The context as a determinant of semantics: use of the local context of a personal computer to conduct searches on the Web

Student: Doina Alexandra Dumitrescu
doina.dumitrescu@uam.es
Advisor: Simone Santini
simone.santini@uam.es

Master Thesis
Escuela Politécnica Superior, Universidad Autónoma de Madrid
November 2008

Contents

List of Figures

Abstract

The large quantity of information accessible across the Internet has raised in a new and urgent form the information retrieval problem. The search engines, services that based on keywords return a list of documents more or less relevant, represent a first response to this problem. Nevertheless, in spite of their great utility, the search engines suffer from several limitations that affect the precision of the results of the queries and, therefore, their utility. The problems arise from the fact that the current search engines use, for the most part, simple keyword matching and often the results list is so large that the user has to see many irrelevant documents or try various queries before finding what s/he was looking for; both options - to go through a long page-by-page browsing or to reinitiate the search with a new query - are time consuming and inefficient.

In the last years the hypothesis of improving the results of the access to Web information by going beyond the superficial form of the documents, towards a *semantic* search, in other words a search that, more that in the linguistic content of a document, will focus in its meaning, has gained ground. The principal problem for implementing such a plan is that the determination of the meaning is notoriously problematic. As an example of the many problems that affect the possibility of a search involving the meaning based on the original text, here we can mention the polysemy problem. Polysemy problem means that a word has more then one meaning. The fact that a document contains a word that one finds in the given query it does not guarantee that the document is of interest for the user, because the word can have in the document a different meaning from what the user meant with the query.

Solutions based on formal annotations, like the ones that are being proposed in the semantic web, suffer important limitations since they do not take into account the context in which the search is performed. In this work we present a search system based on the idea that each search is performed in the context of a certain activity, context which will be formalized.

1 Introduction

The Internet is becoming an increasingly preeminent source of information that not only increases in volume very fast [4], but, the statistics show, is used every day by more people. In the past eight years (2000 − 2008), as Internet World Stats [5] shows, the number of Internet users has increased by 305.5%, a fact that confirms not only an easier access to the Web but also the richness of information that one can find in it. The internet is today a truly impressive reservoir of information, so much so that, finding what one is after has become a challenge, one that needs the intervention of sophisticated technical instruments. There are two main tools to find information on the Internet: browsing and searching [51]. Browsing entails navigating special sites organized according to a category hierarchy. This can be a bit inefficient when the user wants to access quickly the data that satisfies his information needs. In this case a better and more common solution is the use of search engines [27]. Search engines, whose use has grown by 10% in 2006 [1], usually start from a few keywords that the user types and return a list of results. Statistics show that 93% of Internet traffic is generated by a search engine and that 81% of Internet users use a search engine [1]. In other words, search engines have become a fundamental component of the organization of the Web, so much that for many of us a Web without search engines would be almost inconceivable. Still, current search engines are far from perfect. They use, for the most part, simple keyword matching, which makes them very sensible to linguistic phenomena like polysemy and synonymy [24], which affect the precision of the results of the queries. (Polysemy refers to a word that has multiple meanings and synonyms to different words that have the same meaning.) The problem arises from the ambiguities that those phenomena entail, which lead to wrong results if we ignore them in the searching process. In addition, "the Web queries do not represent all information needs" [9] in that many results are irrelevant for the users [52, 37, 40, 48]. For an information retrieval system to be successful is not enough to develop new retrieval techniques or models of Information Retrieval (IR) without taking into account issues concerning the user [12] and for quite some time efforts of establishing the user's information need had been done. Furthermore, the results list is often so large that the user has to see many irrelevant documents or try various queries before finding what s/he was

looking for; both options - to go through a long page-by-page browsing or to reinitiate the search with a new query - are time consuming and inefficient. Nevertheless there are exceptions: the web search engines can be very efficient in the case of very popular queries like giving a certain address in the US and retrieving in the first results the corresponding map, giving a URL or/and finding the right home page [9]. Those can be good examples of queries that will function well but these cases are relatively few.

One way to improve the quality of the results could be to analyze the linguistic structure of the query text in search of a better indication of the users desires. Linguisitic analysis extracts from the query the main concepts and the relationships between them, in order to obtain a *deep semantic structure*[21] that will be used to search for relevant documents. There are a few problems in this approach that we could divide in three classes: lexical, syntactic and semantic. The lexical problems refer to the interpretation of certain words or sentences and it consists mainly of the ambiguity introduced by polysemy. The syntactic problems derive from the structural relation between words or sentences. The semantic problems appear when a sentence is syntactically correct but its meaning is not clear. For example consider the query sentence:

Who held the philosophy classes in Berlin when Hegel retired?

This phrase presents semantic problems as Hegel never retired: he died from a gastrointestinal disease when he was still working.

A more general problem of the linguistic approach has to do with user interaction and the kind of information one is looking for. Many times, a user is not after some very specific piece of information like the one represented by the previous query. Rather, one is looking for some information about "Hegel" and, in these cases it is more common to just type "Hegel" rather than using a complicate sentence. Therefore, even though using the deep structure will improve the precision of the results (i.e. minimize the number of irrelevant documents), there are still some inconvenient. First, one essential condition for those systems to work is that the query has to be entered in the form of complete sentences. In this respect [13] has come to the conclusion that the users make a difference, linguistically speaking, in the way they communicate with a computer or with other people. This could be a problem at the time of the query as the majority of web users are

used to find information by typing a couple of simple keywords, they might not know exactly what they are looking for or the query is generic enough to not need a complex sentence. Second, there are linguistic problems since a computer is not able to understand how users associate meaning to words and will always be limited in the way it understand the true meaning of the user's sentence.

Another possible solution could be to ask the user for more information in terms of extending the query using explicit words that better express their real search intent [47]. However the general tendency among web users is to use no more than three keywords [67, 66, 43, 38] that are not capable of providing a complete description of the information need and cannot always disambiguate ambiguous words. In addition, 89% of users use for the first time a search service [1] and users seldom take advantage of query expansion [39]. Studies concerning interactive query expansion have shown that normally users find it difficult to choose the most useful terms [50, 56], and even for experienced users it is difficult to define their query precisely [18]. In conclusion, explicitly adding a precise description of contextual information is a problematic task that implies a certain effort on the part of the users who have to create large queries. Therefore in order to make it easier for the user and not lose the advantages of query expansion, it is more realistic to try to do automatic query expansion.

1.1 Definition of context

Retrieval results could be improved by involving the user context in the searching process, leading to a better understanding of the user's information needs. Context is an important concept in different computer science fields, but we will focus only on its role in information retrieval. In a non-contextual search, the same query is treated in the same way by a web search engine even if it is made by two distinct persons for whom the query means different things, or by the same person but at different moments of time in which the context and therefore the expected results have changed. For example, in the case of a query consisting of the keyword "bank", the results will be the same for a businessman searching for the latest news in investment or for a biologist searching for information on the ecosystem of river banks for his latest research paper; also, the results will be the same for the biologist

even though at one moment in time his interests is rivers and at another moment he wants to gather information to ask for a loan. It is not possible to understand what the user is really looking for unless we take into account the context in which the query is made. Spink and Saracevic [68] mentions the significance of the user's context in an information demand. In addition, [23] states the importance of the context: "A searcher's context affects how they interact with a retrieval system, what type of response they expect from a system and how they make decisions about information objects they retrieve.". The user context in the search process seems to alleviate some of this problems.

Using the context for improving the search results represents a big opportunity for improving search and a considerable technical challenge. On the one hand, one has to discover ways to identify the contextual information that makes the best improvement in the relevance of the results, and on the other hand one has to find efficient ways to use it in the retrieval process.

In this thesis, we shall propose a system for making contextual searches on the Web that attempts to seize some of these opportunities, and to solve some of these technical challenges. Before we do so, however, we need to clarify a little better the concept of *context* and its role.

If we check the Merriam-Webster Dictionary [6] we will find that the word "context" has basically two meanings:

1. the parts of a discourse that surround a word or passage and can throw light on its meaning, and

2. the interrelated conditions in which something exists or occurs: environment, settings.

The Cambridge Advanced Learner's Dictionary [2] gives a similar definition:

1. the text or speech that comes immediately before and after a particular phrase or piece of text and helps to explain its meaning, and

2. the situation within which something exists or happens, and that can help explain it.

The term "context" is quite frequent in the information retrieval literature and was defined in various ways, although there isn't a generally accepted definition of it [41]. Contextual retrieval is considered to be a long term challenge in IR and a definition is found in [9]: "Combining search technologies and knowledge about query and user context into a single framework in order to provide the most 'appropriate' answer for a user's information needs.". In the Inquirus 2 project [31], the context is represented by the category of the information the user is searching like: "research paper", "personal homepage" or "general information". This contextual information is provided by the user and it is used for query modification, to choose the search engines to send queries to, and for the ordering policy that will be used. The Watson project [17] and the IntelliZap system [28] consider as contextual information the text surrounding a user marked query term from a document; [19] extracts contextual information from open Word documents and Web pages that the user is manipulating at a particular time and use it to create user profiles, [69] uses the recent queries as a basis for the user's context. Another definition of context can be found in [48] where it is defined as a combination of titles and descriptions of clicked search results after an initial query. In [62, 63] the authors propose the use of clicked summary text and previous queries text as source of contextual information. Also the search context in SearchPad [15] is represented by the user recently queries terms and the links of result pages he considered as relevant; [42] discuss the possibility of using physical information as a source of context in information access.

In our work, the context is represented by the user's set of documents that are related with his current interests. In most cases when we are editing or just reading documents is when questions about the information we are seeing surge and make us issue a query based on a word/phrase that we have seen from that particular document. Hence, the activity in which we are engaged just before/while sending a query could be processed in order to extract contextual information. Let's take the following example: a user who just received an announcement about a vacant position in his institution and is reading it, probably wants to find for more possibilities of employment and enters the query "positions". On the other hand, a user that is reading about Yoga exercises, and enters the query "positions", more likely wants to learn new yoga positions. Therefore, it is desirable to develop new ways to understand the user interests at a particular time that could be used in the

retrieval process in order to improve the results of the search.

1.2 How context can be used

As we can see from these studies, "context" means different things in different systems. It can also be used in two majors ways: by expanding the query or by post-processing the results based on the contextual information gathered. The approach presented here belongs to the first class: it focuses on expanding the query in order to improve the precision of the results. Inquirus 2 [31] is an example of a meta search engine that uses the reordering of the results differently for different users based on a information need category provided by the user. For each category there is an associated scoring function. When search engines return the results, the scoring function is computed by downloading and analyzing the text in the Web page. The contextual information is used at different levels of the retrieving process: from the selection of the search engines to send the query to, to the ordering policy used. In [20] user's profiles are constructed based on topic categories identified by the user. Each category has assigned a score that represents the relevance to the user preference. The final result ranking is based on a combination between this score and the distance between topics and search engine results. Also, [30, 65] build user's profiles by selecting categories that will help in result reordering. Although this approach has the advantages that the scoring function is computed only for the first documents and the accuracy of relevance ranking can be improved, a clear disadvantage is found in the limitation of organizing the documents returned by the non-contextual query, which often have low precision in the case of ambiguous query. In other words, if the search engines have low precision, the reordered results will have low precision also. This could be resolved by retrieving a larger number of results but can be very expensive.

The second solution is to use context for query rewriting. Contrary to results reordering, query rewriting involves the modification of the query. The query represent the user's information need by adding contextual information so as to retrieve documents not only relevant to the user query but also to the contextual information. For instance, the user may type the query "check", but this is ambiguous because we do not know if the user is interested in finding tricks for his next chess game or if he just wants to know if he can

cash a check at any bank. Query modification could add the terms "chess" or "bank" depending on the user's contextual information. In this example, the ambiguity was due to homonymy, that is, to a word with multiple meanings. A similarly ambiguous phenomenon is polysemy, the difference between the two of them being subtle and consisting in the etymology of the word. These two phenomena are affecting the precision of the retrieval results. We try to resolve these kinds of ambiguities introduced by these phenomena. The contextual information that is used to be appended to the initial user query was seen in different ways. In [28] the information surrounding the query words in the document that the user was reading prior the search was used as additional terms that provide contextual information. In [17] the contextual information is gathered from the documents the users are manipulating (as word documents, Web pages etc.) and used as an additional information in the retrieval process. Other studies [50] had found that six additional terms are sufficient for improving of the results in the case of an automatic query expansion. A significant advantage of using this approach is that it help in improving the precision of the results. Its main drawback is that through terms expansion the query becomes more complicated and therefore this result in a lower recall, as all the relevant documents contains all the query words.

1.3 Context and semantics

The context issue received a lot of attention in many researches in *semantic web* [14]. A possible solution for the "semantic problem", that is receiving nowadays a lot of attention, consists of formalizing the semantics of the information that will be published in the web, expressing it in a formal and not ambiguous language. This is the line of work chosen for so-called semantic web. Nevertheless, this line of work presents many limitations and many problems, of a conceptual and pragmatic type. From the conceptual point of view, the formalization of the information does not consider that the meaning is not a property of a document, but the result of a dialectical process of interpretation, a process that always forms part of a given activity and that, therefore, develops in a context that depends on this activity [59]. From the pragmatic point of view, the success of the formalization depends on the will of the person who publishes the information to provide a complete

and precise annotation *(meta-data)*. The sociology and the economic reality of the web had made that several authors doubted that these perspectives are realistic [25].

Our work is framed in a somewhat dual perspective: recognizing the priority of the context in the interpretation of documents, our effort is not directed towards formal representation of documents, but towards the representation (formal or less) of the context and the activity in which the search is made. In this field, we follow Sperber and Wilson [64, 55] turning the traditional relationship between context and meaning. The common way of seeing this relation is to consider the effect of the context on the meaning: to interpret a document in a context C supposes giving it a different meaning that if it was interpreting the document in a context C'. Sperber and Wilson, and us with them, consider the opposite relation: the meaning of a document is determined by *the change that the document causes in the context of interpretation*.

We are, for example, in a context C and, in this context, we are interpreting a document d. The presence of the meaning of d supposes a change in our context, from C to C' (if the document does not change anything in our situation, reading it or not would be all the same, that is to say, the document would not make any sense for us). We postulate that the meaning of the document is exactly equal to $\Delta(C, C')$. Some observations about the idea of meaning are in order.

i) This definition provides us a very natural model for the historicity of the context: the context is not fixed, it changes with every act of interpretation. It is a question, as we have seen, of a dialectical process: the process of interpretation determines the meaning of the document and, on the other hand, the fact of interpreting a document changes the context [26].

ii) The literature on formal annotation has been interested occasionally in context using techniques that we might call "contextual annotation". Our way to see the context is very different. In the field of annotations, the context is often a component of the description of a document, namely the interpretation context is also a property of the document: the document admits several interpretive contexts and, in each of them,

its meaning is determined by the corresponding annotation [35]. Conceptually similar it is the case of *emergent ontologies* (also called *folksonomies*). In this case, the user community contributes criteria and interpretations that form the annotation of the document [34]. In these cases, the vision of the context is different from ours. In the contextual annotation several contributions join to create a context that is always a context of a *document*. Each search is done by reference to the context of the document. In our case, each search involves a different context. There is no context associated with the document, but so many different contexts as the acts of search.

1.4 State of art

Various intents have been made in the sense of determining the context of a search of a user, that could be divided into the following categories [17, 47]:

1. Feedback is a technique that could be seen as a method of involving the context in the search because tries to represent the user interests or lack of it. First the user issues a simple query and based on the content of the documents that the user evaluates to be relevant, the initial query is modified by adding words or adjusting the weights of its words. This new query is used to retrieve a new set of documents. The principal drawbacks of it are the requiring of explicit user involvement and the extra time needed. Furthermore statistics showed that users were unhappy with the results and are unwilling to use it [67, 66].

2. Word sense disambiguation tries to reduce the ambiguity of a user information need by asking explicitly the user to provide context information or by implicitly inferring context information. In the case of Inquirus 2[31, 32] the context is explicitly requiring the user to choose the category of the information he needs and based on this decides either to select the search engines where to send the query, modify the queries or select the way the results are ordered. Some disadvantages could be that this work only within given categories and that implies the user involvement. On the other hand, the systems that infer automatically context information has the advantage that does not necessitate the involvement of the user. Watson project [17] is an example of such a

system that uses the text of the Word document or Web page that the user is editing or viewing to automatically guessing the context information that will be added to the search query. Remembrance Agent [54], while a user is manipulating a document in the Emacs editor and based on this text and email messages and research papers of the user, makes suggestions about possible relevant documents for the current user situation. This is a program that searches continually without any user intervention.

3. The representation of the users interests by using user profiles could be seen as another way of collecting user contextual information. This profile could be based on the information given by the user and documents that were previoussly selected as relevant. A posible drawback is that this represents long-term intereses, therefore are not capable of proving accurate information about the current user context. Nevertheless, in [70] an user profile is created using contextual information from Word documents and Web pages that user is working with at the time he issues the query. In this way the authors tries to capture the most recent interests of the user.

4. Knowledge engineering approaches use a model of the user behavior in order to infer the user's interest in information. The Lumiere project [36] uses the user's background, actions and queries in order to infer the user's information needs. The EPOS project [61] gathers the user's context by observation plugins for applications as Mozilla Firefox and Thunderbird and use it in order to support aid in the user's searching, reading, creating and archiving documents. Other researches that use as a measure of the user interest the information captured from the user interaction can be found in [22, 48, 63, 71]. This could be an inaccurate indicator of the user interests due to facts that it is estimated from behaviors depending on diferent individual factors [49].

5. Domain specific search engines represent another way of considering the context by restricting the information that is indexed to a certain domain. With so many domain specific search engines a user could find it very difficult to locate such a service and even more to decide which is the best.

6. Predict the information that a user is searching for is a technique limited to very few and popular user keywords. For example, if a user sends to Google [3] a query that contains a U.S. street name, the returned result will provide a direct link to the map of it.

2 The context

The purpose of this work is to give a precise formulation that can be applied in real circumstances to the idea of *context-determined semantics* that has been explained briefly in the introduction. Our initial hypothesis is that a search is being conducted as part of an activity that takes place, at least in part, on the personal computer of the user. The computer contains documents of different types: text, spreadsheet, images, etc., that the user has organized in a way that reflects the important associations to the activity that is developing; these documents, and their organization contain important information about the context of the search, information that we will use for disambiguation and focusing. The information on the context, in other words, come from two different sources: on one hand the analysis of the contents of documents found in the user's computer (or, as we shall see, some of them), and on the other hand the analysis of the structure of the folders in which the user has organized these documents. We always consider the search as part of an activity, as can be implemented, for example, putting the search programs as plug-in in applications such as word processors and electronic sheets. In other words, when the user searches something he is always working on a document located in a certain folder. The context of the computer, relativized in reference to this folder, will be used to modify and focus the search and filter the results.

As an example, we can mention here the contribution of context in reducing ambiguities due to polysemy. For example, the word "bank" can mean *financial institution*, or *shore of a river/lake*, depending on the context in which appears. If at any given time a user works with documents related to financial data, and makes a query about "bank", she probably refers to the bank as financial institution, and with this meaning of the word the results should be recovered. The ambiguity of the meaning (generated in the example seen by the presence of the polysemic word "bank") is an essential characteristic of the natural language at the point that many modern theories

of interpretation deny that a word could be assigned only one and constant meaning and, consequently, that a text has no fixed meaning [29, 11]; its meaning is not determined only by its contents, but also by the activity in which interpretation is placed. For example, if the user is a biologist, is very likely that in his computer the documents are organized so that a certain folder contains working papers, in which case the word "bank" probably refers to the bank of a river, and perhaps another folder contains information on the bank accounts of the biologist, in which case, the meaning of the word "bank" will be the bank as a financial institution. This example shows that we can't assume that there is a one-to-one correspondence between users and contexts: a user is not characterized by a unique context, but by a multitude of contexts, one for each one of his activities.

2.1 Techniques

The context will be represented using information retrieval techniques based on the text of the documents directly related to the activity that is being developed (*primary context*), and of documents indirectly related to these and that can contribute to the formation of the context (*secondary context*) [58]. For example, if a person makes an access to data while creating a document, the primary context is made up of all the documents that are in the same folder as the document that's being edited, and the secondary context of all the documents that could be found in the folders descendants of the working folder or, according to the situation, in the "sisters" folders. A typical situation, where someone is preparing a presentation for a conference, is presented in figure 1. In this case, which will be our reference case in this work, the context of the search is built through two distinct operations:

i) in each folder we build a representation of the local context of that folder, based on the documents therein present; we call this representation the *context generator* of the folder;

ii) the generators of the primary context folder and of all secondary context folders join to create a complete description of the context of work; this representation will be called the *index* of the context, and will be stored in the folder of the primary context.

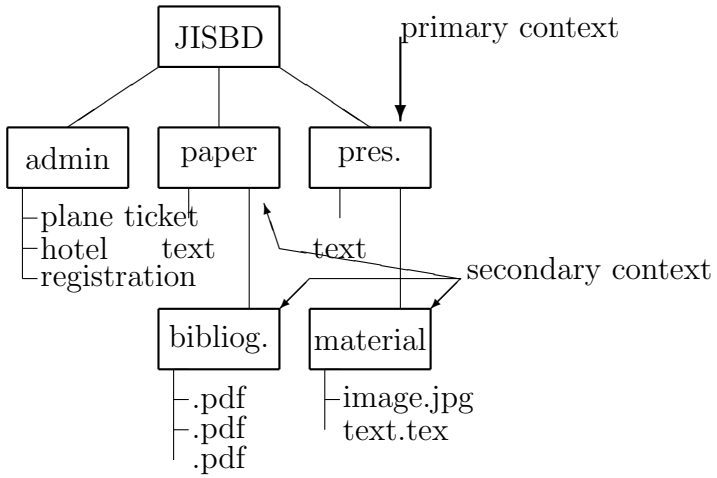

Figure 1: The structure of folders and context for the preparation of a presentation.

In the above example, in each of the six folders a generator will be created, with an appropriate representation of the context of that folder (that is to say, a representation of the documents that appear in the folder). The generators of the *pres* folder (the primary context) and of the folders *paper*, *bibliogr* and *material* (the secondary context), will join using appropriate operators, to form the index of the context of the search, index that is stored in the folder *pres*. In reference to this index, we have to make two comments:

i) the folder of the primary context *(pres)* contains, at the end of the operation, two kinds of representations of the context: a generator of the context, that represents only the context of the folder, and an index, which represents the context of the folder and the secondary contexts folders; only the index will be used in the search operations: the generator of the folder is simply an instrument through which the index is built;

ii) the construction of the index through generators supposes a hypotesis of compositionality of the representation of context: the representation of the global context of two or more folders depend only on the representation of local contexts and the relation between folders; we will see that this condition is checked for the representation used in this work.

2.2 The context of a folder

Consider, for the time being, the representation of the context of a single folder. This context depends only on the documents that are found in it. As we have already mentioned, in this work the representation of the context is carried out using techniques of information retrieval, in particular, using a technique similar to that of the semantic map WEBSOM [44]. This semantic map presents two features that are essential in our case: the representation of context by means of *self-organizing maps* in the Euclidean space of words, and the use of *word contexts* as a working and learning unit of the map.

The self-organizing map forms a certain kind of non-linear *latent semantic* space, and this non-linearity will result very useful when it comes to making changes in context (e.g. to express a query, as we shall see shortly). Such maps can also be useful because over time the context is modified by adding and/or removing documents; the posibilities of the maps to adapt (*learning*) allows us to follow these changes. We note that, on the other hand, the use of linear techniques, such as *Latent Semantics Indexing* [24] produces a series of concepts that relate indirectly terms with documents using *singular value decomposition*. From our point of view, the method has the disadvantage of not allowing local changes of the context, because each change to a linear transformation is global.

Many representations of documents (including the standard version of WEBSOM) starts from the frequencies of words of the document; this representation is insufficient for our problem because if we use only a word by itself, the semantics that derives from the co-location of the words will be lost, and we need co-location to solve problems like the polysemy. On the other hand, in the technique that we will use, the fundamental unit of representation that is extracted from the document is not the word, but a group of words, that is called word context. The number of words of the word context may vary. In this work we consider the simplest case: two words, i.e., we will consider pairs. Each pair of consecutive words in the text is seen as a symbol to which we assign a weight proportional to the number of times the symbol (in other words, the pair of words) appears in the text (fig. 2, left).

These pairs are represented in the typical geometric space of many information retrieval systems, a space in which each word is an axis. Since our basis are the contexts, the points from which the representation is derived

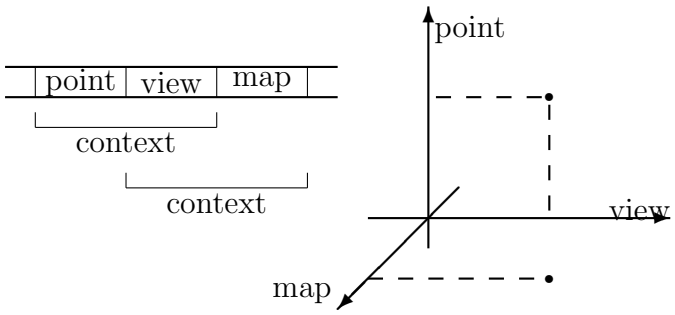

Figure 2: The geometry of the words context.

are not points in one of the axes (as in the case of simple words: each point is a word with its weight), but points in bi-dimensional sub-spaces: each pair is a point in the plane represented by the two words that compose it. Using more complex contexts will result in points contained in spaces of higher dimension.

The preliminary processing steps for the applying of this algorithm are the typical ones of the information retrieval systems: it starts with the full text of the document and the stop-words (words so common that they have no information value) are eliminated (words like *the, of, from, to, ...*). Following this removal, we perform *stemming*. In other words, the suffixies that change the words are removed and we retain only the root (e.g. we eliminate the -ed that marks the last of the verbs, the -s of the plural, irregular expressions like *went* are transformed into their roots like *go*, etc.). The most widely used and effective algorithm for English stemming is Porter's algorithm [53], which consists of a set of rules that are applied in five distinct phases of word reductions for each word in the text. Within each phase in order to select a certain rule some conditions have to be met. All the rules in the phase are tested until either a rule from that phase fires or there are no more rules in that phase to be tested, an the algorithm goes to the following phase. At the end of this five phases that are applied sequentially the process stops and returns the stem. For example, we will obtain the same stem *stem* in the case of the words *stemmer*, *stemming* and *stemmed*. For the words *stemmed* and *stemming* the rules 2 and 3 from the step 1 [53] are true, therefore the suffix *ed* and *ing* will be removed:

stemmed -> stemm

stemming -> stemm

As the second and third rules were true and the words are ending in a double consonant, the following is done:

stemm -> stem

stemm -> stem

None of the rules from the following steps are true, so the stem in both cases will be *stem*.

For the word *stemmer* others rules are applied, the first one that is true is the forth one, therefore the suffix *er* will be removed

stemmer -> stemm

Afterwords, step 5b is true and the final *mm* becomes *m*:

stemm -> stem

Therefore, these operations provide a list of stems from which we extract the word context on which the map operates.

2.3 Self-organizing maps

The self-organizing map (SOM) [46] is one of the most prominent neural network algorithms and was introduced by Teuvo Kohonen. The SOM, based on the unsupervised learning paradigm, constructs from a high-dimensional space a map in a lower dimensional space that preserves the topological relations of the input space. The map consists of a number of neural processing elements called neurons. To each of the neuron will be associated a weight vector that must have the same dimensionality as the input data samples. The neurons usually form a two−dimensional map that can be displayed in a hexagonal or rectangular grid and each neuron contains semantic information. The main idea in order to map a vector from data space onto the map is to find the neuron that is closest in terms of distance with the data vector and to modify the weights of that neuron and his neighbours in order to obtain a better representant of that input vector. In this way different parts of the network will correspond to certain input patterns.

```
Initialize the map with neurons in random positions
for t = 1 to m
    get an input data sample
    find the BMU
    scale neighbors
    increase t
end for
```

Figure 3: The self-organizing map algorithm.

2.3.1 Self-organizing maps main algorithm

The number of neurons is chosen from the start, cannot be modified and influences the accuracy and generalization of the map.

The map organizes itself by making use of competitive learning where each neuron is competing for the representation of an input sample. During the presentation of an input data sample to the network, the neuron that best represent it, that is, the neuron closest to the sample is chosen using a distance measure, in our case the Euclidean distance. The neuron that wins the competition is named the best matching unit (BMU) and is the neuron that is most similar to the data sample (the Euclidean distance between the data vector and the neuron is the minimal). After the BMU has been found, the weights of the winning neuron are adjusted moving it closer to the input vector in the input space in order help it win the next competition. The amount that is used to adjust its position is called *learning rate* α. Also, all the neighbors within the *neighborhood radius* σ of the BMU are moved closer to the input vector. The initial neighborhood radius is chosen in function of the size of the map and is gradually reduced during the learning until it reaches one. This means that over time the neurons get closer to be a better representant of a certain input patterns, so they will only need a slight modification. As well, only the neurons that are closer to the BMU require a bigger similarity with that pattern.

The basic steps of the algorithm are presented in figure 3. The first step in creating a SOM is to initialize the weights vectors of each neuron with random values. Afterwards for a large number of cycles the algorithm chooses

randomly an input data sample and for each node of the map computes the distance and tracks the node that generate the smallest distance (the BMU). There are different ways of calculating the distance between two vectors, nevertheless the most common is to use the Euclidean distance from the formula 1:

$$distance(d_i, n_i) = \sqrt{\sum_{j=1}^{n} (d_i^j - n_i^j)^2} \tag{1}$$

where $d_i = (d_i^0, d_i^1, ..., d_i^n)$ and $n_i = (n_i^0, n_i^1, ..., n_i^n)$ represent the data input and neuron vectors.

The BMU is the neuron that best represent the input sample and at any given moment there can be just one BMU. In order to scale the neighborhoods there are two things that have to be done: determining which neurons are considered to form part of the neighborhood of the BMU and the amount used to make the neurons of the neighborhood become more like the input sample. The neighborhood function in our case is the Gaussian that is defined in section 2.5.

Geometrically speaking, the neurons from the neighborhood of the BMU are moved a bit towards the input data pattern. The initial radius is set up high and decreases over time. Also, the amount a weight can learn also decreases with time. The neurons that are farther of the BMU, will learn less than the ones closer to it.

Therefore, all the neighborhoods of the BMU based on a certain value of learning rate, are rewarded by being brought more closer to the input data. These steps define a single training step and they are repeated a large number of time until the training ends. After the training is over, the output constitutes a representation of a semantic map which associates output neurons with patterns in the input data space.

The SOM algorithm is used in a wide rage of practical applications and the most important advantage of using SOM is the ability to produce an approximation of the topology of the input samples in a lower space. The map uses knowledge about previous winners in order to help finding new ones. The topology makes that a certain part of the output responds for the same inputs. Nevertheless, one drawback with SOMs is that are very computationally expensive since the dimensions of the input samples are big.

2.4 Calculating pair weights

The most common way to measure the importance of a word to a document in a collection in information retrieval is the use of *tf-idf* weighting scheme. The main idea is that a word is more important if it appears many times in a document, but its importance decreases as it is contained in many documents. In our work, we are interested to know the importance of a certain word in the entire collection. Therefore we will use the term frequency for evaluating how important is a pair in that collection, and is defined as the number of times that term appears in the entire collection. In other words, the *weight of a pair* is represented by the frequency of that term in the collection normalized by the number of terms that the collection contains:

$$\omega = \frac{number\,of\,times\,the\,term\,appears\,in\,the\,collection}{number\,of\,terms\,of\,the\,collection} \tag{2}$$

We define the weighting vector ω^i as:

$$\omega^i = (0, \ldots, \omega, 0, \ldots, 0, \ldots, 0) \tag{3}$$

Where ω is calculated the for each pair for the generator. This weights are used at the time of computing the weights for the index, as we shall see in the following section.

2.5 The index of the context

The *index* is a union of the generators of the primary and secondary contexts. In the case of our reference activity, the secondary context is composed of the folders related to the work folder; a problem that the system has to resolve is *which* of these folders should be used. We consider three options: the secondary context,

i) is identical to the primary context;

ii) contains the primary context and the local contexts of all their descendants;

iii) includes the primary context and all related folders (descendants, siblings and the parent).

We call *working context* to the assembly of folders that are used to calculate the index. In order to calculate the weights of the words that compose the index, one must take into account the relation between the folders in question. The folders are organized according to a logical structure and are grouped by themes, so it is very likely that a word in a descendant folder have a meaning very related to the same word as found in its ancestor folder. In our previous example, for the word "bank", it is probable that its meaning in the *material* folder be different from that of the *admin* folder. In the first case, it might refer to the banks of a river (remember that the user is a biologist who wants to send an article about his research), and the second, the data bank accounts (which uses the same user at the time of registration in the congress or to pay the airplane tickets). For implementation reasons, we divide the secondary context in two:

- the secondary context corresponding to the descendants of the primary context folder;
- the secondary context for the sibling and ancestor folders.

More in detail, we will consider the three options above presented, the weight of the pair constitute by the word number i and word number j (in other words, the word pair who has values in the e_i ans e_j axes of the space of words) and that appears in several folders of the work context, determined as follows:

i) the work context is identical to the primary context: as there is only one context folder, the relation between folders does not affect the weights of the words in the work context, that is we will have:

$$\omega^{ij} = \omega^{ij}_{CP}, \tag{4}$$

where CP is the primary context, ω^{ij} is the weight of the pair i, j in the index and ω^{ij}_{CP} is the weight of the pair in the primary context generator;

ii) the working context contains the primary context and the local contexts of all its descendants; in this case the weight of the pair i, j in the index is given by:

$$\omega^{ij} = \omega^{ij}_{CP} + \gamma \sum_{CS} \omega^{ij}_{CS}, \tag{5}$$

where *CS* is the secondary context corresponding to the descendants of the primary context folder (the folder *material* in our example), ω_{CS}^{ij} is the weight of the pair in the generators of the secondary contexts and γ is a constant, $0 \leq \gamma \leq 1$;

iii) the working context coincides with the global context (in other words, the primary context and all the secondary ones); in this case the weight of the pair i, j in the index is given by:

$$\omega^{ij} = \omega_{CP}^{ij} + \gamma \sum_{CS} \omega_{CS}^{ij} + \delta \sum_{CSS} \omega_{CSS}^{ij}, \tag{6}$$

where *CSS* is the secondary context for the sibling and ancestor folders, ω_{CSS}^{ij} is the weight of the pair in the corresponding generators and δ is a constant, $0 \leq \delta \leq 1$.

The pairs of words in the index, with the weights thus determined, are used as inputs for training the self-organizing map. The map consists of a matrix of $N \times M$ neurons, each neuron being a vector in the input space. In our case, that space is the space of words, so if the context is composed of T words, the neuron μ, ν ($1 \leq \mu \leq N$, $1 \leq \nu \leq M$) is a vector

$$[\mu\nu] = (u_{\mu\nu}^1, \ldots, u_{\mu\nu}^T) \in R^T \tag{7}$$

The map learning is trained under the stimulus of a set of points in the input space, each point representing a pair of words. Given a total number of P pairs of words, and given that the pair number k consists of the words number i and j, the corresponding point in the input space is given by

$$p_k = (\overbrace{0, \ldots, \underbrace{\omega^{ij}}_{j}}^{i}, 0, \ldots, \omega^{ij}, 0, \ldots, 0) \tag{8}$$

where ω^{ij} is the weight of the pair of words determined by the equations (4-6). During the learning the p_k vectors are presented several times to the map. We call *event* the presentation of a vector p_k, and *iteration* the presentation of all vectors. Learning consists of several iterations. An event in which the vector p_k is presented entails the following operations:

i) Identify the "winning" neuron, in other words the neuron that is closest to the vector p_k:

$$[*] = \min_{[\mu\nu]} \sum_{j=1}^{T} (p_k^j - u_{\mu\nu}^j)^2 \tag{9}$$

ii) The winning neuron, $[*]$, and a certain number of neurons in its neighborhood are moved toward the p_k point by an amount that depends on the distance between the neuron and the winner and the number of iterations that have been performed so far. To this end, we define the *distance* between the neurons of the map as:

$$\|[\mu\nu] - [\mu'\nu']\| = |\mu - \mu'| + |\nu + \nu'|. \tag{10}$$

For $t = 0, 1, \ldots$ the counter of the iterations of the learning, we define *neighborhood function* $h(t, n)$ such that

$$\begin{array}{ll} \forall t, n \geq 0 & 0 \leq h(t,n) \leq 1, h(t,0) = 1 \\ & h(t,n) \geq h(t,n+1) \\ & h(t,n) \geq h(t+1,n) \end{array} \tag{11}$$

and a learning coefficient $\alpha(t)$ such that

$$\forall t \geq 0, 0 \leq \alpha(t) \leq 1, \alpha(t) \geq \alpha(t-1) \tag{12}$$

Then each neuron $[\mu\nu]$ of the map moves toward the point p_k according to the learning equation

$$[\mu\nu] \leftarrow [\mu\nu] + \alpha(t) h(t, \|[*] - [\mu\nu]\|)(p_k - [\mu\nu]) \tag{13}$$

The function h generically corresponds to a neighborhood of the winning neuron that becomes smaller as the number of iterations increases. In this work the environment function is the Gaussian $h(t, n) = \exp(-n^2/\sigma(t)^2)$, with $\sigma(t) \geq \sigma(t+1) > 0$.

At the end of the learning process the map is deployed in the words space in a way that, in the extreme case of an infinite number of neurons that form a continuum, it optimally approximates the distribution of the points in the space [57]. This map represents the semantic space of the context and, as we mentioned in the previous section, can be assimilated to a nonlinear form of latent semantics.

2.6 The query

In its most complete and general form, the procedure of a query is composed of four phases:

i) through an appropriate user interface or through a program that the user is using, an initial specification of the query is collected. We will call it the *pre-query*. The pre-query can be formed by a few words typed by the user, a paragraph that the user is editing, etc.. In a multimedia system (system that we do not consider in this work) the pre-query also contain an indication of the type of the document that's being searched (text, image, video, etc.)..

ii) The pre-query is used to change the current context, transforming it into a *target context*. In practice, the configuration of the map (index) of the current folder is modified through a partial learning, which will give the context a *bias* towards the pre-query. The resulting configuration from this learning could be considered, in some way, as the interpretation of the pre-query in the actual context.

iii) The difference between the actual and target contexts is the *differential context* and, in our model of semantics, corresponds to the semantic of the ideal document that is searched for: the document that, if assimilated to the current context, will transform it into the target context. An opportune codification of the ideal document is created and sent to the search server to retrieve the documents that more respond to that profile.

iv) The documents elected (e.g. read or downloaded) become part of the context: a new learning is executed so that the current context reflect the new situation.

This general model of a query assumes the existence of a search service *(search engine)* capable of managing contexts. The construction of such a service is one of future goals of our work. For the moment, our objective is to demonstrate the role played by the context using it to focus searches on existing services. Therefore, it is necessary to transform the differential context into a list of words with weights, because the search services only

accepts (if it accepts) this type of queries. Obviously this type of query can't make an optimal use of the possibilities of context but, we repeat it, at this moment our goal is simply to evaluate the influence of the use of the context in the search. In our tests, the pre-query is formed by one or more keywords. A keyword that correspond to the i-th word of the space is represented as the vector $e_i = (\overbrace{0, \ldots, 1}^{i}, 0, \ldots, 0)$. For simplicity we assume that every word in the query has the same weight w. Therefore, the query Q, formed by q words, will be represented as a point in the T-dimensional space: $Q = w \sum_{i=1}^{q} e_i$

This vector is used for a partial learning process using the algorithm presented in the previous section. During this process the neuron $[\mu\nu]$ is moved to the position $[\mu'\nu']$. The differential context is given by the differences of the neurons positions, $\delta_{\mu\nu} = [\mu'\nu'] - [\mu\nu]$ for each $[\mu\nu]$ in a neighbor of the winning neuron, that is, of the neuron closest to the query vector Q.

Projecting the vector $\delta_{\mu\nu}$ on the word axes, we get the weights of the words given by this neuron: $\delta_{\mu\nu} = (v^1_{\mu\nu}, \ldots, v^T_{\mu\nu})$. The *non-normalized weight* of the word i is given by the sum of their weights relative to all the neurons in a neighborhood A of the winning neuron

$$V^i = \sum_{[\mu\nu] \in A} v^i_{\mu\nu} \tag{14}$$

Considering only the K words with greater weights, and normalizing the vector of weights for these words we obtain the query that will be send to the search engine, composed of a set of words each one associated with a weight.

2.7 Changing the context

During a search, the user can add some new documents to the current context, documents retrieved and relevant according to the user. Those documents will become part of the current context by downloading them in a temporary folder in the working directory. Those documents will be part of the new representation of the context by running a new learning that includes them. Changing the context by adding new relevant documents helps in obtaining a better representation of the context and can help in improving the precision of the results.

3 Experiments and results

3.1 Evaluation in Information Retrieval

Evaluation in Information Retrieval is an important and well studied task [60] that tries to measure the effectiveness of the retrieval results in order to identify the drawbacks and look for better methods that will improve the results. It is known that that the evaluation is difficult, time-consuming and expensive to perform [72, 73]. The most common and frequently used retrieval evaluation measures are *recall* and *precision* [8]. Recall and precision are defined as [10]:

$$Precision = \frac{number\,of\,retrieved\,relevant\,documents}{number\,of\,retrieved\,documents} \tag{15}$$

$$Recall = \frac{number\,of\,retrieved\,relevant\,documents}{total\,number\,of\,relevant\,documents\,in\,the\,collection} \tag{16}$$

Precision measures how well the system works in not retrieving documents that are not relevant, while recall measures how well the system finds relevant documents.

These measures revolve around the notion of the relevance of a document, first used in [45]. A relevant document is usually defined as a document that the user decides that suites his information needs. We must underscore that relevance is not evaluated relative to a query but to the user information need. A user will never consider a document as relevant just because it contains the words of the query. For example, in the case of the query "coffee limit drink health", the user may actually want to get information on how many cups of coffee one can drink without involving any health risk, and will not judge a document as relevant just because it happens to contain all the query words. Other measures of effectiveness used in order to test different approaches in information retrieval have been proposed:

- Precision@n (P@n) is defined as the precision at the point when we have only n results and it is used in order to evaluate how many relevant documents are included in the retrieved results. If its value is high it means that the system can search can retrieve relevant documents effectively.

- Average precision (AP) is the average of the precision of all relevant retrieved documents and it is used to evaluate the precision and the coverage. This measure combines the precision, the ranking relevance and the recall. Geometrically speaking, the average precision corresponds to the area under the precision-recall graph.

- R-Precision represents the precision at the point when R relevant documents have been retrieved. If the R-Precision is 1 means that the ranking relevance is perfect.

- Reciprocal Rank (RR) is defined as the reciprocal of the first retrieved relevant documents.

Usually to evaluate a system one calculates the mean average precision (MAP) that is defined as the average of the all APs of all queries.

3.2 Our experimental Study

3.2.1 Context of the search

Relevance can be an ambiguous concept [16, 33, 52]; in our case a document was considered relevant if the information it contains is from the context area.

The goal of this work was to find out whether our way to see and represent context can improve Internet searching by expanding user queries based on the contextual information gathered from the documents contained in his current working space.

We have decided to base our experiments on the search engine $Google^{TM}$[3], which, according to Nielsen Online [7], is the leader in Internet search. Our testbed, called *Coeus v1.1* has been designed as a client application that is running on the user's system and includes an interface that allows users to specify queries and to view the results, the main window of the system is shown in figure 4, where the *algoritm* query had been issued. Some observations have to be made. In order to do a context search, the user has to choose a working context and provide simple keyword query that describe his information need. The working context can be provided by the user through the user interface by choosing a working directory (it is supposed that this directory contains the documents that the user is working with at that moment or

are related with his search context). In order to conduct a context search it is assumed that a prior learning had already been done through by executing a program. An example of the use of the contextual search interface will be described in the following. When a user needs to find information regarding a certain topic, he firstly has to choose a working folder from the application menu. Once the working context is chosen he has to type the keywords in the text field in the main window of the application and selects "Search" to issue the search, as normally done on the Internet. The user keywords are first processed using a common English stop words list and stemmed using Porter's algorithm. After this, in order to extract the additional words that describe the current context, a partial learning is done on the working context, as explained in the previous section. The words that better define the working context (as shown in section 2.6) are appended to the user query together with their context weights. These weights should be passed along with the query terms to the search engine. However, in this testbed, this is not possible: Google does not permit querying using weights for each keyword. Nevertheless there is a possibility of attributing more "weight" to a keyword versus the others, that is ranking in order of importance the keywords. This is done by repeating the words that are more important. When the user initiates a search by providing a query, there are actually two queries that are sent to Google: one is the user query "as is" and the other that includes the user's query plus the contextual information generated by a client module as described in the previous sections. The client application then retrieves the results of both queries and displays them to the user: the first half of the window contains the response of the non-contextual query (the one that the user typed) and the other one includes the user's contextual information. This offers us a good way to see the results in both cases and compare them. The list of results is composed of the first eight Web pages with the titles and descriptions returned by the search engine in each case. We limit the search results to the first eight documents (corresponding to the first result page), as some researches [39] has found that the number of the result pages the user is viewing in very few cases goes beyond the first two ones. The user can see all the words that are contained in the contextual query by selecting in the interface menu the corresponding option. The interface also offers the facility to choose between the search results the ones that the user consider to be relevant in his working context in order to add them to it and use them

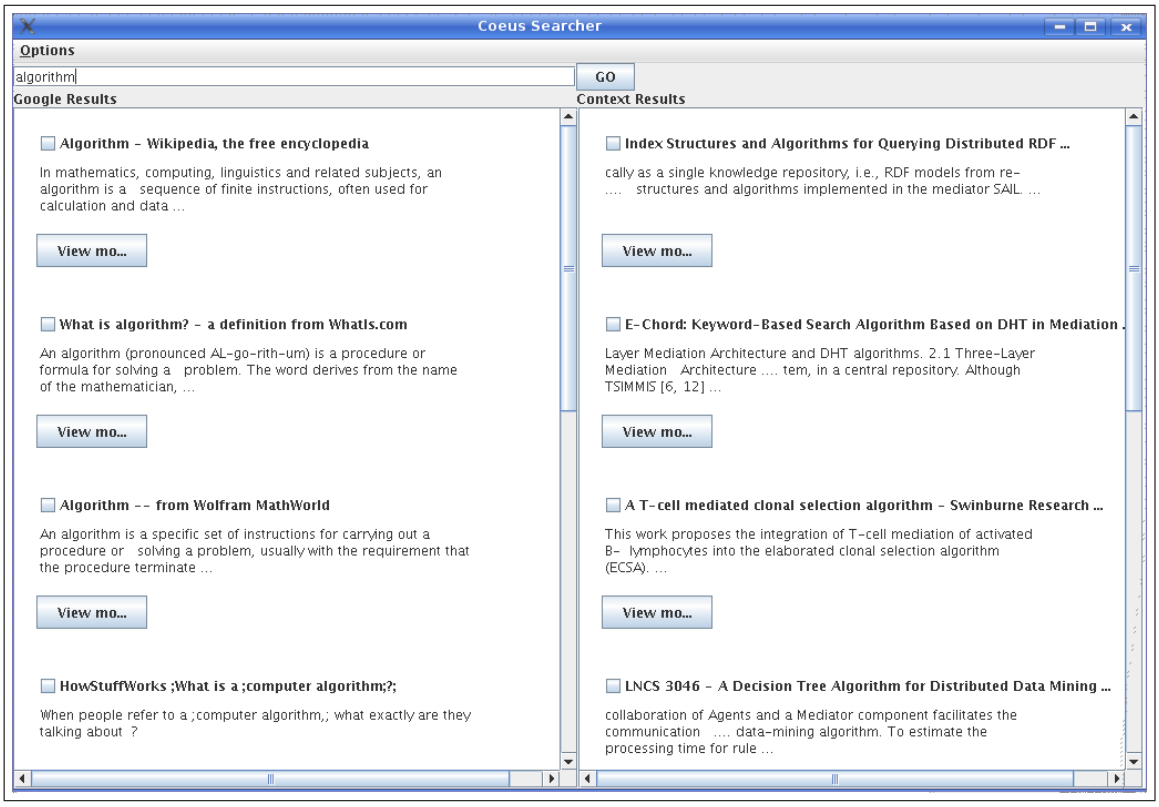

Figure 4: Main window of the Coeus Searcher.

in for updating the representation of the context. The Web pages selected by clicking on the check box in front of the result title are downloaded and parsed with a HTML parser in order to extract the text of the page that will be saved in a temporary folder that will be created in the working folder. In order to use those new documents in the search, a new learning has to be employed.

3.2.2 Textual search results

We conducted the searches for three distinct working contexts, each one from distinct areas: computer science, philosophy and neurophysiology. The computer science documents were taken from a real context of a computer science professor. For philosophy and neurophysiology we downloaded from the Internet some documents specific to each area that we have considered to be relevant. For each area, we conducted searches for 30 different queries with words from the context. For each query we measure the precision for the first eight results. As in real scenarios is not possible to know the number of relevant documents in the collection, recall is impossible to be calculated. The results are shown in the figures 5, 6, 7.

In the following, we will consider some examples of the most representative queries which leads us the attention: the word *relationship* in the context of neuropsychology, the word *experience* from philosophy and *executed* for computing. In the case of the query word *relationship* without taking into account the context, (in other words, the query contains only the word *relationship*), in the majority of the results returned the word of the query has the meaning of couple relationship. By contrast, in the case of query *relationship* to which the contextual query was *relationships relationships relationships relationships neuron contends inside* it can be seen that the results in most of the cases are related to aspects from neuropsychology field. In the case of *experience*, most of the results of the non-contextual query were referring at work experience, but taking into account that the context of the search is philosophy, it is more probable that the user meant experience of life. The contextual query *experience experience experience experience originated originated thereof universe* the results were more relevant for the current context. The word *executed* is also a good example of seeing how context made the difference as the documents retrieved without involving the context are in most of the case referring at execution as death

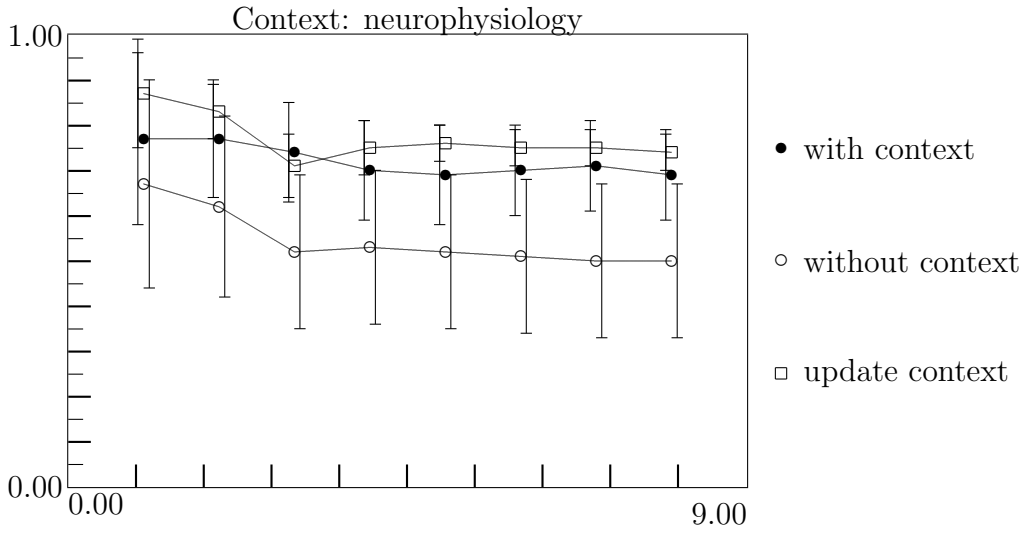

Figure 5: Precision of the results for the neurophysiology context. The query words used: *stimulus, error, neighboring, electrochemical, nucleus, memory, education, energy, depression, cerebral, structure, weak, mediation, features, atoms, neurons, molecules, reactions, anatomy, framework, concentration, hypothesis, learner, net, cells, fire, learning, system, stress, relationships.*

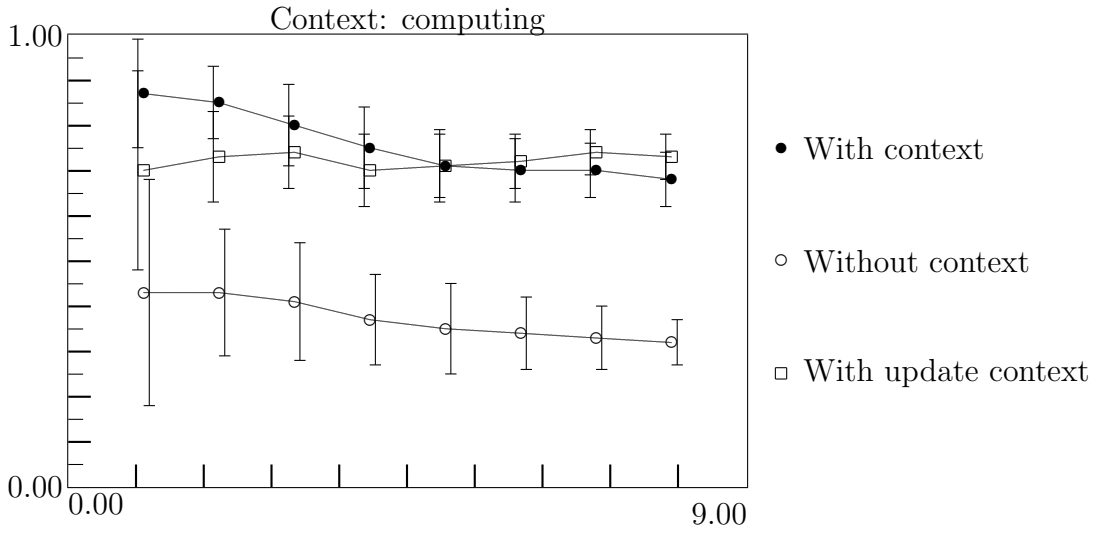

Figure 6: Precision of the results for the computing context.
The query words used: *manipulation, evaluation, product, operation, nodes, speaking, translates, sorts, multimedia, acknowledge, camera, executed, creation, sense, number, rewriting, transform, language, algorithm, technique, applicability, information, discipline, relational, treatment, systems, collect, function, framework, network.*

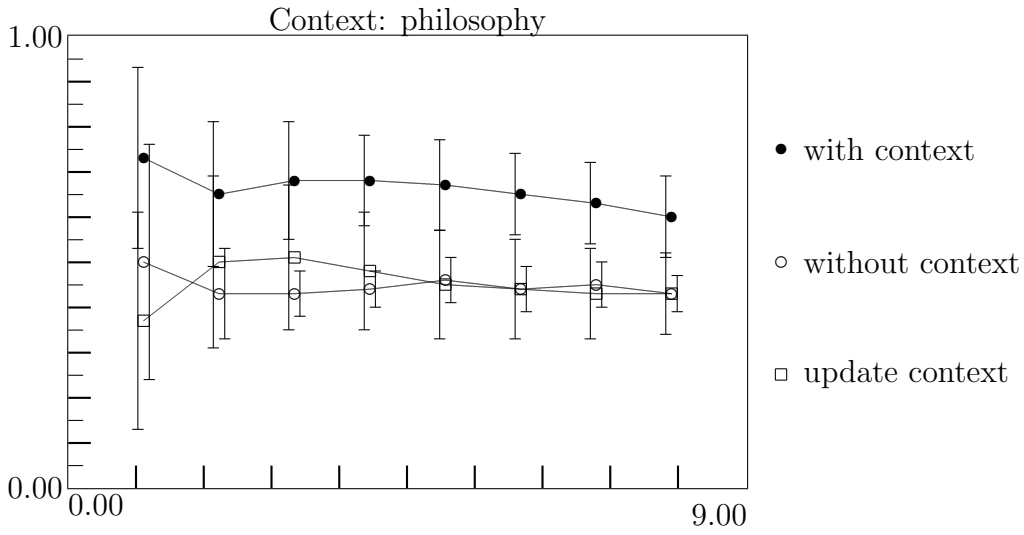

Figure 7: Precision of the results for the philosophy context.
The query words used: *treatment, transcendental, perspective, experience, adopted, fault, article, failure, error, intelligent, function, ways, doctrine, expression, negative, result, implicit, features, essential, statement, dilemma, save, teachings, formal, dramatically, universe, nervous, soul, hopes, brains.*

penalty, and no as execution of a program as it is more probable in the context of computing; the contextual query *executed executed executed executed mediator mediator multimedia unify* helped in retrieving more precise results (see figure 8). In these cases the results were very different. We could see that the context, made the difference when it comes to improving precision of the results. Nevertheless, analyzing more, it was seen that for some words the context didn't make any difference as the words were containing all the contextual information necessary for retrieving good results. For the neuropsychology field, we can give some examples of such words: *cell*, *molecule*, *atoms*, *cerebral* etc. We could observe the same things in the case of the word *algoritm* in computing and *teaching* or *doctrine* in philosophy.

As we can observe from the tests the context can significantly improve the search results. In both cases, the use of context entailed statistically significant improvement in precision but, from a superficial look at the data, the improvement appears more pronounced in the case of computing. The reasons for this might be several, some of which are probably due to the inherent characteristics of the search engine rather than to the use of context, but it is likely that, on the context side, the difference is due in good part to the different nature of the words used in the two contexts. Neurophysiologists, by and large, use neologisms with relatively uncommon Greek roots to indicate important concepts so, in this case, even a single word is sufficient to characterize the context with sufficient precision, and further contextual information brings only marginal advantages. Computing, on the other hand, tends to borrow words from other areas without modifying them, or making only superficial adjustments. That is, computing words are much more ambiguous than neurophysiological ones, and are by themselves a poor characterizer of context. In this case, we can expect that a more complete characterization of context will bring the greatest advantages, as is indeed observed in the experiments.

Nevertheless in the case of philosophy context, the updated context gave similar results to the non-contextual search. This could be because of the fact that philosophy studies general problems and, can contain words that can be found in different contexts. At this time, we don't have enough data to investigate this phenomenon in depth.

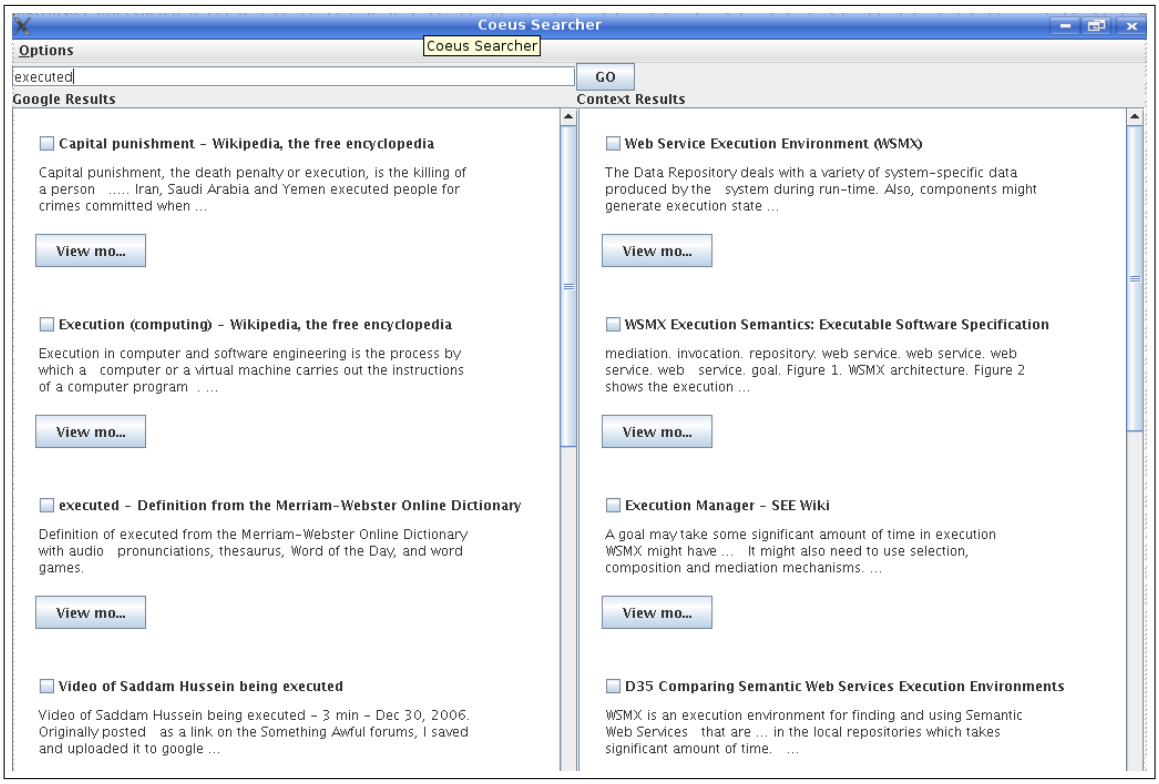

Figure 8: Results for *executed* query for the computing context.

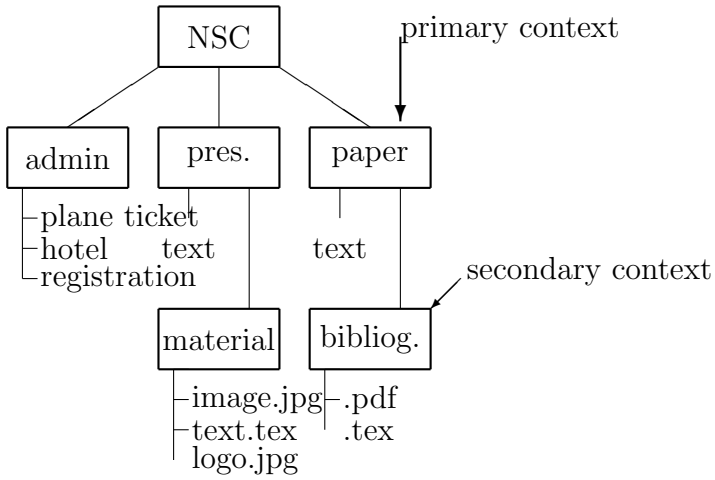

Figure 9: The structure of folders and context for the preparation of a paper.

3.2.3 A experimental case study

Let us consider the following case study example: an academic researcher is currently writing a paper for a neuropsychology conference he is planning to go. Therefore, he has somewhere on his computer a folder that contains not only some interesting papers about recently works done in his study area, but also the paper his is working at. We can get a look at his possible directory structure in figure 9.

The folder *NSC* we can find only documents that are related to the neuropsychology conference that he is planning to go. As he is writing a neuropsychology paper, the working context is represented by all the files included in the folder *paper*. In this case, the primary context is represented by the folder *paper* and the secondary context by all the papers the researcher is using as bibliography from the *bibliog.* folder.

Now, let's suppose that while he was writing his paper he wants to find more papers that he can use as reference in his work. Therefore he initiates a search using the word "mediation", as his paper speaks about mediation in neuropsychology. In figure 10 are shown part of the results he receives to his query. As it was explained previously, the left window includes the results of the user query, in this case only the text *mediation* was considered as query. The right window contains the results for the contextual query represented

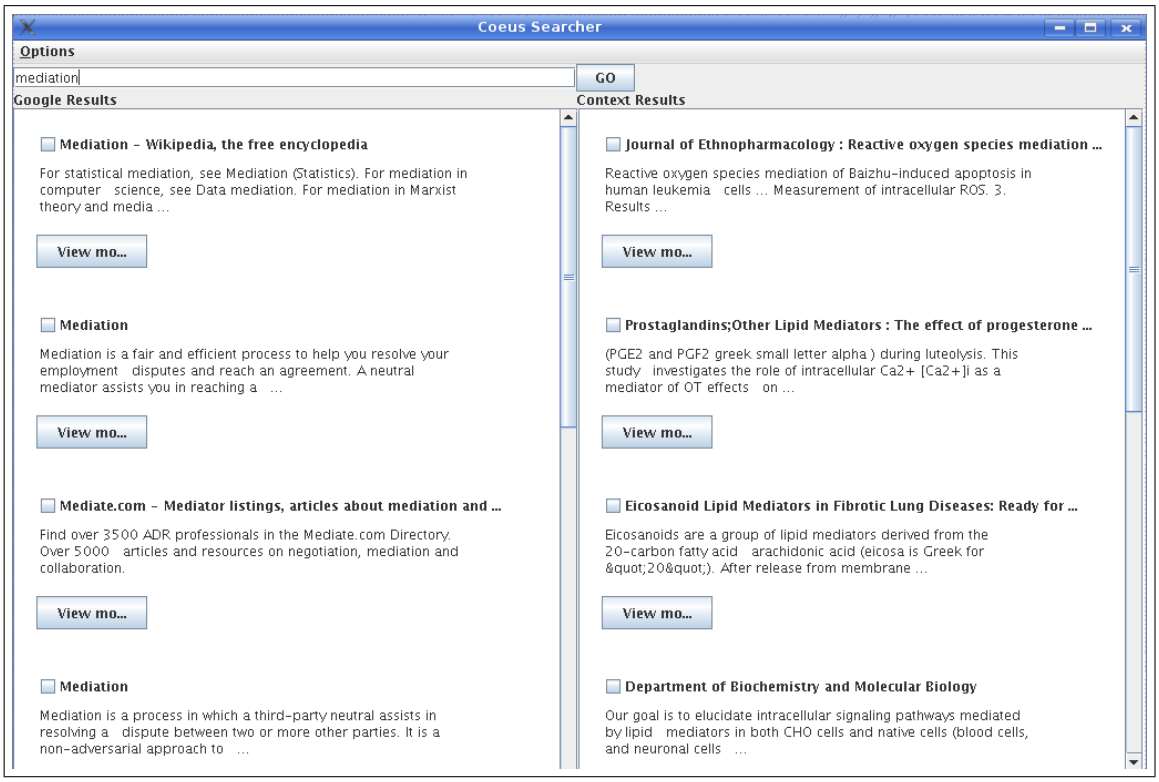

Figure 10: Results for *mediation* query for the neuropsychology context.

by the words *mediation mediation mediation mediation greek intracellular result*. The results of the non-contextual query (the *mediation* query) show that in the majority of them, the word *mediation* has the meaning of a form to get to a dispute resolution, therefore are irrelevant for our user. In the case of the contextual query, the results were more related with the working context of the user.

As we can see from the picture some words that represent the context are *cerebral*, *intracellular*, *electrochemical*. In the case of the query *mediation*, for the neuron that is closest to it and his neighborhood is applied a partial learning. The results of the partial learning for the winning neuron and his surroundings is shown in figure 12, where in red is represented the winning neuron. The contextual query obtained from the difference between the actual and the target context is *mediation mediation mediation mediation*

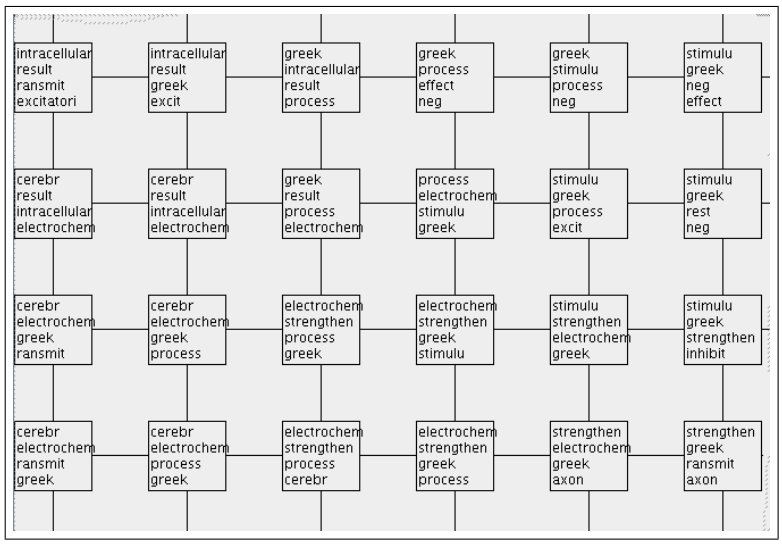

Figure 11: Part of the self-organizing map for the neuropsychology context.

mediat ransmit greek subsynapt	mediat part ransmit greek	mediat ransmit part neg
mediat ransmit part intracellular	mediat intracellular ransmit techniqu	mediat effect neg greek
mediat intracellular result greek	mediat greek intracellular result	mediat greek effect neg

Figure 12: Result of the partial learning for the winning neuron and his neighborhood.

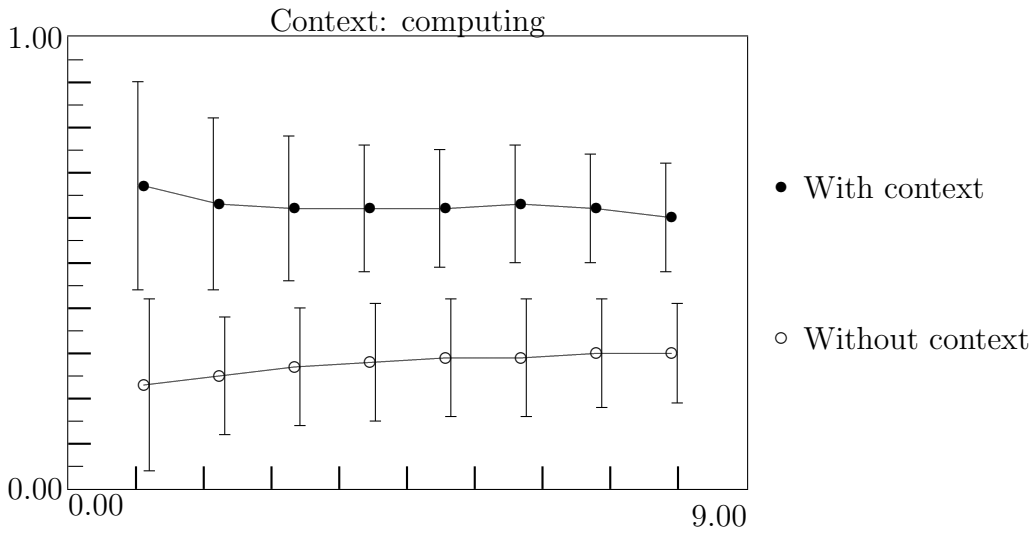

Figure 13: Precision of the image results for the computing context. The query words used: *manipulation, evaluation, product, operation, nodes, speaking, translates, sorts, multimedia, acknowledge, camera, executed, creation, sense, number, rewriting, transform, language, algorithm, technique, applicability, information, discipline, relational, treatment, systems, collect, function, framework, network.*

greek intracellular result.

3.2.4 Image search

The same approach was tested for image retrieval using Google image search engine. In order to conduct the searches we use the computing and neurophysiology context from the previous tests. The results are shown in figures 13 and 14. In each case the results were considered as relevant if the pictures, in the user judgment, could conceivably be used as a technical illustration in a presentation or a paper in the subject of the context (computing or neurophysiology). As we can see the results demonstrate a good starting point, nevertheless it must be stressed again that these are absolutely preliminary results that we use only as an indication of the viability of context use in multimedia. We do believe that a proper way of using context in this case entails the creation of multimedia contexts and the design of specialized context-sensitive search engines.

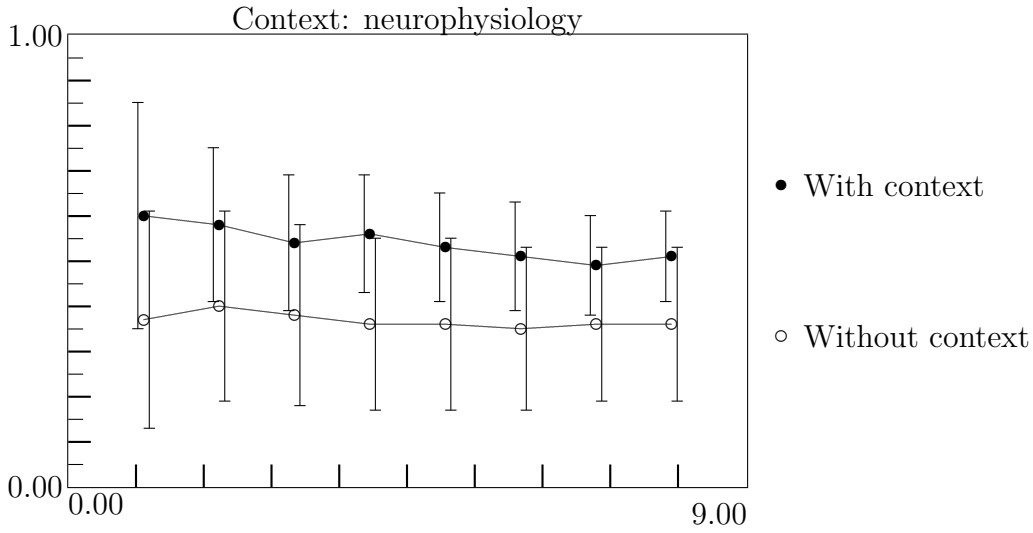

Figure 14: Precision of the image results for the neurophysiology context. The query words used: *stimulus, error, neighboring, electrochemical, nucleus, memory, education, energy, depression, cerebral, structure, weak, mediation, features, atoms, neurons, molecules, reactions, anatomy, framework, concentration, hypothesis, learner, net, cells, fire, learning, system, stress, relationships.*

4 Conclusions and future work

In this work we presented the preliminary results of a semantics search based on the formalization of the search context. This solution contrasts, on one hand, and complements, on the other hand, the current solutions studied for the *semantic web*, which are based on the formalization of the contents of the documents. We have presented results that suggest that the context can be an important factor in focalization and disambiguation of searches. As we could observe during our experiments the number of neurons of the SOM that has to be chosen from the start has a great impact on the accuracy and generalization of the map, therefore as a future work it will be of great use to know how can we choose from the start the number that best represent the context. Also, it will be interesting to find ways of conducting partial learnings on the actual context in order to obtain the target context for being able to issue contextual searches with the target context once documents have been added to the actual context. This work is a interesting starting point for a number of possible research directions, among which we underline: on one hand the creation of a search server capable of managing contextual searches, and on the other, the integration of semantic web techniques and the study of techniques for the representation of the context that can be integrated with the typical logic descriptions of the semantic web.

Acknowledgment

This work has been supported by *Consejería de Educación de la Comunidad Autónoma de Madrid*, *European Social Fund*, *Universidad Autónoma de Madrid*.

References

[1] Avtec media group. http://avtecmedia.com/internet-marketing/internet-marketing-trends.htm.

[2] Cambridge advanced learner's dictionary. http://dictionary.cambridge.org/.

[3] Google search engine. http://www.google.com.

[4] Internet systems consortium. http://www.isc.org/index.pl?/ops/ds/.

[5] Internet usage world stats internet and population statistics. http://www.internetworldstats.com/stats.htm.

[6] Merriam-webster dictionary. http://www.merriam-webster.com/.

[7] Nielsen online. http://www.nielsen-online.com/.

[8] Alemayehu, N. Analysis of performance variation using query expansion. *J. Am. Soc. Inf. Sci. Technol. 54*, 5 (2003), 379–391.

[9] Allan, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, B., Dumais, S., Fuhr, N., Harman, D., Harper, D. J., Hiemstra, D., Hofmann, T., Hovy, E., Kraaij, W., Lafferty, J., Lavrenko, V., Lewis, D., Liddy, L., Manmatha, R., McCallum, A., Ponte, J., Prager, J., Radev, D., Resnik, P., Robertson, S., Rosenfeld, R., Roukos, S., Sanderson, M., Schwartz, R., Singhal, A., Smeaton, A., Turtle, H., Voorhees, E., Weischedel, R., Xu, J., and Zhai, C. Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002. *SIGIR Forum 37*, 1 (2003), 31–47.

[10] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison Wesley, May 1999.

[11] Barthes, R. *S/Z*. Editions du Seuil, 1970.

[12] Belkin, N. J. Some(what) grand challenges for information retrieval. *SIGIR Forum 42*, 1 (2008), 47–54.

[13] Bernard Jansen, A. S., and Pfiff, M. A. A linguistical analysis of world wide web queries. *ASIS 2000: Annual Meeting of the American Society for Information Science* (2000), 169–176.

[14] Berners-Lee, T. The semantic web. *Database and Network journal 36*, 3 (2006), 7 – 10.

[15] Bharat, K. Searchpad: explicit capture of search context to support web search. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking* (Amsterdam, The Netherlands, The Netherlands, 2000), North-Holland Publishing Co., pp. 493–501.

[16] Borlund, P. The concept of relevance in ir. *J. Am. Soc. Inf. Sci. Technol. 54*, 10 (2003), 913–925.

[17] Budzik, J., and Hammond, K. J. User interactions with everyday applications as context for just-in-time information access. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces* (New York, NY, USA, 2000), ACM, pp. 44–51.

[18] Butler, D. Souped-up search engines. *Nature 405*, 6783 (May 2000), 112–115.

[19] Challam, V. K. R. Contextual information retrieval using ontology based user profiles. Master's thesis, Information and Telecommunication Technology Center, University of Kansas, 2004.

[20] Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C. Using odp metadata to personalize search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2005), ACM, pp. 178–185.

[21] Chomsky, N. Deep structure, surface structure and semantic interpretation. In *Semantics: an Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, D. Steinberg and L. Jacobovits, Eds. Cambridge University Press, Cambridge, 1971, pp. 183–216.

[22] Claypool, M., Le, P., Wased, M., and Brown, D. Implicit interest indicators. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces* (New York, NY, USA, 2001), ACM, pp. 33–40.

[23] Crestani, F., and Ruthven, I. Introduction to special issue on contextual information retrieval systems. *Inf. Retr. 10*, 2 (2007), 111–113.

[24] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41* (1990), 391–407.

[25] Doctorow, C. Metacrap: Putting the torch to seven straw-men of the meta-utopia. http://www.well.com/~doctorow/metacrap.htm, August.

[26] Dumitrescu, A., and Santini, S. Context based semantics for multimodal retrieval. To appear.

[27] Fallows, D. Search engine users: Internet searchers are confident, satisfied and trusting – but they are also unaware and naive. http://www.pewinternet.org/ report_display.asp?r=146, 2005.

[28] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst. 20*, 1 (2002), 116–131.

[29] Gadamer, H.-G. *Truth and method.* Continuum, London and New York, 1975.

[30] Gauch, S., Chaffee, J., and Pretschner, A. Ontology-based personalized search and browsing. *Web Intelli. and Agent Sys. 1*, 3-4 (2003), 219–234.

[31] Glover, E. J., Lawrence, S., Birmingham, W. P., and Giles, C. L. Architecture of a metasearch engine that supports user information needs. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management* (New York, NY, USA, 1999), ACM, pp. 210–216.

[32] Glover, E. J., Lawrence, S., Gordon, M. D., Birmingham, W. P., and Giles, C. L. Web search—your way. *Commun. ACM 44*, 12 (2001), 97–102.

[33] Greisdorf, H. Relevance: An interdisciplinary and information science perspective. *Informing Science 3*, 2 (2000), 67–71.

[34] Gruber, T. Ontology of folksonomy: A mash-up of apples and oranges. *Int'l Journal on Semantic Web & Information Systems 3*, 2 (2007).

[35] Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud. 43*, 5-6 (1995), 907–928.

[36] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (1998), Morgan Kaufmann, pp. 256–265.

[37] Höscher, C., and Strube, G. Web search behavior of internet experts and newbies. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking* (Amsterdam, The Netherlands, The Netherlands, 2000), North-Holland Publishing Co., pp. 337–346.

[38] J. R. Wen, J. N., and Zhang, H. J. Query clustering using user logs. *ACM Trans. Inf. Syst. 20*, 1 (2002), 59–81.

[39] Jansen, B. J., Spink, A., and Saracevic, T. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage. 36*, 2 (2000).

[40] Jenkins, C., Corritore, C. L., and Wiedenbeck, S. Patterns of information seeking on the web: A qualitative study of domain expertise and web expertise. *IT and Society 1*, 3 (2003), 64–89.

[41] Jones, G., and Brown, P. The role of context in information retrieval. In *Proceedings of the Information Retrieval in Context Workshop, SIGIR IRiX 2004* (Sheffield, UK, July 2004).

[42] Jones, G. J. F., and Brown, P. J. Information access for context-aware appliances (poster session). In *SIGIR '00: Proceedings of the*

23rd annual international ACM SIGIR conference on Research and development in information retrieval (New York, NY, USA, 2000), ACM, pp. 382–384.

[43] Käki, M. Findex: search result categories help users when document ranking fails. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2005), ACM Press, pp. 131–140.

[44] Kaski, S. Computationally efficient approximation of a probabilistic model for document representation in the websom full-text analysis method. *Neural Process. Lett. 5*, 2 (1997), 139–151.

[45] Kent, A., Berry, M. M., Luehrs, and Perry, J. W. Machine literature searching viii, operational criteria for designing information retrieval systems. *American Documentation 6*, 2 (1955), 93–101.

[46] Kohonen, T. *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*. Springer, Berlin, Germany, 1995.

[47] Lawrence, S. Context in web search. *IEEE Data Engineering Bulletin 23*, 3 (2000), 25–32.

[48] Leroy, G., Lally, A. M., and Chen, H. The use of dynamic contexts to improve casual internet searching. *ACM Trans. Inf. Syst. 21*, 3 (2003), 229–253.

[49] Ma, Z., Pant, G., and Sheng, O. R. L. Interest-based personalized search. *ACM Trans. Inf. Syst. 25*, 1 (2007), 5.

[50] Magennis, M., and van Rijsbergen, C. J. The potential and actual effectiveness of interactive query expansion. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1997), ACM, pp. 324–332.

[51] Manber, U., Smith, M., and Gopal, B. Webglimpsecombining browsing and searching. In *In: Proceedings of the USENIX 1997 Annual Technical Conference* (1997).

[52] Mizzaro, S. Relevance: The whole history. *Journal of the American Society for Information Science 48*, 9 (1997), 810–832.

[53] Porter, M. F. An algorithm for suffix stripping. *Program 14*, 3 (1980), 130–137.

[54] Rhodes, B. J., and Starner, T. Remembrance Agent: A Continuously Running Automated Information Retrieval System. In *Proceedings of the 1st International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technologies* (1996), pp. 487–495.

[55] Rodríguez, J. C. Jugadas, partidas y juegos de lenguaje: el significado como modificación del contexto. *Asunción:Centro de documentos y estudios* (2003).

[56] Ruthven, I. Re-examining the potential effectiveness of interactive query expansion. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (New York, NY, USA, 2003), ACM, pp. 213–220.

[57] Santini, S. The self-organizing field. *IEEE Transactions on Neural Networks 7*, 6 (1996), 1415–23.

[58] Santini, S., and Dumitrescu, A. Context as a non-ontological determinant of semantics. In *3rd International Conference on Semantics and Digital Media Technologies* (Koblenz, Germany, December 2008). To appear.

[59] Santini, S., and Dumitrescu, A. Context-based retrieval as an alternative to document annotation. *6th International Conference on Language Resources and Evaluation* (May 2008).

[60] Saracevic, T. Evaluation of evaluation in information retrieval. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1995), ACM, pp. 138–146.

[61] Schwarz, S. *A Context Model for Personal Knowledge Management Applications*, vol. 3946. April 2006.

[62] Shen, X., Tan, B., and Zhai, C. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2005), ACM, pp. 43–50.

[63] Shen, X., Tan, B., and Zhai, C. Implicit user modeling for personalized search. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management* (New York, NY, USA, 2005), ACM, pp. 824–831.

[64] Sperber, D., and Wilson, D. *Relevance : communication and cognition*. Blackwell, Oxford [Oxfordshire], 1986.

[65] Speretta, M., and Gauch, S. Personalizing search based on user search history. In *Proceedings of CIKM 2004* (2004).

[66] Spink, A., and Jansen, B. J. A study of web search trends. *Webology 1*, 2 (2004).

[67] Spink, A., Ozmutlu, S., Ozmutlu, H. C., and Jansen, B. J. U.s. versus european web searching trends. *SIGIR Forum 36*, 2 (2002), 32–38.

[68] Spink, A., and Saracevic, T. Interaction in information retrieval: selection and effectiveness of search terms. *Journal of the American Society for Information Science 48* (1997), 741–761.

[69] Teevan, J., Dumais, S. T., and Horvitz, E. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2005), ACM, pp. 449–456.

[70] Vishnu, K. *Contextual Information Retrieval Using Ontology Based User Profiles*. PhD thesis, 2005.

[71] White, R. W., Ruthven, I., and Jose, J. M. Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *SIGIR '02: Proceedings of the 25th annual international*

ACM SIGIR conference on Research and development in information retrieval (New York, NY, USA, 2002), ACM, pp. 57–64.

[72] Wilkinson, R., and Wu, M. Evaluation experiments and experience from the perspective of interactive information retrieval. In *In the Proceedings of the Third Workshop on Empirical Evaluation of Adaptive Systems* (2004), pp. 221–230.

[73] Yang, Y., and Padmanabhan, B. Evaluation of online personalization systems: A survey of evaluation schemes and a knowledge-based approach. *J. Electronic Commerce Res. 6* (2005), 112–122.