Using WordNet glosses to refine Google queries

Jan Nemrava

Department of Information and Knowledge Engineering,
University of Economics, Prague, W.Churchill Sq. 4, 130 67 Praha 3, Czech Republic
nemrava@vse.cz

Abstract. This paper describes one of the ways how to overcome one of the major limitations of current fulltext search engines. It deals with synonymy of the web search engine results by clustering them into relevant synonym category of given word. It employs WordNet lexical database and several linguistic approaches to classify results in search engine result page in appropriate synonym category according to WordNet synsets. Some bootstrapping methods to refine the classification are proposed and some initial experiments are described.

Keywords: text mining, text classification, web search engine

1 Introduction

Fulltext search engines have recently become a basic tool for acquiring arbitrary information from the World Wide Web. The amount of queries inserted into Google rises rapidly and so does the number of indexed pages. *'To Google'* became a commonly used verb describing the act of searching any information on the Internet. Nowadays, Google has an Internet domain in 135 world countries and is a world most leading search engine with its 88 language interfaces This provides excellent conditions for an easy access to any kind of information from our desktop PC and makes the proclaimed information society viable. Nevertheless, there exist some limitations that play an important role in searching information within a keyword based search interfaces. One of the web search key problems is that people tend to insert too general queries (according to Search Engine Journal [1], in 2004 more than 50% of all queries inserted were one or two words long), which leads to huge amount of returned hits to a given query. The way how to deal with a huge amount of returned web pages is to arrange the results according to their proper meaning using their synonyms or the word sense disambiguation. The purpose of this paper is to describe some techniques how to arrange returned web sites into appropriate synonym classes using large lexical database *WordNet*[1] for discovering the synonyms and *Hearst Patterns* for discovering is-a relations between the query word and its possible superclass (i.e. hypernym) concept.

[1] http://wordnet.princeton.edu/

The structure of this paper is as follows: Section 2 describes our goals and techniques used for this approach and gives examples how they work. Section 3 shows the same tests and initial experiments with classification results into proper synonymic class. Before concluding, some drawbacks and limitations and also some relevant work on this topic are discussed in section 4.

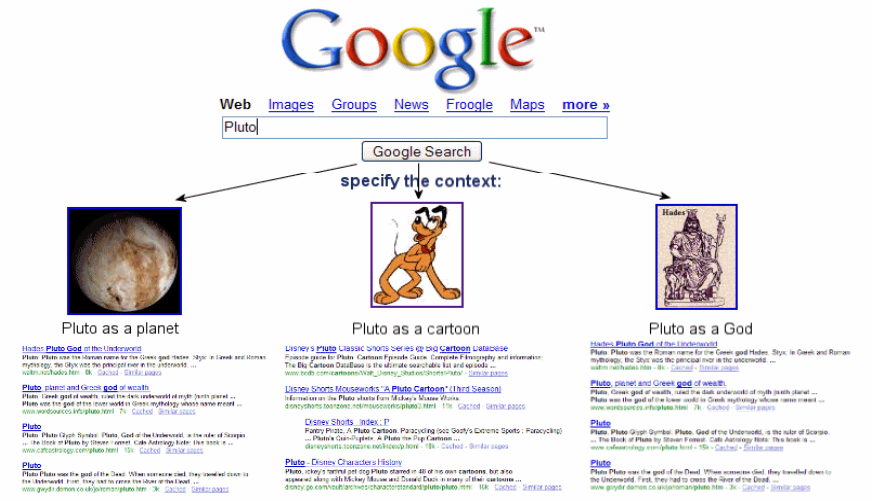

Fig. 1. A context suggestion interface

2 Motivation

As it was stated in the Introduction, the problem of ambiguous queries is a strong limitation of current web search technology. There are already some query refinement techniques emerging, which allow users to zoom into more specific query, but most of the time they only provide a "query modification" lists as a single list without distinguishing between the real meanings of given word. Another query refinement method recently introduced by leading fulltext search engine is offering real time suggestions while the user is typing in his query. One of the advantages is that the user sees the most suitable word form (though may not be the grammatically or semantically best one, but it is the one that is used by most of the users). Google Suggest[2] is good example of this method.

[2] http://www.google.com/webhp?complete=1

To our knowledge there isn't any fulltext search engine that would be able to separate returned results according to their meanings, except Vivisimo[3], which is not known in public.
In this paper we would like to present approach that use existing dictionary and glosses describing its concepts together with the largest text corpora available, the Internet, to discover meanings that the word inserted carries. This work was inspired by Philipp Cimiano's work on Pankow system and the idea of using heterogenous evidence for confirming *is-a* relation.

3 Information Sources

In this section, we will describe the above mentioned techniques in detail. All approaches used here are well known among semantic web community for a long time. They are frequently used for ontology learning and creating is-a relations and taxonomies. Namely they are:

- **WordNet** - large lexical database containing words ordered in synsets (synonym sets).
- **Hearst Patterns** - technique exploiting certain lexico-syntactic patterns to discover is-a relations between two given concepts
- **monothetic clustering** - information retrieval technique used for grouping documents according to specified features
- **fulltext search engine** - Google API interface
- **NLP** - natural language processing techniques.

3.1 WordNet

The main source of information is WordNet [6]. WordNet is a huge lexical database containing about 150,000 words organized in over 115,000 synsets for a total of 203,000 word-sense pair. Each word comes along with a short description called a *gloss*. The glosses are usually one or two sentences long. Beside the fact that all ordinary part of speech are present it contains nouns which are of major importance for us, because one of them is most likely a super concept (a hypernym) to the given word. Each gloss is preprocessed and then labeled by POS tagger first. The preprocessing contains elimination of punctuation, hyphenation and stop words. Next step is POS tagging and only nouns are kept and saved as *candidate words*.

3.2 Hearst Patterns

Hearst patterns are lexico-syntactic patterns firstly used by M.A.Hearst[7] in 1992. These patterns indicate the existence of class/subclass relation in unstructured data source, e.g. web pages. Examples of lexico-syntactic patterns that were described in [7] are following:

[3] http://www.vivisimo.com

- NP_0 *such as* NP_1, NP_2,...,NP_{n-1} *(and | or)* NP_n
- *such* NP_0 *as* NP_1, NP_2,...,NP_{n-1} *(and | or)* NP_n
- NP_1, NP_2,...,NP_{n-1} *(and | or) other* NP_0
- NP_0 *(including—especially)* NP_1, NP_2,...,NP_{n-1} *(and | or)* NP_n
- and very common "NP_i *is a* NP_0"

Hearst firstly noticed that from patterns above we can derive that for all NP_i, $1 \leq i \leq n, hyponym(NP_i, NP_0))$. Given two term t_1 and t_2 we are able to record how many times some of these patterns indicate an $is-a$-relation between given t_1 and t_2. Some normalizing techniques should be employed as some of the patterns will likely occur more frequently than the others. Although Cimiano [2] noticed that Hearst patterns occur relatively rarely in closed corpus and as described later, it is applicable also on Internet, their results provide valuable information. The main drawback is that Google search does not offer to use proximity operators and with the query requested as an exact match user must enter exact order of the whole pattern. For example searching for pattern *"planets such as Pluto, Neptune and Uranus"* will provide about 50 results, while *"planets such as Pluto, Uranus and Neptune"* won't return any. The most powerful pattern that we use for primary decisions is the "NP_i *is a* NP_0".

3.3 Clustering

Associating documents to relevant category (synonym category in our case) is a task very similar to a classic information retrieval task named by van Rijsbergen[12] *polythetic clustering*, where documents' membership to a cluster is based on sufficient fraction of the terms that define the cluster. As stated in [13] creating is-a relations is a special case of polythetic clustering where subclass belongs only to one superclass and this means that the membership is based only on one feature, called *monothetic clusters*.
This alternative form of clustering has two advantages over the polythetic variety. The first is the relative ease with which one can understand the topic covered by each cluster. The second advantage of monothetic clusters is that one can guarantee that a document within a cluster will be about that clusters topic. None of this would be possible with polythetic clusters.

3.4 Google API

The world leading fulltext search engine provides direct access to its huge databases through Google API[4]. It has limited daily number of queries and compared to HTML based interface is relatively slow, but it provides easy access from any programming language. Each query is responded in the same way as is the HTML interface. User can get number of results, web page titles, links and snippets(short description of web page based either on META tag description or part of text with emphasized keywords). Our algorithm search for very specific

[4] http://www.google.com/apis

text patterns and we are interested only in aggregate number of results.

Next session describes application of above described information sources and some initial results.

4 Discovering the synonym classes

It was already described in a section about WordNet, that certain nouns from so called glosses are of our main interest. According to our observation glosses mostly contain one noun that is a hypernym to the given concept. This is a core prerequisite for our method as our aim is to find that hypernym noun among the words in gloss. After application of simple NLP methods we discover *candidate nouns* for each gloss. What follows is a description of concrete situation that our script deal with. The example is a word *Pluto* which can be found in three different contexts according to WordNet. Pluto can be either a planet, a god or a cartoon.

- Candidate nouns for concept Pluto.
  - SYN 1 *planet;sun;orbit;planets;*
  - SYN 2 *Greek;god;underworld;mythology;brother;Zeus;husband;Persephone;*
  - SYN 3 *cartoon;character;Walt;Disney;*
- Patterns applied on SYN 1 - number of returned results is in brackets
  - "Pluto is a planet" (1550), "Pluto is planet" (145)
  - "Pluto is a sun" (2), "Pluto is sun" (0)
  - "Pluto is a orbit" (0), "Pluto is orbit" (1)
  - "Pluto is a planets" (0), "Pluto is planets" (0)

It is necessary to take into a consideration the total amount of web pages where the words are mentioned and use this value to normalize the values.

$$w(i) = tf(i)/TC(i) \tag{1}$$

where i represents the $i-$th synonym class, tf is number of results for given pattern and TC is number of web pages returned when querying two terms without any constraints, it represents the popularity of the given pair of terms. Candidate for the hypernym noun is then simply the highest value from all synonymic class array.

$$W = max(w(i)) \tag{2}$$

This candidate noun needs to be validated and confirmed by another Hearst patterns. The problem with a necessity of strict word order was discussed in previous session. We must cope with this problem in order to find another pattern to validate the results from "is a" step. Pattern NP_{n-1} *and other* NP_0 was chosen, because we predicts its bias to be the lowest among all remaining patterns. In this pattern we had to cope with creating a plural form of each word. Some simple rules were adopted, such as adding "ies" suffix at the end of the word when the last character is "y" etc.. No language exceptions were taken in consideration.

– Patterns tested in a validation step (returned hits are in brackets)
  - "Pluto and other planets" (57)
  - "Pluto and other planet" (0)
  - "Pluto and other suns" (0)
  - "Pluto and other sun" (0)
  - "Pluto and other orbits" (0)
  - "Pluto and other orbit" (0)
  - "Pluto and other planetss" (0)
  - "Pluto and other planets" (57)

Maximum value from the array is considered as *hypernym noun* If both patterns determine the same noun, it is taken as a hypernym noun. In the opposite case some other techniques to confirm or reject this hypothesis should be applied. The possibilities are discussed in last section. The process of assigning right hypernym noun is repeated for all synonym classes that were given by WordNet.
The test set consists of about 50 of proper nouns from space, travel and zodiac area. At the beginning it was necessary to manually check whether all the words from the test are listed in WordNet. The result was that 96% (i.e. 48 from 50) proper nouns have their gloss in WordNet. After all the tests has been carried out, it was necessary to check the correspondence of the discovered hypernym with the real world concepts.
From tested set, 62% (31 words which contained 61 synonymic classes in total) were assigned by a hypernym that corresponded to real life objects. 9 words and all their meanings were assigned wrongly. More detailed analysis of words that were incorrectly labeled can be found in Table 2.

Mining for other synonyms than those explicitly stated in WordNet would definitely provide better results in some cases, on the other hand the certainty of wrongly assigned hypernym noun would undoubtly rise.

Table 1. Overall precision

Total number of words in list	50	(100%)
Words listed in WordNet	48	(96%)
Correct	39	(78%)
- completely correct	31	(62%)
- partially correct	8	(16%)
Wrong	9	(18%)

5 Related work

This section discusses work related to exploitation of WordNet glosses and some query refinement system. Since word ambiguity present an important issue in

Table 2. Statistics of wrongly discovered terms

Number of wrong instances	17 (100%)
Both patterns wrong	7 (41%)
"is a" correct, "and other" wrong	4 (23%)
"is a" wrong, "and other" correct	6 (35%)

Table 3. Examples of negatively labeled synonyms.

Proper Noun	"Is a" pattern	"and other" pattern
Greenland	island	Arctic
Reykjavik	Iceland	Iceland
Kenya	Great	Great
Luxembourg	-	-
Luxembourg	city	city

Information Retrieval community, there has been a lot of efforts invested to discover how to deal with the problem. The importance of disambiguated words and concept further increased with introduction of ontologies as a core of the so called Semantic Web. Nowadays, there is an enormous effort on this research field. The most successful approaches so far, either reuse some knowledge stored existing sources (exploiting Web directories structure, dictionaries or tagged corpuses) or make use of the inherited redundancy of information that are present on Internet (e.g. Armadillo [4] or KnowItAll [5]). Both of these systems continually and automatically expands the initial given lexicon by learning to recognize regularities in the large repositories by discovering some regularities, either internal to a single document or external across set of documents.

Project that use similar ideas to ours is one called WordNet::Similarity [9]. It is a tool kit written in Perl implementing several algorithms for measuring semantic similarity and relatedness between WordNet concepts. Two of algorithms (lesk and vector measures in concrete) uses WordNet glosses. Lesk finds overlaps between two given glosses to count the relatedness of them. The vector measure creates a cooccurrence matrix for each word used in the WordNet glosses from a given corpus, and then represents each gloss/concept with a vector that is the average of these cooccurrence vectors.

Project that inspired this work is called PANKOW (Pattern-based Annotation through Knowledge on the Web) and was created by Cimiano et al. [3]. This work focuses on application of Hearst patterns over a given ontology to discover is-a relations solely from Internet. Some of the data tested in our paper were actually taken from their work.

Query refinement and semantic structural interconnections

6 Conclusions

Approach for discovering synonym classes of given proper nouns was presented in this paper. In particular we discussed usage of several freely accessible information sources, WordNet lexical database and lexico-syntactic patterns with fulltext search engine in particular. List of some commonly used proper nouns was created and the proposed method was tested with this list. From 50 test concepts with 92 synonyms in total we get precision 62 percent The results were appropriate to estimations and with regard to the fact that this technique has been recently implemented and is far from mature, they found them satisfying. There are several drawbacks and suggestion for future work that will be discussed in this section.

One of the drawbacks is the system speed which depends on Google API responses which are quite slow recently. The average time to resolve one synonymic class is about 50 seconds with average 20 Google queries per one synonym class. Another objective drawback is the limitation of current Google web search interface. It has no proximity operators and the query must be either inserted as an exact match or connected with AND boolean operator. Besides these technological problems there is also a limited amount of daily queries to one thousand which is sufficient only to process about two tens of concepts.

It remains for further work to find out how to exploit the WordNet hierarchy and involve glosses from class instances and subconcepts. Introducing another validation pattern would definitely increase the precision of the system. So far, the system can handle only single word queries. Handling more words queries and deriving proper synonyms categories could be an interesting challenge. Another task would be to implement a way how to deal with words and concepts not included in WordNet. Cimiano's PANKOW similar system might be beneficial for this task.

Although this application has certain drawbacks, we showed that the idea of exploiting WordNet glosses for discovering certain facts about given concepts is viable and with some improvements in speed and validation it could serve as a helpful tool for unexperienced Internet users.

ACKNOWLEDGEMENTS

The author would like to thank to Vojtech Svatek for his comments and help. The research has been partially supported by the FRVS grant no. 501/G1.

6.1 Citations and references

References

1. Baker L.:*Search Engine Users Prefer Two Word Phrases*, Search Engine Journal `http://www.searchenginejournal.com/index.php?p=238`
2. Cimiano P. et al.: *Learning Taxonomic Relations from Heterogeneous Evidence*
3. Cimiano, P. and Staab S.:*Learning by googling.* SIGKDD Explor. Newsl. 6, 2 (Dec. 2004), 24-33.

4. Ciravegna F. et al.:*Learning to Harvest Information for the Semantic Web*, Proceedings of the 1st European Semantic Web Symposium, Heraklion, Greece, May 10-12, 2004
5. Etzioni O. et al.:*KnowItNow: Fast, Scalable Information Extraction from the Web*, Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, p.563–570, October 2005
6. Fellbaum C.: *WordNet, an electronic lexical database*, MIT Press, 1998.
7. Hearst M. A.: *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
8. Navigli R., Velardi P.:*Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 7, pp. 1075-1086, July 2005.
9. Pedersen S., et al.: *Wordnet::similarity - measuring the relatedness of concepts*. In Appears in the Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), 2004.
`http://citeseer.ist.psu.edu/644388.html`
10. Porter M.: *Porter Stemmer Algorithm*, [online],
`http://tartarus.org/~martin/PorterStemmer/`
11. Ratnaparkhi A.: Adwait Ratnaparkhi's Research Interests, [online],
`http://www.cis.upenn.edu/~adwait/statnlp.html`.
12. Van Rijsbergen C.J.: Information retrieval (second edition), Chapter 3, Butterworths, London, 1979.
13. Sanderson M., Croft B.:*Deriving concept hierarchies from text*,[online]
`citeseer.ist.psu.edu/cimiano03deriving.html`
14.
15. Weiss S.M. et al: *Text Mining - Predictive Methods for Analyzing Unstructured Information*. Springer, 2005, ISBN 0-387-95433-3.
16.