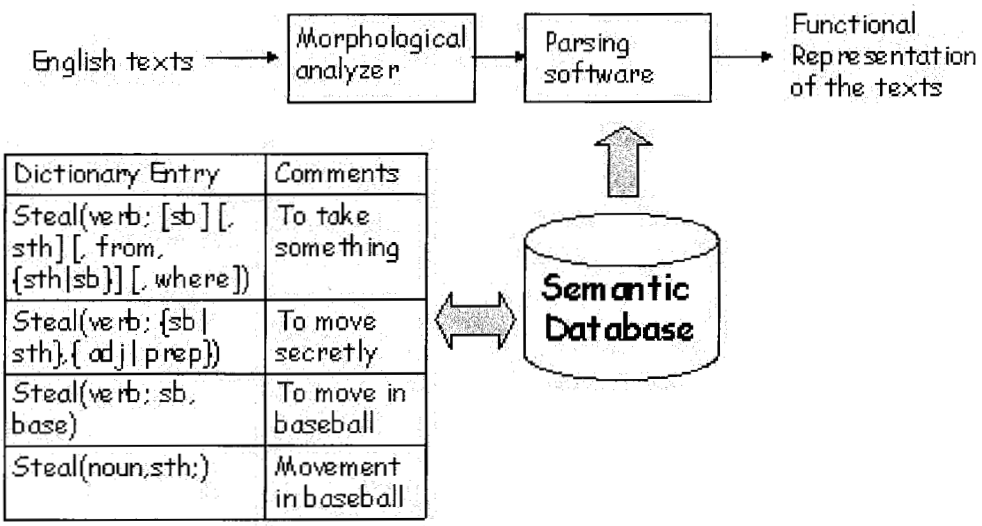

Fig. 2 Architecture of the prototype

5. Transformation of Documents and Queries into Functional Representation

The main idea of the algorithm to convert sentences into the functional form is as follows:

- In each sentence the verb is a key. At the beginning, the algorithm is matching the verb with one of the alternatives to assign the parameters. If it is successful doing the selection in the way of one by one, then the superposition is created. If there is no success, then it is wrong English sentence (our solution does not consider such sentences and does not try to recover them). The algorithm tries to find the closest variant adding, deleting, replacing words (preposition, fixed argument for the verb, etc)
- For words he, she, they, etc. (pronouns) the following transition rules are applied: the text is looked back until the corresponding name or noun is found. The first candidate is the possible value for the pronoun. It is put into the set of the possible values.
    - For example, the initial text is as follows "*John and Mary entered the house. John looked at the enemy. He shot him twice.*" It will be transformed according

to the rules specified to the form: *"John and Mary entered the house. John looked at the enemy. John shot the enemy twice."*

- Documents are indexed by verbs in the hash table.

A query transformation

- A query is converted in the same way as a document (One sentence with the possible question mark at the end).
- Synonyms of the verb and the words are applied to create a set of "semantically" related superpositions utilizing the thesaurus and the collocation dictionary.
- The set of queries generated at the previous step is used to search. Search can be done in parallel.

6. A Structure of the Dictionary Entry

To create the semantic dictionary, we work with electronic versions of different English dictionaries [7 - 12]. A structure of the dictionary entry is presented below:

<word definition> :: <alternative> { <alternative>}*
<alternative> :: (<part of speech>; <list of arguments>; <preceding word>; <following word>; <list of synonyms>)
<list of arguments> :: [<argument>]{,<argument>}*
<argument> :: <part of speech> | <specific word>
<preceding word> :: <argument>
<following word> :: <argument>
<list of synonyms> :: <specific word>{<specific word>}*

Each <alternative> represents the separate meaning of the word. For example, according to the Oxford Advanced Learner's Dictionary of English [7], the word steal has four different meanings. The entry for this word in our semantic dictionary has four alternatives. Each idiom or phrasal verb has is mapped to the corresponding alternative. <Preceding word> and <following word> components specify words which can precede and follow the defined in the entry word. The <specific word> component is for the concrete value (word). Lists of synonyms are taken from [9]. They are crucial when the query is transformed to the functional representation.

7. Discussions

Our prototype is at the alpha testing stage. To simulate the work of the ISP server, we use a powerful computer Dell Precision T7400. Its specifications are as follows: two Quad Core Intel® Xeon® Processor E5405 (2.00GHz,2X6M L2,1333), 4GB memory, and 500GB SATA HDD. The natural language processing software, we have applied is PetaMem Language Server[14] and the Machinese Syntax and Machinese Phrase Tagger for English [15]. To move to experiments with real data, we need to create the dictionary for the *Oxford 3000 Wordlist* [13]. For ambiguous queries our prototype will retrieve garbage. Please have a look at the example:

Document 1: *"John and Mary entered the house. John looked at the enemy. He shot him twice."*
Document 2: *"Mark put down the receiver. He shot the several picture of his enemy."*
Query 1: *Who fired?*
Query 2: *Who fired the enemy?*

In response to the first query the prototype retrieves two documents. In response to the second query only the first document will be recognized as relevant. The meanings of the word shoot such as "fire a bullet" and "take a picture" are ambiguous from the point of view of our model. When the

system analyses only the sentence: *"He shot the several picture of his enemy"*, it is not clear which action has been done.

Searching algorithms utilizing the functional approach are presented in [16].

8. Conclusion

In this paper, we propose the architecture of the retrieval system based on the functional approach to natural languages (English). The basic mechanisms applied to the prototype are discussed. To implement this approach, we need a new model of Web service. This model is examined as well.

The main advantage of our solution is in the use of a natural language (English) to express user queries. In many cases, this way of query representation is much easier for users to describe their information needs compared to the traditional keyword style.

Our prototype acts as a filter for a general purpose search engine. It discards the large portion of garbage before presenting results to the user.

9. Acknowledgement

This study has been performed as part of the project "Intelligent Tool to Search the Web Using a Natural Language" funded by the Japan Science and Technology Agency, 2007.

References

[1] Monika R. Henzinger, Hannes Marais, Michael Moricz, and Craig Silvestein. Analysis of a very large Alta Vista query log. Technical Report 1998-014. Digital SRC, October 1998.

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008, On-line version: http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html [Accessed in March 2008].

[3] Shanmukha Rao, B. Rao, S.V. Sajith, G.. A user-profile assisted meta search engine. Proc. of the TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region.

[4] Google search engine. http://www.google.com [Accessed in March 2008].

[5] Amy N. Langville and Carl D. Meyer. Google's PageRank and Beyond: The Science of Search engine Rankings, Princeton University Press, 2006.

[6] Vitaly Tuzov. Computer Semantics of Russian, Saint Petersburg State University Press, 2004 (in Russian).

[7] Oxford Advanced Learner's Dictionary, 7ed, Oxford University Press, 2005.

[8] Oxford Dictionary of English 2nd Edition, Oxford University Press, 2003.

[9] Oxford Thesaurus of English, Oxford University Press, 2004.

[10] Oxford Collocation Dictionary for Students of English, Oxford University Press, 2002.

[11] Oxford Phrasal Verbs Dictionary for Learners of English, Oxford University Press, 2001.

[12] Collins COBUILD Advanced Learner's English Dictionary, new edition, HarperCollins Publishers, 2004.

[13] Oxford 3000 Wordlist: http://www.oup.com/elt/catalogue/teachersites/oald7/oxford_3000/oxford_3000_list?cc=gb [Accessed in March 2008].

[14] PetaMem. http://www.petamem.com/ [Accessed in March 2008].

[15] Connexor. http://www.connexor.com/m_semantics.html [Accessed in March 2008].

[16] Vitaly Klyuev and Vladimir Oleshchuk, Semantic Retrieval of Text Documents, 7th IEEE International Conference on Computer and Information Technology (CIT 2007), pp. 189-193, 2007.