Semantic Access to Data from the Web

Raquel Trillo[1], Laura Po[2], Sergio Ilarri[1], Sonia Bergamaschi[2], and Eduardo Mena[1]

[1] IIS Dept., Univ. of Zaragoza, María de Luna 1,
Zaragoza, 50018, Spain.
{raqueltl,silarri,emena}@unizar.es
[2] II Dept., Univ. of Modena and Reggio Emilia,
Via Vignolese 905, Modena, 41125, Italy.
{po.laura,bergamaschi.sonia}@unimore.it

Abstract. There is a great amount of information available on the web. So, users typically use different keyword-based web search engines to find the information they need. However, many words are polysemous and therefore the output of the search engine will include links to web pages referring to different meanings of the keywords. Besides, results with different meanings are mixed up, which makes the task of finding the relevant information difficult for the user, specially if the meanings behind the input keywords are not among the most popular in the web. In this paper, we propose a semantics-based approach to group the results returned to the user in clusters defined by the different meanings of the input keywords. Differently from other proposals, our method considers the knowledge provided by a pool of ontologies available on the Web in order to dynamically define the different categories (or clusters). Thus, it is independent of the sources providing the results that must be grouped.

1 Introduction

The big explosion of the World Wide Web in the last fifteen years has made a great and ever-growing amount of information available to users. In this context, search engines have become indispensable tools for users to find the information they need in such an enormous universe. However, traditional search engine techniques are becoming less and less useful because there is simply too much information to search.

Thus, for example, the use of polysemous words in traditional search engines leads to a decrease in the quality of their output [1, 2]. For example, if a user inputs "mouse" as a keyword because he/she is interested in obtaining information about "the numerous small rodents typically resembling diminutive rats", the search engine can return a very high number of hits. Particularly, in this case Google returns about 218,000,000 hits[3]. However, the first hit that refers to the animal is on the third page of the results (the 36th hit), while the previous hits refer to other meanings of the word "mouse", such as "pointer

[3] Data obtained on 24th April 2009.

device", "non-profit organization that pioneers innovative school programs", different companies, etc. Unfortunately, users usually check only one or two pages of the results returned by the search engine [3]. So, an approach is needed to tackle this problem to provide the user with the information that he/she needs.

The problem is that hits referring to different meanings of a user keyword are usually mixed in the output obtained from a search engine. Moreover, the top positions in the ranking are occupied by the meanings which are the most popular on the Web (in the previous example, "pointer device"), hiding the huge diversity and variety of information available to the users on the Web. So, presenting the results to the user classified in different categories, defined by the possible meanings of the keywords, would be very useful, providing interesting advantages, as it is claimed in several previous works (e.g., [4, 5]). Thus, it helps the user to get an overview of the results obtained, to access results that otherwise would be positioned far down in a traditional ranked list, to find similar documents, to discover hidden knowledge, to refine his/her query by selecting the appropriate clusters, to eliminate groups of documents from consideration, and even to disambiguate queries such as acronyms. Summing up, it improves the user's experience by facilitating browsing and finding the relevant information.

In this paper we propose a new approach, based on semantic technologies, that is able to classify the hits in clusters according to the different meanings of the input keywords. Moreover, it discovers the possible meanings of the keywords to create the categories dynamically in run-time by considering heterogeneous sources available on the Web. As opposed to other clustering techniques proposed in the literature, our system considers knowledge provided by sources which are independent of the indexed data sources that must be classified. Thus, it relies on intensional knowledge provided by a pool of ontologies available on the Web instead of on extensional knowledge extracted from the sources to be grouped by means of statistical information retrieval techniques.

The structure of the rest of the paper is as follows. In Section 2 the main elements of our proposal are presented. Then, in Section 3 some possible improvements of the basic architecture are detailed. In Section 4 a methodology to evaluate our proposal is proposed, and in Section 5 some related works are presented. Finally, some conclusions and plans for future work appear in Section 6.

2 Architecture of the System

Along the last decades, different techniques to cluster documents have been proposed. However, traditional clustering algorithms cannot be applied to search result clustering [6, 4, 7] because, for example, it is not feasible to download and parse all the documents due to the need to provide quick results to the user. In the following, we first present the features that a clustering approach for web search should present. Then, we propose a basic architecture that, enhanced with the improvements described in Section 3, complies with these features. Finally, we explain the workflow of the system and illustrate it with an example.

2.1 Requirements of a Web Search Clustering Approach

Several works have identified the features that a suitable clustering approach in the context of web search should exhibit. Thus, in [6] the following six features are indicated:

- *Relevance*: the clustering approach should separate the pages relevant to the user's query from the irrelevant ones.
- *Browsable summaries*: it should provide informative summaries of the clusters generated.
- *Overlap*: one web page could belong to more than one cluster, as web pages may have overlapping topics.
- *Snippet-tolerance*: methods that need the whole text of the document should be avoided, due to the excessive downloading time that they would require, in favor of approaches that only rely on the snippets[4]. Moreover, in [6], the authors indicate that "Surprisingly, we found that clusters based on snippets are almost as good as clusters created using the full text [...]".
- *Speed*: clustering should be fast.
- *Incremental*: the clustering process should start as soon as some information is available, instead of waiting until all the information has been received.

Some authors emphasize or propose slightly different features; for example, [4] considers *coherent clusters* (which implies the need to generate overlapping clusters), *efficiently browsable*, and *speed* (*algorithmic speed* and *snippet-tolerance*). However, the previous six features are clearly a good representative.

2.2 Basics of the Proposed System

Considering the previous features, we have defined a basic architecture of a system, whose goal is to provide the user with data that satisfy his/her information needs, from the data obtained by a traditional search engine. We have also considered possible improvements to this basic approach, that we detail in Section 3, to totally fulfill the previous features. The proposed system performs two main steps (see Figure 1 for a general overview):

Step 1: Discovering the semantics of user keywords
In this step, the system needs to discover the intended meaning of the user keywords to only consider the hits returned by the traditional search engine that correspond with that semantics. So, it is required to find out the possible meanings (interpretations or senses) of each input keyword (ki) of the user, and then to select one interpretation for each keyword. In other words, a *Word Sense Disambiguation* (*WSD* [2]) algorithm is performed in two phases in run-time:

[4] A snippet is a segment that has been snipped off the document returned as a hit. Typically, it is a set of contiguous text around the user keywords in the selected document.

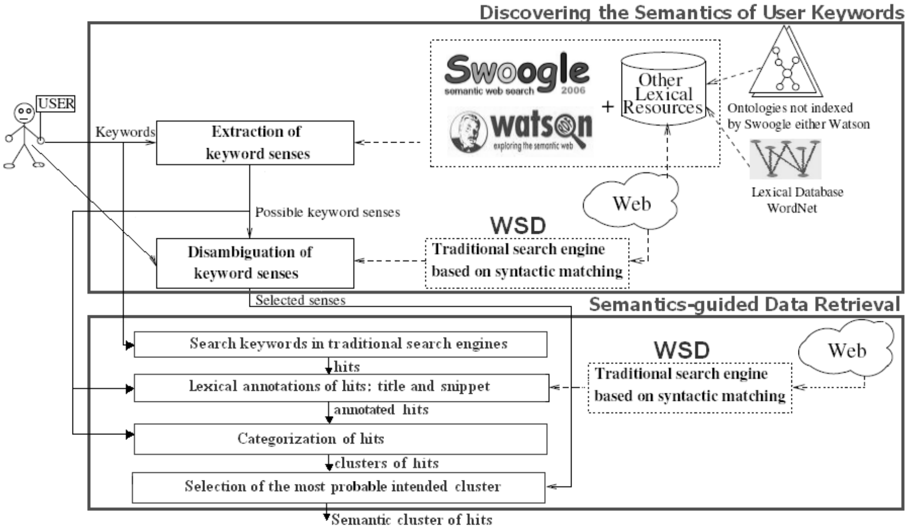
Discovering the Semantics of User Keywords
USER
Keywords
Extraction of keyword senses
Swoogle semantic web search 2006
watson exploring the semantic web
+
Other Lexical Resources
Ontologies not indexed by Swoogle either Watson
Lexical Database WordNet
Web
Possible keyword senses
WSD
Disambiguation of keyword senses
Traditional search engine based on syntactic matching
Selected senses
Semantics-guided Data Retrieval
Search keywords in traditional search engines
hits
Lexical annotations of hits: title and snippet
annotated hits
Categorization of hits
clusters of hits
Selection of the most probable intended cluster
Semantic cluster of hits
WSD
Web
Traditional search engine based on syntactic matching

Fig. 1. Overview of the approach

1. In the *Extraction of keyword senses* phase, the system has to obtain a list of possible meanings for each keyword ki (called *possible keyword senses* and denoted as {si1, si2, ...sin}), so semantic descriptions are required. These descriptions are provided by different sources such as thesauri, dictionaries, ontologies, etc., and extracting these descriptions from them is needed.
   At this point, we consider two possibilities to tackle this task:
   - *Consulting a well-known general-purpose shared thesaurus such as WordNet.* The main advantage of this option is that it provides a reliable set of the possible meanings of a keyword, and allows to share with others the result of the process. Moreover, the fundamental peculiarity of a thesaurus is the presence of a wide network of relationships between words and meanings. The disadvantage is that it does not cover with the same detail different domains of knowledge. So, some terms or meanings may not be present; for example, the term "developer" with the meaning "somebody who design and implements software" is not stored in WordNet. Moreover, new possible interpretations of a word appear along time; for example, life as "the name of the well-known magazine" or "the name of a film". These considerations lead to the need of expanding the thesaurus with more specific terms (this can be easily done using a MOMIS component, called WordNet Editor, which allows adding new terms and linking them within WordNet [8]). In contrast, other terms in the thesaurus may have many associated and related meanings; for example, the word "star" in WordNet has three very similar meanings "someone who is dazzlingly skilled in any field", "an actor who plays a principal role" and "a performer who receives prominent billing". Some

users could consider all these meanings as equivalent ones, whereas others could be interested into treating them as different ones.

- *Consulting the shared-knowledge stored in different pools of ontologies available on the Web and using synonym probability measurements to remove redundant interpretations.* For this task we perform the process defined in [9, 10] for each keyword. In short, these works rely on the following idea: the more ontologies consulted (each one representing the point of view of their creators), the more chances to find the semantics that the user assigned to the entered keywords. Firstly, the ontological terms that match the keyword (and also their synonyms) are extracted from the whole ontologies by means of several techniques describe in [9]. Then, the system treats the possible overlapping among senses (different terms representing the same meaning). For this task, the different terms extracted are considering as input of the following iterative algorithm. For each ontological term, its similarity degree with respect to the others terms is computed. If the similarity degree is lower than a threshold given as parameter (synonym threshold), the two terms are considered to represent different meanings of the keyword; otherwise, they are considered synonyms (they represent the same interpretation of the keyword) and they are merged (integrated) into a single sense following the techniques described in [9], in order to form a new multi-ontological term that will be compared with the rest of terms. Finally, the output of the process is a set of integrated senses, where each element corresponds to a possible meaning of the keyword. The main advantage of this approach is the use of an unrestricted pool of ontologies available on the web because it maximizes the possible interpretations for the keywords; for example, the meaning of “developer” related to ”computer science” appears in some ontologies[5]. Moreover, it allows new interpretations of the words to be considered without extra effort. The system can also be set up to work with different synonym thresholds so the output can be dynamically changed. Notice that if the synonym threshold is risen, then the integration of terms decreases, so more fine-grained senses are provided as output, whereas if the synonym threshold decreases there is a higher degree of integration and less interpretations are provided for the same keyword. The main disadvantage of this approach is that it could introduce noise and irrelevant information. Besides, the complexity of the process could have a high cost in time, decreasing the average speed.

The trade-off between the two previous approaches is not totally clear. Therefore, we decide to begin the implementation of the proposed architecture considering only WordNet for simplicity reasons, and then to replace the keyword senses extraction module by the techniques described in [9, 10] and to compare the performance of the two approaches.

[5] For example: http:www.mindswap.org2004SSSW04aktive-portal-ontology-latest.owl, http:keg.cs.tsinghua.edu.cnontologysoftware and http:tw.rpi.eduwikiSpecial:ExportRDFCategory:Software_Developer.

2. In the *Disambiguation of keyword senses* phase, the system selects the most probable intended meaning for each user keyword (called *selected senses*, denoted by s1x, s2y, ..., snz). Many features can be considered to discover the more suitable sense of a word in the context of a document written in natural language; however, the disambiguation process is more complex when a no so-rich context is available, as in this case where the input is a list of plain keywords, so we cannot take advantage of the syntax of a whole sentences or the collocation of the words. Moreover, user queries in keyword-based search engines normally has a length lower than five words. Nevertheless, even under these circumstances, in many cases for a human it is possible to select/discard meanings of the polysemous words. Thus, for example, the word star is expected to be used with the meaning "celestial body" when it appears with the word "astronomy", and with the meaning "famous actor" when it appears with the word "Hollywood". Therefore, we try to emulate this behavior by taking into account the context in which a keyword appears, i.e. the possible meanings of the rest of user keywords, to select its most probable intended meaning. In order to do that, we consider the use of the semantic relatedness measure based on information provided by traditional syntactic search engines and the disambiguation method defined in [1, 11]. However, the proposed architecture does not depend on a specific WSD method because not all WSD methods are valid or perform well in all possible contexts, as it is said in [12]. So, other approaches such as a *Probabilistic Word Sense Disambiguation* (*PWSD*) [12, 13] or using the profile of the user or other information available in the disambiguation process could be considered. In addition, it is also a possibility to ask for user intervention when the context is highly ambiguous (even for humans) as for example "life of stars".

Step 2: Semantics-guided data retrieval step

The goal of this step is to provide the user with only the hits retrieved by a traditional search engine, such as Google or Yahoo!, in which he/she is interested and filter out the irrelevant results. In other words, the system must select the hits that have the same semantics as the intended meaning of the user keywords and discard the others. This process is performed in four phases in run-time:

1. The first phase requires performing a traditional search of hits on the Web, by taking as input the user keywords and using a traditional search engine such as *Google* or *Yahoo!*. This search returns a set of relevant ranked hits, which represent the web pages where the keywords appear. The order of the hits, i.e. the ranking of the results provided by the search engines used, depends on the specific techniques used by the engine for that task and its numerous internal parameters. At first we only consider the first 100 hits returned for the search engine to be provided as input of the next phase. Notice that this process can be performed in parallel with the *Discovering the semantics of user keywords* step.

2. In the *Lexical annotation of hits* phase, each of the hits (composed of a title, a URL and a snippet), obtained in the previous phase, is automatically annotated lexically. In more detail, firstly, each returned hit goes through a cleansing process where stopwords are filtered out; then, a *lexical annotation* process is performed for each hit[6]. Each user keyword that appears in each filtered hit is marked with the most probable keyword sense by considering the context in which it appears (i.e., by considering its relevant neighbor words) and using WSD methods. As in the *Disambiguation of keywords senses* phase, here it is also possible to consider other WSD approaches such as PWSD[7] (see Section 3 for more details). In this way, a more realistic approach would be considered, as it is usually very difficult even for a human to select only one meaning of the word when the context is limited. Besides, for each keyword we consider the possible meanings obtained in Step 1 and also a new *unknown meaning*. This allows the system to take into account the evolution of a natural language: new senses for a keyword appear when these senses start being used, and only after these senses become widespread they will be integrated in the resources where the semantic descriptions are provided. So, when a user keyword within the snippet of the hit cannot be annotated, it is assumed that it corresponds with the unknown sense.

   It should be emphasized that only the information provided by the snippets of the hits is used in this step, without accessing to the full document by means of the URL provided. This is because an approach that needs to download the whole documents would be unsuitable for clustering web search results, due to the requirement to provide a quick answer to the user (see Section 2.1). Besides, to further reduce the processing time, we initially propose considering only the first 100 hits returned by the search engine[8]. Moreover, if needed, several hits could be annotated in parallel by using multiple computers/processors. Notice that this process can be performed in parallel with the *Disambiguation of Keyword Senses* phase, but it is needed that the *Extraction of keywords senses* had finished.
3. In the *Categorization of hits* phase, the hits, annotated in the previous phase, are grouped in clusters/categories by considering their lexical annotations. Firstly, the system defines the categories that are going to be considered. The potential categories are defined by the possible combinations of senses for the input keywords. For example, if the user introduces two keywords ($k1$

[6] A lexical annotation is a piece of information added to a term that refers to a semantic knowledge resource such as dictionaries, thesauri, semantic networks or others which express, either implicitly or explicitly, a general ontology of the world or a specific domain.

[7] Probabilistic Word Sense Disambiguation automatically annotates keywords and associates to any lexical annotation a probability value that indicates the reliability level of the annotation. Besides, it is based on a probabilistic combination of different WSD algorithms, so the process is not affected by the effectiveness of single WSD algorithms, in a particular context or application domain.

[8] We plan to perform experiments to determine a good value for the number of hits to analyze, by studying the trade-off between the recall and the answer latency.

and $k2$) and, in the previous step, two senses are discovered for $k1$ ($S11$ and $S12$) and one sense for $k2$ ($S21$), then the following potential categories are considered: (S11, S21), (S12, S21), (U1, S21), (S11, U2), (S12, U2), where $U1$ and $U2$ represent the unknown meanings considered for the keywords $k1$ and $k2$ respectively. Secondly, each hit has to be assigned to the category defined by the meanings of the input keywords corresponding to the lexical annotation of that hit. The hits of each category are ordered following the ranking returned by the search engine. Therefore, popular hits within a category will appear first, as they reference well-known web pages with topics belonging to that category. Notice that, in this way, only considering the basic architecture, a hit is only assigned to one category (the most probable one). Besides, categories will be labeled with the definitions of the corresponding meanings and potential categories that are allocated no hits will be discarded. This process can be performed in parallel with the disambiguation of keyword senses, but once the *Extraction of keywords senses* phase has finished. Moreover, the different hits can also be classified in parallel.

4. Finally, the system considers the result of the *Disambiguation of keyword senses* phase, where a sense is obtained for each keyword, and the cluster that corresponds with these selected senses is provided to the user. We also advocate in this step the use of *AJAX* to improve the user's experience (the clustering can be performed in background while the user uses the system).

It would be also possible to consider a different final output for the semantics-guided data retrieval, such as presenting all the hits obtained by traditional search engines grouped in categories considering the different meanings of the input keywords. For more details, see Section 3.

2.3 Workflow of the System and Running Example

In this section, we will illustrate the steps performed by our system with an example. Let us assume that the user is interested in the way that the jaguar species has evolved and he/she enters the input keywords "jaguar" and "evolution", which we will denote by $k1$ and $k2$ respectively. The following steps take place (see Figure 2, which emphasizes the steps that can be performed in parallel):

- The semantics of the user keywords are discovered (right part of Figure 2). For this, first the possible senses of each of the input keywords are obtained. For simplicity, let us assume that the system uses a well-known general purpose shared thesaurus (another alternative was discussed in Section 2.2). WordNet 3.0 gives just one possible meaning $S11$ for the word "jaguar" (the animal) and two meanings $S21$ and $S22$ for "evolution" ("a process in which something passes by degrees to a different stage" and "the sequence of events involved in the evolutionary development of a species or taxonomic group of organisms"). Then, the disambiguation process selects $S11$ and $S22$ as the most probable meanings of $k1$ and $k2$.

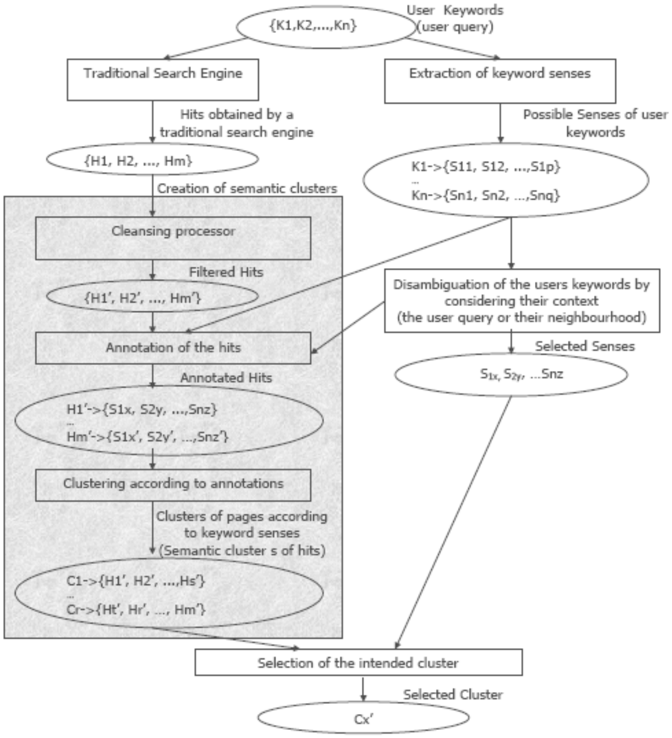

Fig. 2. Workflow indicating the steps performed for identification of the user's needs

- Data are retrieved based on the semantics (left part of Figure 2). First, the input keywords are entered in a traditional search engine and the first 100 hits (title, URL, and snippet) are cleaned and collected. Then, each hit is annotated by marking each keyword appearing in the hit with its most probable sense. The keyword $k1$ can be annotated with two possible senses ($S11$ and $U1$) and the keyword $k2$ with three possible senses ($S21$, $S22$, and $U2$), by considering the possible senses discovered for the keywords and the unknown senses. As an example, in the first hit returned by Google[9], with title "Jaguar Evolution premium Class 5.25 inch Scissors" (a model of scissors), $k1$ is annotated with $U1$ and $k2$ with $U2$. In the second hit, with title "Jaguar Evolution Poster at AllPosters.com" (a poster showing the evolution of different models of the Jaguar car), $k1$ is annotated with $U1$ (the meaning of jaguar as a type of car is not available in WordNet) and $k2$ is annotated with $S21$. As a final example, in the 20th hit, with title "Paleontology (Dinosaurs): Evolution of Jaguars, lions and tigers...", $k1$ is annotated with $S11$ and $k2$ with $S22$; this is the first hit that is actually relevant for the user. These hits are then grouped in categories defined by

[9] Data obtained on 12nd May 2009.

the possible combinations of senses. Two hits are grouped under the category characterized with $S11$-$S22$, three hits within the category $S11$-$U2$, 36 hits within $U1$-$S21$, 2 within $U1$-$S22$, and 57 within $U1$-$U2$. Finally, the cluster with the intended meaning (in this case, the cluster with the annotations $S11$ and $S22$) is presented to the user.

As illustrated in the previous example, our approach saves a lot of effort to the user in locating the relevant hits. Only the hits in the positions 20th and 63rd are relevant for the user and presented immediately to him/her within the relevant cluster.

3 Improvements of the Basic Architecture

In this section, we describe two improvements for the basic architecture presented before. First, we propose to use a *Probabilistic Word Sense Disambiguation* to consider the probabilities of the different senses of a keyword instead of just using the most probable sense. Second, we advocate considering synonyms of the user keywords to increase the coverage of the results returned to the user.

3.1 Probabilistic Word Sense Disambiguation

In the basic architecture described in the previous section, the system selects the most probable meaning for each user keyword and presents to the user the cluster with the hits that correspond to those meanings. However, trying to select the most probable sense could be risky. Indeed, in many cases, even the user himself/herself may find difficulties to assign a single meaning to his/her keywords. An alternative approach could make use of PWSD techniques to assign probabilities to the different senses of the keywords. Based on this idea, we suggest three possible ways to improve our proposal:

1. *Show more interpretations to the user.* This is a slight variant of the approach described in Section 2.2 where, instead of returning just one cluster to the user, which corresponds with the selected senses in the *Disambiguation of Keyword senses* phase, the system shows all the categories considered containing hits. Thus, this approach recognizes the possibility that the user needs to explore clusters other than the most probable one. For example, if the user keywords are k1 and k2, and k1 has three possible meanings (S11, S12, S13) and k2 two possible meanings (S21 and S22), then the system retrieves twelve combinations/interpretations (taking into account the unknown senses). The order in which the different created clusters are shown depends on the probability of the selected senses that represent each category.
2. *Multi-classification.* In this case, the user keywords and categories are interpreted as in the previous case. However, we adopt a different approach to classify the hits. Thus, during the disambiguation of the keywords that

appear in the hits, we assign not just one meaning for each keyword but a list of possible meanings (it can be a ranked list if we are using a PWSD). So, depending on the meanings that are assigned to a keyword in a hit, the hit can be classified in different clusters. As a result, selecting different interpretations of the user keywords, the user can retrieve the same hit.

3. *Multi-classification with ranking.* In this case, the hits are clustered as in the previous case using a PWSD technique but, in addition, the hits in each cluster are ordered by considering their annotated probabilities and not only the ranking of the search engine from which they are retrieved, as in the previous case. During the *Lexical Annotation* phase the PWSD can retrieve different meanings for each user keyword in the snippet or title of a hit, each one associated with a probability value. So, an interpretation/combination of meanings in a hit (for example S11, S22) is associated to a probability computed as the joint probability of the meanings (Prob(S11) * Prob(S22)). This probability can be combined with the rank retrieved by the traditional search engine considered, providing a new rank of hits. As a result, selecting an interpretation of the user keywords, the user can retrieve a ranked list of hits by considering their lexical annotations.

We consider that these techniques will be useful to facilitate the user in the task of finding the relevant web pages. An experimental evaluation is needed to compare these approaches.

3.2 Retrieval of Synonyms of User Keywords

Important improvements can be achieved by considering synonyms of the user keywords to retrieves more pages to be classified in the different categories [14]. Based on the meanings of the keywords and the semantic resources used, the system is able to identify words that are synonyms. Using these synonyms as input to the search engine, it is possible to retrieve new relevant hits. For example, the words "truck" and "lorry" are synonymous. If we enter "truck" in Google we obtain about 166,000,000 hits, but if the user enters "lorry" the number of hits is approximately 5,220,000. Therefore, if the system considers the synonyms of the user keywords and searches for both "truck" and "lorry" it will provide a better coverage of the relevant results.

It should be noted that if a word used to extract new hits is a polysemous word then the system needs to discard the hits that do not correspond with the intended meaning of the user. For example, if a user inputs "actor" he/she is also probably interested in "star" (as in "film star"). However, there will be some hits with the input keyword start that are irrelevant for the user, such as those hits containing information about the "celestial body"; only hits with the sense "someone who plays a main role in a film" will be interesting for the user. So the systems also performs a retrieval by using these synonyms and, after that, it must lexically annotate the snippets obtained in this enrichment and filter out those annotated hits with a no relevant meaning. Besides, the hits returned will be grouped in the same clusters, so the number of clusters will not be affected and more relevant hits will be retrieved for each cluster.

It should be emphasized that this *synonym expansion approach* enhances the recall of the system especially in those cases where the user selects keywords that are not so common (instead of synonyms that are more popular).

4 Experimental Evaluation Methodology

At this point, we ask ourselves how this system can be evaluated. As indicated in [5], there is no general consensus about a good metric to evaluate the quality of the search result clusters; besides, comparing different clustering and search engines approaches is not easy because most of them have different goals (oriented to a specific topic vs. general coverage, allowing queries whose length is one word vs. several words, etc.). Despite this fact, three different methodologies to evaluate a clustering of search results are identified in [5]:

- *Anecdotal evidence* for the quality of the results. Following this methodology, we could evaluate the quality of the results by analyzing the way in which the users behave when using the system. Collecting significant anecdotal evidence is quite challenging.
- *User surveys* with different input queries. This approach suggests asking the users to evaluate the relevance of the results returned for different sample queries. This approach has also some shortcomings, such as the difficulty to find a statistically significant number of users or sample queries.
- *Mathematical measures* [5, 15]. These methods propose the use of metrics to evaluate the quality of the clusters. There are also some difficulties inherent to these methods, such as defining measures that consider the content of each cluster and not only the expressiveness of its labels. Some metrics that could be considered included in this category are: distance measurements between clusters (or ideal distributions), *entropy*, the *Rand Index*, the *mutual information*, *purity*, *precision* and *recall*, etc.

Based on these ideas and the experimental evaluation realized in similar works [16, 14], we plan to perform an evaluation of our system as follows. First, we will define several sets of user keywords, representing different queries. Then, we will enter each of these sets of keywords into a standard search engine and record the first results obtained (the first 100 hits for each user query). The third step involves performing a survey with real users. For each query, we will present the following information to the user: the list of words in the set, the possible meanings of these words (as identified by our knowledge bases), the list of categories created by the system, and the hits returned by the search engine (title and snippet). Then, the user will have to manually classify the hits in categories defined[10]. With all the groups manually performed by the users,

[10] The user will be reminded that a hit can be classified in different categories simultaneously.

we will obtain a probabilistic grouping of the results: a set of groups with hits tagged with the probability that the hit belongs to that group (estimated as the percentage of users that classified the hit under that group). We will compare this "ideal" grouping with the one obtained by our approach, using mathematical measures mentioned above. The membership probabilities of the hits in the ideal distribution can be used either to remove from its clusters those hits with a probability lower than a predefined threshold or to compute the mathematical measures by considering also the difference between the probabilities in the ideal and the computed distribution. Besides these surveys, we also plan to present the user with the output from other clustering engines and use a set of questions to perform a qualitative comparison.

5 Related Work

The clustering of data is a problem that has been studied for a long time and applied in many different application scenarios, based on the so-called *Cluster Hypothesis* that relevant documents tend to be more similar to each other than to non-relevant documents [17, 5] (i.e., there is a high intra-cluster similarity and a low inter-cluster similarity [5]). In particular, several methods have been proposed to cluster collections of documents (hierarchical methods [18], K-means-based approaches [6], etc.). As opposed to *classification methods* [19, 20], *clustering methods* do not require predefined categories (the categories are discovered during the clustering process) and are therefore more adaptive to different types of queries [7, 5]. It may seem that our approach considers a set of predefined categories (the meanings of the user keywords); however, the potential categories used to allocate the hits retrieved are dynamically obtained depending on the user keywords and the knowledge bases queried, and besides they can be merged or refined by considering a synonym threshold given as parameter.

In the context of the web, document clustering can be either pre-computed over a complete collection of documents (e.g., [21]) or computed on-the-fly considering only the documents returned as a result of a search. The latter case (called *ephemeral clustering* in some works, such as [22]),which our proposal is based on, leads to better results because it focuses only on the documents that are considered relevant (hits) for the user's query [17, 4, 23]. Besides, it adapts more naturally to the fact that the Web is constantly evolving.

Several search (or meta-search) engines that perform clustering of results are available on the Web, such as *Clusty* (`http://clusty.com`) by *Vivisimo* (`http://vivisimo.com`, a Carnegie Mellon University's software spin off), *Grokker* (`http://www.grokker.com`), *Armil* [24] (`http://armil.iit.cnr.it`), *iBoogie* (`http://iboogie.com/`), *Cuil* (`http://www.cuil.com`), *JBraindead* (`http://bender.lsi.uned.es:8080/ModuloWeb/jbraindead.html`), *WebCat* [25] (`http://ercolino.isti.cnr.it/webcat`), and *SnakeT* [5, 26] (*SNippet Aggregation by Knowledge ExtracTion*, `http://snaket.di.unipi.it`). Unfortunately, most of the previous search engines are commercial products and little technical details are available about their implementation. From the previous works, *Clusty* is considered the state-of-the-art in many researches in this area [27, 28, 5, 29].

It is not our goal to provide an in-depth or exhaustive study of existing clustering approaches oriented to web-snippets; for that, we refer the interested readers to [5]. Nevertheless, it should be emphasized that our approach distinguishes itself from other proposals because of its ability to use, not the extensional knowledge provided by the data that have to be clustered by means of classical techniques from Information Retrieval, but the intensional knowledge provided by sources which are independent of the indexed data sources. Thus, we precisely consider the intended semantics of the keywords introduced by the user. Some other proposals that also use semantic techniques are [14, 16]. However, in [14] only WordNet is used and it is assumed the existence of a predefined set of categories. In [16], the use of WordNet is also proposed but they cluster different senses of a word. Nevertheless, this system is limited to queries with a single keyword and does not allow overlapping categories (i.e., hits being classified in more than one category). These limitations are avoided with our proposal.

6 Conclusions and Future Work

We have presented a semantics-based approach to group the results returned by a standard search engine in different categories defined by the possible senses of the input keywords. Our proposal satisfies desirable features identified in the literature [6] for this kind of systems: 1) relevance: the hits that are probably more relevant for the user's query will be presented in a single cluster at the top of the ranked list of the groups identified; 2) browsable summaries: each cluster will be labeled with the selected meanings of the user keywords and with the snippet of the most representative hit; 3) overlap: a single hit can be classified in different categories by considering Probabilistic Word Sense Disambiguation (PWSD); 4) snippet tolerance: only the snippets of the hits are used to form the groups, without accessing to their whole corresponding document; 5) speed and incremental: we have proposed several techniques to provide results to the user, such as parallel processing and the use of AJAX-based techniques to perform some processing in the background, while the user is interacting with the system.

The next step will be devoted to the development, implementation, and testing of the system proposed. Besides we consider to adapt the approach to be used against heterogeneous structured or semi-structured data repositories such as data bases or UDDI repositories following the ideas of Bernstein et al. [30], that consider that a keyword search approach can be used against structured data to get a quick feel for what is available and set the stage for a more precise integration. The following topic must be also studied in more detail. Clustering the results provided by traditional search engines by considering the meanings of the keywords facilitates the search of information by the users but it is not enough. Thus, even when the semantics of the keywords in both the hits and the user keywords have been identified, the user could be looking for several different queries. For example, if a user inputs "fish" and "person" and indicates her/his interest in the cluster whose meanings are “a creature that lives and can breathe in water” and “a human being”, she/he could be trying to find information for either mermaids (“a fabled marine creature, typically represented as having the upper part like that of a woman, and the lower like a fish”) or fishermen/fisherwomen (“people that earn their living fishing”) and these options are mixed yet. So, one more phase must be included to deal with these situations.

Acknowledgements

We would like to thank A. Illarramendi for her valuable comments. This work was partially supported by the CICYT project TIN2007-68091-C02-02, the MUR FIRB Network Peer for Business project (http:www.dbgroup.unimo.itnep4b), and the IST FP6 STREP project 2006 STASIS (http:www.dbgroup.unimo.itstasis).

References

1. Gracia, J., Trillo, R., Espinoza, M., Mena, E.: Querying the web: A multiontology disambiguation method. In: Sixth Int. Conf. on Web Engineering (ICWE’06), Palo Alto, California, USA, ACM (July 2006) 41–248
2. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys **41**(2) (2009) 1–69
3. Jansen, B.J., Pooch, U.: A review of web searching studies and a framework for future research. Journal of the American Society of Information Science and Technology **52**(3) (2000) 235–246
4. Zamir, O., Etzioni, O.: Grouper: a dynamic clustering interface to web search results. In: Eighth International World Wide Web Conference (WWW’99), Toronto, Canada, Elsevier (May 1999) 1361–1374
5. Gullí, A.: On Two Web IR Boosting Tools: Clustering and Ranking. PhD thesis, Dipartimento di Informatica, Universit‘a degli Studi di Pisa (May 2006)
6. Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: 21st Int. Conf. on Research and Development in Information Retrieval (SIGIR’98), Melbourne, Australia, ACM (August 1998) 46–54
7. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: Learning to cluster web search results. In: 27th Int. Conf. on Research and Development in Information Retrieval (SIGIR’04), Sheffield, United Kingdom, ACM (July 2004) 210–217
8. Benassi, R., Bergamaschi, S., Fergnani, A., Miselli, D.: Extending a lexicon ontology for intelligent information integration. In de Mántaras, R.L., Saitta, L., eds.: ECAI, IOS Press (2004) 278–282
9. Trillo, R., Gracia, J., Espinoza, M., Mena, E.: Discovering the semantics of user keywords. Journal on Universal Computer Science (JUCS). Special Issue: Ontologies and their Applications, ISSN 0948-695X **13**(12) (December 2007) 1908–1935
10. Gracia, J., d’Aquin, M., Mena, E.: Large scale integration of senses for the semantic web. In: Proc. of 18th International World Wide Web Conference (WWW’09), Madrid, Spain, ACM, ISBN 978-1-60558-487-4 (April 2009) 611–620
11. Gracia, J., Mena, E.: Web-based measure of semantic relatedness. In: Ninth Int. Conf. on Web Information Systems Engineering (WISE’08), Auckland, New Zealand. LNCS (September 2008) 136–150

12. Po, L.: Automatic Lexical Annotation: an effective technique for dynamic data integration. PhD in Computer Science, Doctorate School of Information and Communication Technologies, University of Modena e Reggio Emilia, Italy (2009)
13. Po, L., Sorrentino, S., Bergamaschi, S., Beneventano, D.: Lexical knowledge extraction: an effective approach to schema and ontology matching. In: ECKM. (2009) accepted for publication.
14. Hao, T., Lu, Z., Wang, S., Zou, T., GU, S., Wenyin, L.: Categorizing and ranking search engine's results by semantic similarity. In: Second Int. Conf. on Ubiquitous Information Management and Communication (ICUIMC'08), Suwon, Korea, ACM (January-February 2008) 284–288
15. Wu, J., Chen, J., Xiong, H., Xie, M.: External validation measures for K-means clustering: A data distribution perspective. Expert Systems with Applications **36**(3, Part 2) (2009) 6050–6061
16. Hemayati, R., Meng, W., Yu, C.T.: Semantic-based grouping of search engine results using WordNet. In: Eight Int. Conf. on Web-Age Information Management (WAIM'07), Huang Shan, China. LNCS (June 2007) 678–686
17. Hearst, M.A., Pedersen, J.O.: Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In: 19th Int. Conf. on Research and Development in Information Retrieval (SIGIR'96), Zurich, Switzerland, ACM (August 1996) 76–84
18. Zhao, Y., Karypis, G.: Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery **10**(2) (2005) 141–168
19. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34**(1) (2002) 1–47
20. Kules, W.M.: Supporting exploratory web search with meaningful and stable categorized overviews. PhD thesis, University of Maryland at College Park (2006)
21. Haveliwala, T.H., Gionis, A., Indyk, P.: Scalable techniques for clustering the web. In: Third International Workshop on the Web and Databases (WebDB'00). (2000)
22. Maarek, Y.S., Fagin, R., Ben-Shaul, I.Z., Pelleg, D.: Ephemeral document clustering for web applications. Technical Report RJ 10186, IBM Research (2000)
23. Tombros, A., Villa, R., Rijsbergen, C.J.V.: The effectiveness of query-specific hierarchic clustering in information retrieval. Information Processing and Management: an International Journal **38**(4) (2002) 559–582
24. Geraci, F., Pellegrini, M., Pisati, P., Sebastiani, F.: A scalable algorithm for high-quality clustering of web snippets. In: 21st Annual ACM Symposium on Applied Computing (SAC'06), Dijon, France, ACM (April 2006) 1058–1062
25. Giannotti, F., Nanni, M., Pedreschi, D., Samaritani, F.: WebCat: Automatic categorization of web search results. In: 11th Italian Symposium on Advanced Database Systems (SEBD'03), Cetraro (CS), Italy, Rubettino Editore (June 2003) 507–518
26. Ferragina, P., Gulli, A.: A personalized search engine based on web-snippet hierarchical clustering. Software Practice & Experience **38**(2) (2008) 189–225
27. Osiński, S.: An algorithm for clustering of web search results. Master's thesis, Poznań University of Technology, Poland (June 2003)
28. Wang, X., Bramer, M.: Exploring web search results clustering. In: 26th SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intelligence (AI'06), Cambridge, United Kingdom. (December 2006) 393–397
29. Geraci, F., Pellegrini, M., Maggini, M., Sebastiani, F.: Cluster generation and cluster labelling for web snippets: A fast and accurate hierarchical solution. In: String Processing and Information Retrieval (SPIRE'06), Glasgow, United Kingdom. LNCS (October 2006) 25–36
30. Bernstein, P.A., Haas, L.M.: Information integration in the enterprise. Communununications of the ACM **51**(9) (2008) 72–79