Improving Web Search Results with a Natural Language Filter

Liem Luong
University of Houston - Downtown
Email: liemluong1986@yahoo.com

Professor Ali Berrached - Advisor
Computer and Mathematical Sciences Department
University of Houston - Downtown
Email: berrached@uh.edu

Abstract

Natural Language Processing (NLP) is a field of research that aims to enhance the interaction between computers and humans. Many techniques for NLP have been developed over the past years. Pertaining to web searches, NLP has the potential of improving the human-computer interaction by allowing the user to express their search query more naturally and help the search engine better focus its search on the specific information being requested, thus reducing the amount of missed hits. Therefore, applying a natural language filter will be a good technique in order to enhance the search. Our proposed model includes four phases: first, we extract the main objective of the query based on the categorized topic search. Second, we search using Microsoft Live Search API and store the returned results in an XML database. Third, we use various filtering techniques to filter out the precise results. Finally, we display the final results to the user. The number of results returned will be less than the number from a traditional search engine such as Google Live Search because we have filtered out the ones not related to the given query.

Introduction

The internet has become a powerful and widely used tool. It helps connect people from around the globe and allow them to engage in various activities that can range from reading the news to playing online games, to conducting business. Search engines are the most widely used services on the internet. Though search engines have improved significantly over the past few years, they still lack the interactivity and user friendliness that most users need. Obviously, most humans prefer to interact using their native “natural” language. NLP, in general and specifically in natural language web search, can help users who have little computer experience (not computer geeks). Our proposed model uses a natural language filter tool implemented based on a categorized topic search in order to determine the main objective of a search query [1].

Model

Our initial work submitted to the UH's Research Digest [2] [3] [4]. This is a follow-up work extending the previous work to a full paper. In this section, we give a brief overview of each phase.

3.1 Query submit phase: This phase consists of 4 steps:

1. Parse the query to determine the basic sentence structure by classifying tokens into nouns, adjectives, adverbs, and verbs.
2. Determine the type of question inferred by the given query based on the categorized topic of who, what, where, and when [1].
4. Calculate the total weight (an identified number) of the query based on each individual weight of the tokens [4]. The weight of each token is computed based on the token's distance from the original word and the structure of the word. The weight of a token is calculated based on the following rules:
   - First, the weight of synonymous word is determined on how close it is to the original word. The closest synonym of the original word will have the highest order (smallest weight is 0). Then the synonym of that closest synonym will have the next lower order (increasing weight in increments of 1) than the closest synonym. The list of these synonyms will be created to record these weights.
   - Second, the structure of the original word (the word before and the word after the original word) can have the high order (smallest weight start from 0) than the other structures of that original word. The list of these structures will be created to record these weights.

3.2 XML database phase: This phase consists of 2 steps:

1. Submit the given query to Live Search API (Microsoft search engine) in two formats:
   - The first format is the structure of the original word and the objective of the given query from the previous phase in order to find the relative results.
   - The second format will be the same from the first format but has to be inside the double quote (“...”) in order to find the exact results.
2. Retrieve the results and store in the XML database.

3.3 Filter phase: This phase consists of 3 steps:

1. Calculate the total weight of the results from the search API. This weight will be calculated based on the ratio of the exact results and the relative results. Weight = (N exact results) / (N relative results).

2. Filter by comparing the weight of the submitted query to that of the results. The list of weights (generated from the query-submit phase) and the list of weights (returned results) will be added together. The list will be created to rank these final weights in increasing order.
3. Filter by checking the order of the appearance of each token of returned results along with the submitted query. This second filter is important because most of the traditional search engines such as Google and Microsoft Live Search use keywords for searching and return results with the keywords that are far away in content. Therefore, this can improve our precision by deleting irrelevant responses.

3.4 Display results phase:

This is the final phase and consists of displaying the final results from the XML database to the user in decreasing order of accuracy.

Implementation

So far we have been able to implement the first three steps of the first phase. The raw data files we use are from the WordNet from Princeton University. (See figure 2 – figure 5). The top portion shows the raw data while the bottom portion shows the word and its synonym extracted from that raw data [5]. Note that each word is assigned a unique ID.

Conclusion

In conclusion, we have achieved some results for the four structure files and its synonyms. We are working on the objective of the input query and also on the categorized topic searches based on proper English grammar rules. The results that we achieved from this semester will play a significant role for the next phases in our proposed model of natural language web search. For future works, the next phase will be the SQL database that we will tackle on to store XML formatted results from the Microsoft Live Search API.

Acknowledgements

This work is partially supported by a NSF Grant CNS-0540592 and A U.S. Army Research Office grant W911NF-04-1-0024. The authors would also like to acknowledge the Center for Computational Science and Advanced Distributed Simulation (CCSADS) and the Scholars Academy at UHD for their continuous support.

References

4. Ojima, Rokus, Roth, Dan, and Sammons, Mark., “Token-level Disambiguation of VerbNet Classes” (Computer Science Department – University of Illinois at Urbana-Champaign).