Semantic Clustering Approach Based Multi-agent System for Information Retrieval on Web

Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy A. Essa

Faculty of Computing and Information Technology
King AbdulAziz University
Jeddah, KSA

Summary
Document clustering is an important technology which helps users to organize the large amount of online information, especially after the rapid growth of the Web. This paper focuses on semantic document clustering method and its application in search engine. We proposed a multi-agent based information retrieval system to enhance the search process. The agents retrieve the results of Web search engine and organize the results by clustering them into different categories for a given query. We utilized WordNet ontology and several approaches to cluster results in appropriate category according to WordNet synsets. The experiment shows that semantic clustering work better than original clustering.

Key words:
Document Clustering, WordNet, Semantic Clustering, information retrieval, Multi-Agent System

1. Introduction

Information Retrieval plays an important role in our daily life and its largest role is observed in search engines. Most users rely on Web search engines to look for specific information from the Web. These search engines often return a long list of search results that would be ranked by their relevance to the given query. Web users have to go through the long list and inspect the titles, and snippets sequentially to recognize the required results. Filtering the search engines' results consumes the users' effort and time especially when multiple sub-topics of the given query are mixed together [1]. This problem raise because the current search engines make little effort to understand users' queries and they use traditional techniques relay on matching terms and phrases; the search engine return a page just when the page includes the target keywords. Also, Web users sometime sent very short queries (only one word) and often it has multiple meanings that make a task of finding relevant information from the Web using just few words is a tremendous challenge.

Manuscript received January 5, 2012
Manuscript revised January 20, 2012

Search results' clustering attempts to solve this problem by automatic organizing a list of search result returned by a search engine into a set of meaningful thematic categories. The available clustering search engines tend to use string matching, rather than true linguistic analysis, to identify keywords and phrases which documents share in common. Then they generate clusters based on these shared keyword and phrases [2]. In this work, the words are semantically analyzed using WordNet to produce precise analysis of how the documents might relate to each other.
This paper describes how to overcome some of the major limitations in the current search engines. We proposed a multi-agent based information retrieval system to enhance the search process. We used different types of agents each of them has its own responsibility. We organize the results of Web search engine by clustering them into different categories for a given query. We utilized WordNet ontology and several approaches to cluster results in appropriate category according to WordNet synsets.
The paper is organized as follows. Some related works are introduced in Section 2. The basic concepts are defined in Section 3. In Section 4, we describe the proposed system Architecture. The implantation of the proposed system is presented in Section 5. Section 6, we describe the results from the applied clustering technique and its evaluations, which associated with examples. Finally we conclude the paper and give some future works in Section 7.

2. Related work

One of the very well-known approaches for query ambiguity is search result clustering. The basic idea of this approach is to cluster a given query with the list of snippets returned from search engines based on some measures of similarity. Algorithms for clustering Web search result have been reported in many papers such as [3, 4, 5]. One of the pioneer works in this respect is the Scatter/Gather project [3], but this project has some limitations because it used a traditional heuristic clustering algorithms. Y. Wang et al. [4] have proposed a clustering Web search results

algorithm based on hyperlinks. It is mandatory in the hyperlinks algorithm to download the original Web pages that make the clustering so slow. Lingo algorithm, proposed by Stanislaw Osiński and Dawid Weiss [5], uses frequent phrases to identify candidate cluster labels, and then assigns snippets to these labels.

However, all of the pervious algorithms perform clustering based on the syntactical similarity but not semantic similarity. Actually, there is some research on using clustering for a different search applications but only little work has been done on using semantic clustering for organizing search results. Some of the Techniques for clustering and categorizing Web documents using WordNet or other ontologies have also been extensively studied [6, 7, 8]. Hotho, Staab and Stumme study [6] showed that enhancing the bag of words with WordNet synsets from the words in the text and their hypernyms (up to a certain distance) does make better clusters than a plain bag of words representation. As a follow up, Sedding and Kazakov [7] showed that using a more precise word sense disambiguator one can obtain even better results than the results by Hotho. Jing et al. [8] uses the same technique as Hotho et al. and enhances it by computing a word similarity measure based on what they call “mutual information” over their clustering corpus. However, their technique did not produce any considerable improvement over Hotho et al.’s baseline.

Our approach to cluster Web search results is an extension to the Lingo algorithm by adding semantic recognition to the frequent phrase extraction phase. This is achieved by finding the synonyms of frequent words in the WordNet database, and adding the synonyms to the pool of frequent terms that comprise the cluster label candidates.

3. Basic concept

3.1 Multi Agent Based system

There are several definitions of agents, given by different researchers, each involving the most valuable characteristics related to their context. We believe that the concept of agent can be summed up by the following definition:

"An agent is a software program that can perform specific task for a user and processes a degree of intelligence that permits it to perform parts of its tasks autonomously and to interact with its environment in a useful manner". [9]

There are different agents' types. We will describe Information agent which is the agent that can be used to enhance the search service. An information agent is an agent that can access at least one and many information sources, and is able to collate and manipulate information obtained from these sources in order to answer users' queries. The information agent should be capable to perform the following tasks: locate information sources, extract information from the sources and prepare and present the results in an appropriate form [10].

Multi-agent system (MAS) is a system composed of multiple interacting intelligent agents that can be considered as a loosely coupled network of problem solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity. Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent or a traditional system to solve [11]. More recently, the term multi-agent system has been given a more general meaning, and it is now used for all types of systems composed of multiple autonomous components showing the following characteristics [12]:

- Limitation in solving a problem individually.
- No global system control.
- Data decentralization.
- Asynchronous computation.
- Capability to cooperate or to compete.

Our system composed of several agents; each of them has its own reasonability and will be explained in section5.

3.2 WordNet

WordNet ontology is one of the most important resources available to researchers in the field of text analysis, computational linguistics, and many others related areas. WordNet [13] is ontology of lexical references whose design was inspired by the current theories of human linguistic memory. Nouns, verbs, adjectives and adverbs are grouped into sets of synonyms (synsets), each representing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations such as hypernym /hyponym (is.a), and meronym/holonym (part.whole).

The WordNet purpose to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications. WordNet is used in many text classification methods as well as in Information Retrieval (IR) because of its broad scale and free availability.

3.3 Clustering Result Algorithms

We use Results Clustering to dramatically reduce search time and effort, where search results clustering organize the search results into topics, fully automatically and without external knowledge. Designing a web search clustering algorithm is a big challenge because we have to ensure that both content and description (labels) of the resulting groups are meaningful to humans.

Most of the open text clustering algorithms follows a scheme where content clustering is performed based on the snippet, accordingly the labels are identified. This process does not focus on the cluster label where readable and unambiguous labels of the thematic groups are an important factor of the overall quality of clustering.
To avoid such problems Lingo clustering algorithm [14] reverses clustering process it attempts to ensure creating human-perceivable cluster label and then assigning documents to the cluster. Specifically, first it identifies cluster labels and only then assigns documents to the labels to form final clusters. To find the labels, Lingo builds a term-document matrix for all input documents and decomposes the matrix to obtain a number of base vectors that well approximate the matrix in a low-dimensional space. Each such vector gives rise to one cluster label. To complete the clustering process, each label is assigned documents that contain the label's words. To improve the quality of the clustering, we extend Lingo algorithm by adding semantic recognition to label identification phase.

4. Architecture of Proposed system

WordNet, Information Retrieval System (IRS) and phrase based document clustering are combined together to formulate the framework of our work. The architecture diagram of our proposed multi-agent system is shown in figure 1. The functionality of the proposed system has been partitioned into six main processes, presented in the following sections:

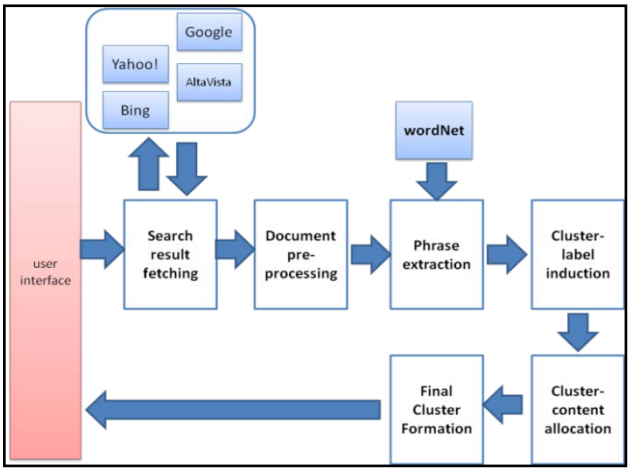

Fig 1. General Architecture of a proposed system

4.1 Search result fetching

In Phase One, variants of the input query are optimized and submitted using the mobile agents, and each retrieves up to 100 results. We must then perform a re-ranking on the results as a whole, based on some relevance measures. We assume the returned result is informative enough because most of the search engines are designed to facilitate user's relevance judgment only by the title and snippet. Then the extracted result items are used as input to next process.

4.2 Preprocessing

Pre-processing is selecting the most suitable terms that describe better content. At this stage, we typically use a combination of three common text-preprocessing methods:

- Lemmatization: Words with the same meaning appear in various morphological forms. A lemmatizer converts a word to its normalized form, called a lemma. For instance, a lemmatizer results for, compute, computing, and computed have the same lemma compute, whereas computer and computers have the same lemma computer. Lemmatizer requires a part-of-speech tagging which assigns a part of speech label to each word in a text depending on the labels assigned to the preceding words.
- Stop word Removal: Stop words are very common words that do not convey any meaning i.e. "the, a, of, for, in, ...” In this process, we are leaving them in the input because they can help users understand the meaning of longer phrases.
- Tokenization: Tokenize process is used to determine sentence boundaries, and to separate the text into a stream of individual tokens (words) by removing extraneous punctuation. It separates the text into words by using spaces, line breaks, and other word terminators in the language. Document texts must be tokenized correctly in order for the noun phrase extractor to parse the text efficiently. It also works as a basis for phrase boundary detection.

4.3 Phrase extraction

Frequent terms and phrases are extracted according to certain criteria. We are using a term frequency threshold to determine the minimum number of times a term should appear in snippet to be considered a frequent term. Cluster label should not be cross sentence boundaries and also should give a complete meaning. The Complete phrases should allow clearer cluster descriptions than partial phrases (for example, “King Abdul-Aziz” versus “King Abdul-Aziz University”).
The modification carried out for this paper adds an extra step that involves adding the query synsets to the document input. To make this addition valuable we increase the term frequency.

4.4 Cluster-label induction

We counteract a term-document matrix of the frequent, and we calculate the weight for each term using the standard tf-idf formula [15]. We applied Singular Value Decomposition (SVD) method [16] to help in identifying semantic concepts that link the documents together.

4.5 Cluster-content allocation

In this phase, the input snippets are assigned to the selected clusters-labels that determined from the previous process. In this process, the input snippets match against a single cluster label if the similarity between a snippet and the label exceeds a predefined threshold, then we allocate the snippet to the corresponding cluster. We can also use the similarity values to sort snippets within their groups, making the most relevant snippets easier to identify. Finally, we created an "other topics" group for those snippets that do not match any of the cluster labels.

4.6 Final Cluster Formation

Clusters are sorted according to a ranking function, and the top clusters are displayed to the user.

5. Implementation of Proposed system

The implementation of our system is based on the Carrot2 framework developed by Dawid Weiss [17]. Carrot2 is an open-source environment that facilitates experiments with processing and visualization of web search results. Carrot's architecture is based on a set of distributed components that cooperate with each other by exchanging XML data.
We change the carrot framework to agent based system. In a multi-agent based system there are multiple agents cooperate together in order to perform a specific task. Our system has the following agents:
Interface agent: This agent provides a graphical interface that links the user with other agents. The major tasks of the interface agent are to:

- Accept a natural language query from the user.
- Cooperate with search agents and send the accepted query.
- Cooperate with a main agent in order to receive results.
- Display the results to the user.

Search agent: This is a mobile agent search in different search engine's servers .The major tasks performed by this agent are:

- Accept a natural language query from interface agent
- Search in the main search engine and return the top 100 result from each search engines.
- Send the result to the Main Agent.

Main Agent: this agent can be considered as administration and cooperation agent. The major tasks obtained by the main agent are:

- Receive the result from the search agent.
- Filter the result and remove the duplicated document then send the result to the cluster Agent.
- Forward the results to the interface agent, when the cluster agent returns with the results.

Cluster Agent: perform the clustering process that we explained the earlier then sent the result to the main agent.

6. Experment

We conduct the experiments as following. We use 10 queries of three different types, some of them were proposed in [18], those are listed in table 1.

Table 1: sample of the queries and query types used in the evaluation.

Type	Queries
Ambiguous queries	apple, NLP, Pluto
Entery names	dell, disney, world war 2
General terms	health, flower, music
Complex queries	clustering search results

6.1 Clustering quality

Evaluating the precision and recall of the document clustering system is not a well defined task because there is no single ordering the results. There are many possible approaches to this evaluation. If the cluster were perfectly labeled, so that the user always chose the most relevant cluster, then it would be sufficient to evaluate the precision clusters and their labels. Clustered results are judged by human subject with main criteria, whether a cluster's title is meaningful and agreeable with majority of inside results. We compute precision of cluster title before and after adding the semantic.
Usually, users only spend time on the top k results. In this experiment, only the first one hundred results are retrieved and clustered. Following are average results we get from above queries:

Table 2: Shows result of the precision for each cluster

query types	precision of cluster label	precision of semantic cluster label
Ambiguous queries	0.84	0.91
Entity names	0.80	0.93
General terms	0.86	0.92
Complex queries	0.83	0.90
Average	0.85	0.92

According to the table 2 and figure 2 the cluster quality was improve by about 6.39%.

6.2 Performance

The system runs on a machine with CPU Intel Core i5 2.8 GHz, 4 GB RAM, and 520 GB Hard Disk. The average execution time for traditional clustering of all 100 results is 1085 ms seconds. The average execution time for semantic clustering of all 100 results is 1665 ms seconds. So, the performance decreased when we add the semantic because the system spends more time when semantic clustering is applied.

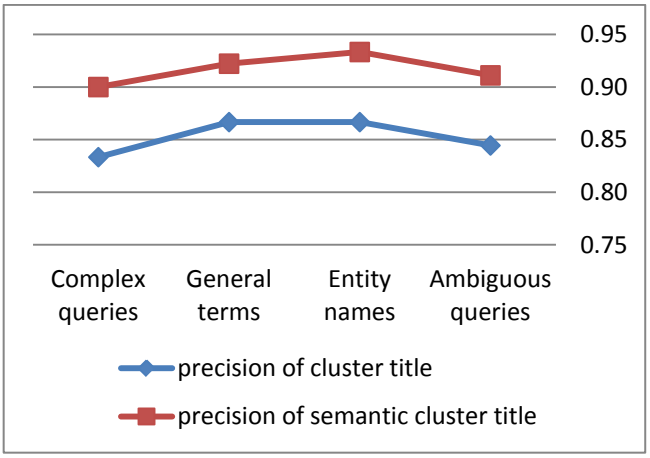

Fig 2. Precision of traditional cluster Precision of semantic cluster for different query types.

7. Conclusion and future work

In this paper, we investigated the problem of how to cluster the search result from search engines. Queries are often ambiguous because many words have multiple meanings. By clustering the search results based on the semantic of the query term, it makes it easier for users to identify relevant results from the retrieved results. We proposed a modified version of the lingo algorithm that combines both WordNet ontology and clustering techniques. Our preliminary experimental results indicated that our semantic clustering algorithm is effective, achieving an accuracy of about 90%. We also showed that this algorithm is significantly better than original lingo cluster by about 6.39%.

We plan to continue this research in the following directions. First, we will work on some criteria to avoid clusters overlapping that mean document cannot be assigned to more than one cluster. Second, we will try to remove the near duplicate cluster label.

References

[1] Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: "Learning to cluster Web search results. In: SIGIR '04". Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY,USA, ACM Press (2004) 210–217

[2] T. de Simone and D. Kazakov. "Using WordNet Similarity and Antonymy Relations to Aid Document Retrieval". Recent Advances in Natural Language Processing (RANLP), 2005.

[3] M. A. Hearst, J. O. Pedersen. "Re-examining the Cluster Hypothesis: Scatter/Gather on Retrieval Results". In Proceedings of the ACM SIGIR Conference, 1996.

[4] Y. Wang, M. Kitsuregawa. "Link-based Clustering of Web Search Results". In Proceedings of The Second International Conference on Web-Age Information Management (WAIM2001), Xi'An, P.R.China, Springer-Verlag LNCS, July, 2001.

[5] Stanisław Osiński, Dawid Weiss. 2005. "A concept-driven algorithm for clustering search results". IEEE Intell. Syst. 20, 3, 48–54.

[6] A. Hotho, S. Staab, and G. Stumme. "Wordnet improves text document clustering". In Proc. of the SIGIR 2003 Semantic Web Workshop, pages 541–544, 2003.

[7] J. Sedding and D. Kazakov. "Wordnet-based text document clustering". ROMAND, page 104, 2004.

[8] L. Jing, L. Zhou, M.K. Ng, and J.Z. Huang. "Ontology-based distance measure for text clustering". In Proc. of SIAM SDM workshop on text mining, Bethesda, Maryland, USA, 2006.

[9] Walter Brenner, Rudiger Zarnekow, Hartmut Wittig. "Intelligent Software Agents: Foundations and Applications". Springer Verlag; ISBN: 3540634118. July,1998.

[10] F. Eassa, and H. Alghamdi, "Agent Based Information Retrieval System", Proc. Of the 17th National Computer Conference at AlMadinah, KAAU , 5-8/4/2004, 265-279.

[11] Durfee, E.H., Lesser, V.R. and Corkill, D.D. "Trends in Cooperative Distributed Problem Solving". In: IEEE Transactions on Knowledge and Data Engineering, March 1989, KDE-1(1), pages 63-83

[12] Genesereth, M. "An Agent-based Framework for Interoperability". In: Software Agents, J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997, pages 317-345

[13] Amine A., Elberrichi Z., and Simonet M., "Evaluation of Text Clustering Methods Using WordNet" The International Arab Journal of Information Technology, vol 7, no. 4, pp. 349;357, 2010.

[14] Stanisław Osiński, Jerzy Stefanowski, Dawid Weiss: "Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition". Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM´04 Conference, Zakopane, Poland, 2004, pp. 359—368
[15] tf–idf. (n.d.). In Wikipedia. Retrieved December 4, 2011, from http://en.wikipedia.org/wiki/Tf%E2%80%93idf
[16] K. Baker, "Singular Value Decomposition Tutorial", 200S. Available at http://www.cs.wits.ac.za/~michael/SVDTut.pdf
[17] tanisław Osiński, Dawid Weiss. "Carrot2: Design of a Flexible and Efficient Web Information Retrieval Framework". Springer Lecture Notes in Computer Science, vol. 3528, pp. 439—444, Proceedings of the third International Atlantic Web Intelligence Conference (AWIC 2005), Łodź, Poland, 2005
[18] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, Jinwen Ma. "Learning to Cluster Web Search Results". SIGIR 2004.