Can Ontologies Improve Web Search Engine Effectiveness Before the Advent of the Semantic Web?

Jacques Robin¹ and Franklin Ramalho²,¹
¹ Centro de Informática, Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – Recife – PE - Brasil
² Departamento de Sistemas e Computação, Universidade Federal de Campina Grande (UFCG)
Caixa Postal 10106 - Campina Grande – PB - Brasil
e-mail: jr@cin.ufpe.br, franklin@dsc.ufcg.edu.br

Abstract

The advent of the semantic web will not be immediate, due to a shortage of knowledge engineering manpower to annotate a critical mass of web documents with concepts from ontologies. In this paper, we argue that wide-coverage linguistic ontologies have an alternative usage to improve web search effectiveness while the web remains predominantly textual. They can be used by search engines to perform simple semantic processing, such as automatically expanding the query keywords with alternate wordings of the same or immediately neighboring concepts. We present a conclusive empirical experiment that shows that sizable relative effectiveness gains can be achieved by various expansion strategies for both unbounded and bounded retrieval result sets. This experiment also shows that the effectiveness of such ontology-boosted search on a textual web with no semantic annotations remains far away from the very high precision and recall promise of the semantic web.

1. Introduction

Web search engines have become a pervasive part of our daily life. We use them at home and at the office, for accessing the global Internet as well as organizational intranets and for a wide range of purposes such as e-business, e-shopping, e-entertainment and e-education. They are a billion-dollar business with one of the most massive user base in the software industry. Yet, their effectiveness as Information Retrieval (IR) systems remains frustratingly low. They still typically return way too many irrelevant documents (low precision) and miss way too many relevant ones (low recall). At the core, this is due to the fact that the current paradigm for web search still consists of matching queries and documents at the superficial level of keyword linguistic forms, instead of the deep level contextual semantics. The pervasiveness of lexical polysemy and synonymy in natural languages make a query expressed in terms of keyword surface linguistic forms a both ambiguous and incomplete specification of the user’s underlying semantic information need.

One proposed middle-to-long-term solution to this problem is the vision of the Semantic Web [3]. In this vision, content published on the web will no longer be encoded in a purely textual, natural language form. Instead, it will be encoded in a hybrid form, with textual fragments annotated by disambiguating semantic tags referring to instances of concepts formally defined in web accessible reference ontologies. Such ontologies come in several flavors: common-sense ontologies (or alternatively top ontologies or upper ontologies) define
general, high-level concepts common to most semantic domains, while *domain ontologies* define specialized concepts specific to restricted domains; and *linguistic ontologies*\(^1\) define concepts based on the existence of words to refer to them in one or several natural languages, while *purely conceptual ontologies* define concepts solely based on their usefulness to assist intelligent agents in various automated reasoning tasks. In the semantic web vision, content seekers will formulate their information needs in terms concepts from one or several such ontologies, instead of using mere keywords. This will empower search engines with the ability to match document content with user information need at the deep semantic level, making them extremely accurate.

Significant progresses have already been made towards this vision of turning the web into a world wide *knowledge base* for both human users and software agents. A layered set of increasingly semantic representation languages have been defined as standards for encoding ontologies, distributing them on the web, and annotating web document textual fields with concepts defined in them: XML [7], XMLS [8], RDF [15], RDFS [16] and DAML-OIL [5]. Numerous general and specialized ontologies have been developed and made available on the web [4, 9]. Practical tools are beginning to appear for distributed authoring and maintenance of ontologies, for composing web documents linked to them, and for implementing web agents that reason automatically with the knowledge they contain, to provide a variety of intelligent information services [6].

By cleverly leveraging and integrating results from decades of Artificial Intelligence\(^2\) (AI), distributed computing and software engineering research, these progresses have been impressively fast paced. Only a few pieces of the puzzle are now missing before of a full, interoperating set of semantic web enabling technologies becomes available. However, such availability will prove insufficient to turn the web semantic overnight. Much to the contrary, the web is likely to turn semantic only very slowly as well as only partially. This is because a fully semantic web would in effect constitute a pendulum comeback to the database, formal language, high-skill user information system paradigm. But it is precisely its original shift away from this paradigm and towards the textual, natural language, low-skill user information system paradigm that was the key to the web’s exponential growth and success. While the textual paradigm prevents very high precision and recall for automated search, it also allows content production and publishing to be spontaneously and massively distributed among authors with minimal technical knowledge. Composing a simple HTML document requires little skills beyond writing, formatting and understanding navigational linking. In contrast, semantically annotating documents with instances of concepts from ontologies, even with user-friendly tools, requires some basic notions of knowledge engineering that the average web page designer does not possess and will not easily acquire. This probably explains a current paradox: while XML is already broadly known in both the industrial and academic communities, XML pages still represent an infinitesimal portion of the web. It will thus take some time before the proportion of web pages published with ontological semantic annotations is large enough to impact on web search engine effectiveness.

In the meantime, how can we improve this effectiveness? Could some of the very resources and technologies being developed to turn the web semantic, could also be used to improve search engine effectiveness before it does turn semantic? In particular, could

\(^1\) Common-sense linguistic ontologies are sometimes called lexical databases and domain linguistic ontologies sometimes called terminologies.

\(^2\) In this paper, we use AI in its most inclusive sense, including all paradigms (symbolic, probabilistic, fuzzy, connectionist, evolutionist, hybrid) put forward to model intelligent behavior and build intelligent systems.
ontologies be used to improve this effectiveness in a different way than the one put forward by the semantic web vision?

In this paper, we shed some preliminary light on this vast question. In particular, we investigate whether providing a regular search engine access to a linguistic, common-sense ontology such as WordNet, can significantly improve search effectiveness in retrieving a set of purely textual documents with no semantic annotation. In such ontology, each concept is defined in terms of a set of synonym words called synsets. Synsets are organized in a class hierarchy through the hypernymy/hyonymy relation (i.e., the superclass/subclass or is-a relation). In addition, individual words are also linked, across synsets, via other semantic relations such as meronymy/holonymy (i.e., the part-of/has-part relation), and antonymy (an opposition relation along some semantic dimension). Because of its dual conceptual and lexical nature, an ontology of this type seem intuitively very promising to support intermediate search approaches that bridge the gap between the purely lexical matching approach of current web search engines and the purely conceptual matching approach of the semantic web. It opens the possibility to embed some simple semantic reasoning inside the search engine, while processing a document base that is still purely textual.

One such simple semantic reasoning is the automatic expansion of the user query keywords by the other members of its synset, e.g., expanding a query with the word car with the word automobiles. Intuitively, such expansion should considerably improve search recall. However, lexical ambiguity could lead to expansion for other meanings than the one intended by the user and negatively affects precision, e.g., in the context of a cable car query, expanding car with automobile would lead to return more irrelevant documents. Can overall effectiveness, balancing precision and recall, be significantly improved by such simple semantic processing? Previous IR experiments, in settings that were not representative of searching textual documents on the web, have brought contradictory hints about this questions, with some yielding positive results [1, 18], and others negative ones [22].

In this paper, we revisit this question in an IR setting that specifically emulates the task of current web search engines, bringing new positive results. We show that query expansion using the WordNet ontology improves overall effectiveness for a variety of expansion strategies, and both for metrics that consider the entire set of retrieved documents as for those that consider only the top-ranked ones. The paper is organized as follows. In the next Section, we describe the ExpanSyns workbench that we developed to carry out empirical experiments that measure the impact on retrieval effectiveness of coupling web search engines with linguistic knowledge resources. In Section 3, we present a set of such experiments that measure the impact of coupling a commercial search engine with WordNet to perform automatic query expansion. In Section 4, we highlight the original contribution of these experiments by comparing them with related work. In Section 5, we conclude by identifying the questions left unanswered by our research and indicating directions for future work.

2. ExpanSyns: a Workbench to Measure Effectiveness Gains from Providing Web Search Engines with Linguistic Knowledge

We developed ExpanSyns a workbench for carrying out empirical experiments that measure effectiveness gains from various intelligent search strategies relying on linguistic knowledge resources. As shown in Figure 1, its architecture consists of three layers: the data management layer, the application logic layer and the user interface layer.

The data management layer accesses three types of data sources. The first is the knowledge base used to support the simple semantic reasoning being evaluated in the experiment. The second is the test collection used that is used in the case of an automated,
batch experiment. Such collection generally contains a set of documents, a set of queries and a set of relevance judgments specifying which documents are relevant for which queries. The last one is a simple relational database in which all the data of the experiment is stored for future examinations.

The application logic layer consists of six components, one for each processing stage in a typical experiment. The first stage involves accessing the queries, from the user in an interactive experiment or from the test collection in a batch experiment, and preprocessing them using linguistic or ontological knowledge, e.g., expanding them with WordNet synonyms. The second stage involves translating the preprocessed query into the query language of the web search engine used for the experiment, and submitting the translated query. The third stage involves extracting from web page set returned by the engine, the links to the retrieved documents, filtering out noisy links such as banners, pictures, navigating aids and so forth. The fourth stage involves checking the relevance of the returned documents. In the case of interactive experiment this is done by sending the pages to the user interface for marking. In the case of a batch experiment it is simply a matter of accessing the relevance judgment base. The fifth stage involves calculating the effectiveness metrics from the relevance judgments. The sixth and last stage involves recording the metric values in the experiment database.

Currently, ExpanSyns supports the information retrieval metrics shown in Figure 2. In addition to standard IR precision and recall measure over unbounded result sets, ExpanSyns also calculates the F-measure [21] over such sets, as well as precision over bounded sets. Since recall can be improved at the expense of precision and vice-versa, F-measure is needed to assess overall effectiveness with a single parameter. As the harmonic mean of precision and
recall, F-measure has the desirable property to increase not only when precision increase for a fixed recall and vice-versa, but also when both get close to one another, thus considering as most effective search strategies that strike a balance between the two. Precision over results bounded to the N top ranked is a key metric for web search engines, since (1) it takes into account the ranking scheme, and (2) so few users are willing to inspect more than 10 or 20 results. Note that this bounded results setting differs from the unbounded results setting in one important way: the result bound turns the precision ratio between two strategies equal to the recall ratio between them (see [17] for a formal proof). The two metrics thus become redundant instead of complementary.

\[
\begin{align*}
\text{Precision } P_u &= \frac{RR}{RR + IR} \\
\text{Bounded Precision } P_b &= \frac{RR}{RR + IR} \\
\text{Recall } R_u &= \frac{RR}{RR + RN} \\
\text{F-measure } F_u &= \frac{2 * P_u * R_u}{P_u + R_u}
\end{align*}
\]

Where:

- \(RR\) = number of relevant retrieved documents;
- \(IR\) = number of irrelevant retrieved documents;
- \(RN\) = number of relevant not-retrieved documents;
- \(B\) = bound on the number of retrieved documents considered.
- \(RR_k^B\) = number of \(k\) top ranked relevant retrieved documents, with \(1 \leq k \leq B\);

**Figure 2: Effectiveness metrics formulas calculated by ExpanSyns.**

### 3. Case Study: Improving a Commercial Web Search Engine Effectiveness with the WordNet Ontology

We used ExpanSyns to evaluate the impact on retrieval effectiveness of expanding query keyword with some of their synonyms and hypernyms encountered in the WordNet ontology for some of their word senses. In this experiment, we used the four effectiveness metrics supported by ExpanSyns, using bounds of 10, 20, 30, 40 and 50 top-ranked results for the bounded precision metric. In what follows, we describe in detail, the web search engine, test collection, semantic relations and expansion strategies that we used in this experiment. We then turn to the results we obtained.

**Search Engine**

The clone of the commercial search engine that we use for our experiment indexes, retrieves and ranks documents solely based on their content, without any form of collaborative filtering. In addition, it approximates document and query content by the word *forms* they
contain, without performing any semantic processing. It indexes documents using an inverted file that contains for each word form \( W \): (1) its frequency in the inverted file, (2) the URL list of all documents \( D_i \) in which \( W \) occurs, (3) the list of frequencies of \( W \) in each \( D_i \), (4) the list of positions of \( W \) in each \( D_i \) and (5) the list of HTML markups of \( W \) in each \( D_i \). It retrieves documents using boolean search in this inverted file. Its ranking strategy is based on the vector model. It ranks documents heuristically taking into account query term frequency & inverted document frequency (TF-IDF), query term position and HTML tagging in the document, and query terms proximity within the document. Since all these techniques are fairly standard, this system constitutes a convenient base line for evaluating the effects of linguistic knowledge resource on web search engines.

For our experiments, the Radix Company gave us access for research purposes, to an exact copy of their operational commercial search engine (http://www.radix.com). We had access to the source code and therefore started from a precisely known base point, on top of which to test various extensions.

**Test Collection**

For the experiment we used disk 2 of the TIPSTER test collection that was produced as a result of the Text Retrieval Conference (TREC) and TIPSTER workshops [11], this disk contains: (1) 231,000 documents totaling 2Gb of text and covering a wide variety of topics and styles, (2) 50 benchmark queries, and (3) relevance judgments for about 78,000 of the 11,55 million possible query-document pairs. The pair subset for which relevance judgment is available, called the pool, consists of the union of the 1000 top ranked documents retrieved, for each query, by each of the 20 IR systems that participated to TREC-1\(^3\). Recall of each system is then approximated by counting as irrelevant any document out of this pool, and for which relevance judgment is in fact simply unavailable. This technique, known as pooling, is an attempt to estimate recall for document collections that are so large as to make manual inspection of each possible query-document pair impractical. In our experiment, the four metrics are measured using extraneous pooling since the pool considered here is the one built-in in TIPSTER-2 using the TREC-1 participants and not one built specially for our experiment. The advantage of this extraneous pooling technique is to permit fully automatic evaluation. Its drawback is a risk of underestimating effectiveness of the evaluated strategies in case they manage to retrieve numerous relevant documents that no TREC-1 participant could retrieve. For our purpose of comparing various query expansion strategies this leads to conservative estimates, possibly underrated the effectiveness gains of query expansion. The TIPSTER collection was ideal for our purpose for the following reasons\(^4\): (1) its inclusion of queries and relevance judgments allowed us to automate the evaluation process, (2) it is one of the largest and most heterogeneous collection available and size and diversity are two crucial characteristics for web search evaluation, (3) it has become a benchmark used by many researchers allowing cross-experiment result comparison and (4) its queries are structured in multiple fields describing an information need at various granularity. An example query is given in Figure 3. Among the query fields we chose the title field for its shortness makes it closest to typical web search queries.

---

\(^3\) i.e., a document is part of the pool for a given query if it appeared among the 1000 first ranked results of at least one system.

\(^4\) It was only after we carried out the experiment that the TREC web document collection was made available.
Tipster Topic Description
Number: 086  
Domain: Finance

Topic: Bank Failures
Description:
Document will identify recent bank failures within the United States, specifying name, location, assets, and federal or state authorities taking action.

Summary:
Document will identify recent closings by the Federal Deposit Insurance Corporation of a failed commercial bank chartered within the U.S.

Narrative:
A relevant document, in addition to giving specific identifying information on a failed bank (or banks), must be clear that the bank was chartered within the United State and that it is a bank, as opposed to another financial institution such as a savings and loan or credit union, and must specify the actions taken or being taken by the public authorities.

Concept(s):
1. Bank failure  
2. Federal Deposit Insurance Corporation, FDIC, Comptroller of the Currency, state banking authorities, state bank regulators

Factor(s):
Nationality: U.S.  
Time: current

Linguistic Resource and Expansion Strategies Evaluated

The linguistic resource we used for query expansion was the freely available Princeton Wordnet 1.6. This version contains 95,600 word forms hierarchically indexed by 70,100 word senses, themselves cross-linked by six semantic relations: antonymy, hypernymy/hyponymy (is-a relation), three different meronym/holonym (part-of relations) and synonym. As the basis for expansion, we limited ourselves to two relations: synonymy and first level hypernymy, both between nouns. This focus on nouns is motivated by the fact that all of the 100 most frequent of the queries submitted to the commercial search engine that we used for our experiment contain only nouns. Wordnet contains approximately 57000 noun forms organized into approximately 48800 noun meanings (synsets).

More precisely we evaluated the following expansion strategies:
- no expansion (NE),
- expansion with synonyms for all senses of each noun in the query (AS)
- expansion with synonyms for the 3 most common word sense of each noun in the query (3S)
- expansion with synonyms for the 2 most common word sense of each noun in the query (2S)
- expansion with synonyms for the most common word sense of each noun in the query (1S)
- expansion with synonyms and 1st level hypernyms for the 2 most common word sense of each noun in the query (2SH)
- expansion with synonyms and 1st level hypernyms for the most common word sense of each noun in the query (1SH)
In all cases, we call expansion the substitution of each keyword noun in the query by the Boolean disjunction of itself and its Wordnet listed synonyms and/or hypernyms.

Results

The experiment results are summarized in Table 1 and 2. Both have one cell for each metric-strategy pair. In each cell, the first line contains the metric mean value over the 27 queries of the experiment, while the second and third lines contain its absolute and relative variations from the no expansion base case. One key result shown in Table 1 is that all considered expansion strategies improve overall effectiveness (F-measure) by improving recall more than they worsen precision (in relative terms). Thus, synonym and/or hypernym based query expansion works. Another key result in Table 1 is that the best strategy to improve overall effectiveness is to limit expansion to synonyms of the most common word sense of each query noun, boosting F-measure +37.28% relative to the no expansion case. A third key result is that recall can be boosted up as much as +72.4% relative to the no expansion case, when expanding with both synonyms and first-level hypernyms for the first two word senses of each query noun. The last key result in Table 1 is that in absolute terms, even the best query expansion strategy yields only 11% of all relevant documents together with 77.5% of irrelevant ones. Thus, fully automatic synonym based query expansion alone, is far from sufficient to get web search close to truly effective retrieval.

<table>
<thead>
<tr>
<th>Table 1. Unbounded Metrics</th>
<th>Precisão</th>
<th>Cobertura</th>
<th>Medida-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE</td>
<td>0.254</td>
<td>0.075</td>
<td>0.067</td>
</tr>
<tr>
<td>TS</td>
<td>0.200</td>
<td>0.114</td>
<td>0.083</td>
</tr>
<tr>
<td>-26.8%</td>
<td>+51.0%</td>
<td>+23.5%</td>
<td></td>
</tr>
<tr>
<td>-5.36%</td>
<td>+3.84%</td>
<td>+1.58%</td>
<td></td>
</tr>
<tr>
<td>3S</td>
<td>0.203</td>
<td>0.113</td>
<td>0.0875</td>
</tr>
<tr>
<td>-24.5%</td>
<td>+50.2%</td>
<td>+24.2%</td>
<td></td>
</tr>
<tr>
<td>-5.08%</td>
<td>+3.78%</td>
<td>+1.63%</td>
<td></td>
</tr>
<tr>
<td>2S</td>
<td>0.208</td>
<td>0.112</td>
<td>0.086</td>
</tr>
<tr>
<td>-21.87%</td>
<td>+49.2%</td>
<td>+27.5%</td>
<td></td>
</tr>
<tr>
<td>-4.56%</td>
<td>+3.70%</td>
<td>+1.85%</td>
<td></td>
</tr>
<tr>
<td>1S</td>
<td>0.225</td>
<td>0.110</td>
<td>0.093</td>
</tr>
<tr>
<td>-12.81%</td>
<td>+46.4%</td>
<td>+37.28%</td>
<td></td>
</tr>
<tr>
<td>-2.88%</td>
<td>+3.49%</td>
<td>+2.51%</td>
<td></td>
</tr>
<tr>
<td>2SH</td>
<td>0.184</td>
<td>0.123</td>
<td>0.075</td>
</tr>
<tr>
<td>-38.0%</td>
<td>+72.4%</td>
<td>+11.9%</td>
<td></td>
</tr>
<tr>
<td>-6.99%</td>
<td>+5.44%</td>
<td>+0.80%</td>
<td></td>
</tr>
<tr>
<td>1SH</td>
<td>0.197</td>
<td>0.124</td>
<td>0.080</td>
</tr>
<tr>
<td>-29.1%</td>
<td>+64.5%</td>
<td>+19.2%</td>
<td></td>
</tr>
<tr>
<td>-5.72%</td>
<td>+4.85%</td>
<td>+1.29%</td>
<td></td>
</tr>
</tbody>
</table>

Perhaps the most encouraging result of our experiment is given in Table 2: namely the somewhat unexpected fact that synonym and/or hypernym based query expansion improves precision for the top ranked retrieved documents. This precision gain probably results from the synergy between an improved recall and an effective ranking. As recall improves, more relevant documents get retrieved. If ranking is effective, some of these documents will overcome irrelevant documents in the top-ranked list, thus improving precision bounded to.
this list. Which strategy yields such a precision gain for which ranking bound is highlighted in gray in Table 2. Interestingly, for this bounded case, just as for the unbounded case, the best strategy is expansion limited to synonyms of the most common word sense. This strategy improves bounded precision for the top 20, 30, 40 and 50 results, yielding a 37% precision gain high for the top 50 results. These results empirically confirm the quality of the word sense ranking in WordNet.

<table>
<thead>
<tr>
<th>Table 2. Bounded Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 10</td>
</tr>
<tr>
<td>NE</td>
</tr>
<tr>
<td>0.300</td>
</tr>
<tr>
<td>TS</td>
</tr>
<tr>
<td>-24.5%</td>
</tr>
<tr>
<td>-5.9%</td>
</tr>
<tr>
<td>3S</td>
</tr>
<tr>
<td>0.237</td>
</tr>
<tr>
<td>-26.6%</td>
</tr>
<tr>
<td>-6.3%</td>
</tr>
<tr>
<td>2S</td>
</tr>
<tr>
<td>0.263</td>
</tr>
<tr>
<td>-12.3%</td>
</tr>
<tr>
<td>-3.7%</td>
</tr>
<tr>
<td>1S</td>
</tr>
<tr>
<td>0.296</td>
</tr>
<tr>
<td>-1.35%</td>
</tr>
<tr>
<td>-0.4%</td>
</tr>
<tr>
<td>2SH</td>
</tr>
<tr>
<td>0.196</td>
</tr>
<tr>
<td>-53.1%</td>
</tr>
<tr>
<td>-10.4%</td>
</tr>
<tr>
<td>1SH</td>
</tr>
<tr>
<td>0.222</td>
</tr>
<tr>
<td>-35.1%</td>
</tr>
<tr>
<td>-7.8%</td>
</tr>
</tbody>
</table>

4. Related Work

While there is a large body of previous work in evaluating intelligent IR techniques that rely on linguistic and/or ontological knowledge, very few publications relate experiments that aim at evaluating techniques individually and in IR settings similar to web search. Most previous experiments measure the global impact of technique combinations (e.g., [12]), leaving the reader wondering what is the individual contribution of each technique to the reported results. The few experiments that measure such individual contributions tend to do so for queries that are more sophisticated and far longer than those submitted to web search engine by typical users. For example, [14] measures the impact of morphological processing for queries with an average length of 9.65 words. In contrast, none of the 1000 most frequent queries submitted to the site of the commercial search engine that we used for the experiment were over five word long: 60.7% consisted of a single word, 33.1% contained two words, 5% three words, 1% four words and 0.2% five words.

Various uses of Wordnet as a linguistic knowledge resource to improve IR effectiveness have been empirically evaluated in previous work [1, 10, 12, 18, 19, 20, 22, 23]. Yet other works evaluated the use of other thesauri for the same purpose [13].
Here we only have space to compare our work with the three most directly related to ours [1, 19, 23] in that they too:

- evaluate the use of Wordnet synonym sets (synsets for short) and other Wordnet relations as the sole knowledge source for retrieval improvement;
- evaluate the use of synsets only to expand term vectors during query processing;
- perform document-query matching directly on expanded, yet straightforward term-based representations.

In contrast, the other works:

- evaluate the combined use of Wordnet together with other knowledge sources, generally corpus-based term co-occurrence statistics;
- and/or use synsets for a variety of purposes, such as document indexing and contextual term disambiguation, during both document and query processing;
- and/or perform document-query matching indirectly on complex graph and cluster structures constructed through elaborated, sometimes multi-layer, statistical processing of document terms, query terms and related Wordnet terms.

Just as we do, [195] and [23] both evaluate query noun expansion with synsets. Yet, their experiment differs from ours in several key aspects. First, they measure the combined impact of using query noun synonym term expansion together with query noun disambiguation, whereas our measures the individual impact of query noun synonym expansion alone. In addition, while appropriate for their own, different purpose, these experiments do not correspond to practical web search settings for two reasons. First disambiguation is manually performed. Second, either the documents, in the case of [19], or the queries, in the case of [23], used for the experiment are of inappropriate size to be representative of typical web documents and queries. Working on caption-based image retrieval, [19] developed a special purpose collection of image captions and textual queries for their experiment. The average document (i.e., the image caption) size was 5.89 words, well below a typical hypermedia web page. While [23] used as we did the TIPSTER collection (including the TREC disk 1 plus the TREC disk 2 that we used), she used different fields of the TIPSTER queries (an example of such query and fields was given in Figure 1), in three distinct runs, all three resulting in queries far too long to be representative of those typically submitted to web search engines. In the first run, she uses as queries the entire text of the TIPSTER queries, with all its sub-fields, resulting in a query length average of 52.5 words. In the second run, she selects only the summary and concept fields resulting in an average query length of 29.2 words. Even the third run restricted to the sole summary field nevertheless yields an average query length of 11 words. In contrast, we picked the shorter, title field and further filtered out queries with a title field longer than four words. The previous experiment most similar to ours is that of [1]. They measure the individual contribution on retrieval effectiveness of both noun and verb query expansion using various combinations of Wordnet relations, without sense disambiguation. Just as [19] they are concerned with image retrieval, but instead of matching textual queries with captions, they match them with content-based retrieval dedicated meta-data associated with each image. Due to this emphasis, [1] also run their evaluation on a special purpose document collections whose elements, the image meta-data, differ from typical web pages not only by being far shorter, but also by being structured

---

5 Note that here we are not referring to the entire experiment reported in [11], but only to Run 2. The other seven runs reported in the paper fall in the not directly related category mentionned above together with the earlier paper [10].

6 Although one could consider that limiting expansion for synsets and hyponyms to only the one, two our three most common word senses listed in Wordnet, as in some of our expansion strategies, constitutes a coarse disambiguation heuristic.
data instead of free text. For this reason, the setting of this experiment is also completely different than that of general web search.

To sum up, although the idea of WordNet-based query expansion for IR is not new, we believe our experiment to be the first that evaluates the individual impact of such expansion in a IR setting that corresponds to the task of typical web searches: huge, heterogeneous document collection, long, unstructured free text documents and very short, noun only queries.

In terms of results, [23] reported that query expansion with WordNet synsets actually worsened retrieval effectiveness, even with the semantically correct synonyms manually selected. In contrast, both [1, 19] reported opposite results, with such expansion improving retrieval effectiveness, and in the case of [1], even so without disambiguation. Our own positive results, shed a new light on these somewhat contradictory previous results: they suggest that the key factor that hampered query expansion to yield any benefit in experiment of [23] was neither the large, heterogeneous document collection, nor the long documents but precisely the long queries.

5. Conclusion

The semantic web holds much promise to bring about an overdue breakthrough in web search and the intelligent e-services (in business, retail, entertainment, education and so forth) that can be aggregated on top of it. However, we argue that it is still quite a distant goal and will be no silver bullet. Even when a critical mass of web documents will embed ontology-mediated semantic annotations, a large part of the web is bound to remain purely textual. Thus, intelligent retrieval of semantic content hidden in text-only documents should remain an important goal of AI research.

In this paper, we presented an experiment that empirically measured the impact on retrieval effectiveness of automatically expanding web search engine query keywords with their synonyms and/or hypernyms. We found that such expansion always improves average effectiveness for the complete set of retrieved documents. We also found that the best effectiveness gain is obtained when restricting expansion to synonyms of the most common word sense of each keyword, and that this is the case both for the entire set of retrieved documents (with F-measure improving 37%), and for the top 20, 30, 40 and 50 documents, (with precision respectively improving 1%, 12%, 17% and 37%). To our knowledge, this is the first experiment to date to evaluate the individual impact of a fully automated ontology-based query expansion technique in an experiment setting similar to real world web search. The encouraging results, suggest that some of the same knowledge sources can advantageously be used in a variety of ways for improving search effectiveness of both the semantic and non-semantic parts of the web.

In future work, we intend to reuse ExpanSyns to evaluate other knowledge base usages for improving effectiveness of purely textual documents, including query word sense disambiguation, morphological processing and using multiple ontologies. Another interesting direction is to measure the comparative and cumulative effectiveness gains obtained from such linguistic knowledge based techniques on the one hand, and collaborative filtering techniques on the other.

Acknowledgments

This research was supported by grants from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) of the Brazilian Federal Government.
References

