Google Sets

Project Report

Michael Grazier
graz0018@d.umn.edu
05/07/04

Problem Description:

The heart of the Internet has always been Information Retrieval. There are many search engines that allow users to enter in keywords to search the engine's internal database to obtain the results. But what if you don't know exactly what you're searching for? For example, perhaps you are a big fan of the musicians Jimmy Page and Jimi Hendrix, and you would like to find similar musicians. Simply searching for the keywords: "Jimmy Page AND Jimi Hendrix" will most likely not give you the results you are looking for. Likewise, you could find new authors of styles similar to the authors you know.

Google Sets (*http://labs.google.com/sets*) does just this. Through its own internal algorithms, Google Sets is able to find a taxonomy (classification) that fits the user's input, and from that taxonomy, it is able to derive similar words or phrases. What I see as the real benefit of this tool is to be able to limit search queries based on the taxonomy that Google Sets finds from the user's input. For example, if a user wanted to find the top speed of the feline Jaguar, it would be useful to limit the search queries to the taxonomies that pertain to the cat-like animal instead of the taxonomies that pertain to automobiles. Using the example from the paragraph above, entering in "Jimmy Page" and "Jimmy Hendrix" into Google Sets retrieves a list of all the great guitarists, Clapton, Satriani, Vaughan, Beck, etc.

Two papers that discuss organizing data into taxonomies are: (Chakrabarti, et. al., 1998), (Hearst, 1992). The first paper describes an automatic system that organizes a large text database heirarchically by topic. They describe a method that uses statistical pattern recognition to seperate the feature of the text (the words that describe the text's domain) from the noise words (words that do not help describe the text's domain). The second paper is very similar. It describes various patterns in english text that are useful in determining hyponym relationships (A Jaguar is a hyponym of felines -- that is, a jaguar is a (kind of) feline). I intend on using English language patterns as described in (Hearst, 1992) as the basis for my program.

Overview of Solution:

Using Google's web API, I intend to search for a series of patterns commonly found in english when enumerating ideas. An illustrative example would be in the sentence, "...works by such authors as Herrick, Goldsmith, and Shakespeare." We can see that Herrick, Goldsmith, and Shakespeare can all be classified as authors. Similarily, there are other such patterns that can be found through observation of natural text. Furthermore, it is suggested that these relationships can be found automatically, via computation (Hearst, 1992).

To further illustrate, I queried Google with the search string: "such as Eric Clapton". Returned in the first few results was the string: "In the 1960s and 1970s artists such as Eric Clapton, Janis Joplin and Jimi Hendrix..." From this we can clearly see that Eric Clapton, along with Janis Joplin and Jimi Hendrix, are artists -- more specifically, 1960's and 1970's artists.

After determining hypernym-hyponym relationships, I will take the most commonly occuring hypernym (artists in the above example), and use that in a new query where I will use the language patterns previously mentioned to find Google results with terms similar (in some form) to the terms entered by the user. After parsing enough results, I should have a large enough set of hyponyms from which the final output set will be constructed from. Tentatively, I plan on taking the most frequently occuring hyponyms in order to construct the final output set. This is in order to help prevent anomalies from appearing in the final set.

To reiterate, the steps I will take to reproduce results similar to that of Google Sets, I will:

1) Use English language patterns combined with the user's search terms to query google. "*such as running*"
2) Parse the results to determine which hypernym the search terms might belong to. "*In competitive **sports**, such as running...*"
3) Use that hypernym in a new Google query that uses English language patterns as in Step 1 to help find related terms.

"*sports, such as running*"

4) Parse the results from the query in Step 3 to populate a large set of possibly related terms.

5) Take the most frequent (perhaps most probable?) terms to use as the final output set.

The English language patterns that I will be using are:

1) "Such <hypernym> as <hyponym1>, <hyponym2>.."

2) "..<hyponym1>, <hyponym2> or other <hypernym>"

3) "..<hyponym1>, <hyponym2> and other <hypernym>"

4) "..<hypernym>, including <hyponym1> and/or <hyponym2>"

Evaluation Plan:

In order to determine the validity of my results, I propose to use results taken from WordNet. WordNet "... is an attempt to organize lexical information in terms of word meanings rather than word forms." WordNet v2.0 contains nearly 115,000 unique noun strings including compounds and proper nouns. WordNet's results are impressive and can be considered as a good basis for comparison. I will access WordNet using perl's built-in system function. Using WordNet's coordinate terms option, I will produce a list of words similar to the user's term for comparison. No quantitative measure will be produced due to lack of reliability.

Experiments:

Note: WordNet results have been omitted for clarity.

The first experiment I ran used the input terms *blue* and *red*. The output of gsets.pl is:

```
csdev045% perl gsets.pl blue red 10
ESTIMATED NUMBER OF RESULTS: 88
(Possibly) Related Words:
        green
        move
        she
        can
        solid
        orange
        lasers
        state
        new
        gold
```

My program was able to identify three new terms which are results that I consider to be exceptionally good. A larger stoplist of words (a list of common words that the program will use to block from the results) would help limit false results somewhat. Perhaps a Part of Speech tagger would also help get rid of words that are clearly not nouns in this case.

The second experiment used the input terms of *magazine* and *book*.

```
csdev048% perl gsets.pl magazine book 10
ESTIMATED NUMBER OF RESULTS: 315
(Possibly) Related Words:
        magazine
        book
        who
        every
        consignment
        sale
        person
        condition
        business
        newspaper
        electronic
```

The program was able to find one new related term, *newspaper*. These results are not as good as the previous experiment's; I hypothosize the reason is due to Google producing results that are largely commercial and geared towards selling products (I would guess in this case the results came from websites selling e-books or other e-publications). Again, it would appear that a larger

stoplist and the inclusion of a Part of Speech tagger as a filter would improve results here.

My third experiment, in homage to the Friends finale, was run with the input terms *Rachel* and *Ross*.

```
csdev049% perl gsets.pl Rachel Ross 10
ESTIMATED NUMBER OF RESULTS: 0
```

This experiment shows the weakness in this approach. Looking for patterns is only useful *if the patterns exist*. Here, we can clearly see that Google finds no such English language pattern containing Rachel and Ross. This is the most severe limitation of this approach and there is no solution other than to develop another approach to finding sets of similar words.

References:

Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, Prabhakar Raghavan: *Scalable Feature Selection, Classification and Signature Generation for Organizing Large Text Databases into Hierarchical Topic Taxonomies*. VLDB J. 7(3): 163-178(1998)

Marti A. Hearst. *Automatic acquisition of hyponyms from large text corpora*. In Proceedings of the Fourteenth International Conference on Computational Linguistics, pages 539--545, Nantes, France, July 1992.