Identifying the Multiple Contexts of a Situation

Aviv Segev

Technion - Israel Institute of Technology, Haifa 32000, Israel
asegev@technion.ac.il

Abstract. The paper presents a contexts recognition algorithm that uses the Internet as a knowledge base to extract the multiple contexts of a given situation, based on the streaming in text format of information representing the situation. Context is represented here as any descriptor most commonly selected by a set of subjects to describe a given situation. Multiple contexts are matched with the situation. The algorithm yields consistently good results and the comparison of the algorithm results with the results of people showed that there was no significant difference in the determination of context. The algorithm is currently being implemented in different fields and in multilingual environments.

Keywords: Matching context, Context recognition, Metadata, Text analysis.

1 Introduction

The question of context recognition is defined as one of the main questions addressed by the international interdisciplinary context community [1]. A context is a descriptor (such as a word or an image) or a set of descriptors that defines a situation. A context can convey a different facet, a different point of view, or a different understanding of a situation. Therefore, many situations are characterized by several different contexts. This paper presents an algorithm of contexts recognition that analyzes a given situation, represented in text format, and identifies its multiple contexts, textual descriptors. The performance of the algorithm was compared to the human process of multiple contexts recognition and yielded consistently good results. The algorithm is now being implemented in different fields and in a multilingual eGovernment project.

Section 2 reviews related works in the literature. Section 3 presents a formal definition of contexts recognition and divides the problem into two sub-problems. Section 4 describes the contexts recognition algorithm, which consists of five main processes: the collection of data, the selection of contexts for each text, the ranking of the contexts, the identification of the current contexts, and the clustering to achieve the multiple contexts, and describes the use of the Internet as a context database. Section 5 presents the analysis of the algorithm. Section 6 discusses the applications of the algorithm in different domains of knowledge and in different languages. Finally, section 7 presents some conclusions and directions of future research.

2 Related Work

2.1 Formal Definition of Context

Context is defined as a descriptor (such as a word or an image) or set of descriptors that can represent a situation or a scenario. A scenario is defined as ”the world state”, a

T.R. Roth-Berghofer, S. Schulz, and D.B. Leake (Eds.): MRC 2005, LNAI 3946, pp. 118–133, 2006.

situation that is a snapshot or an instance of the world at some given time, namely, all attributes of the world, including all objects, their properties and internal states, and the relationships between them [21].

McCarthy [18] defined the formalization of the notion of context as one of the main problems in the field of artificial intelligence (AI) and argued that a most general context does not exist. Consequently, McCarthy [19] worked to formalize context and to develop a theory of introducing context as formal objects.

Since a situation is characterized by many different features, it may have multiple contexts. McCarthy's formal definition of context is used in this work to identify the multiple contexts of a situation.

2.2 Blackboard Model

The model architecture for the contexts recognition is based on the blackboard model [7]. The blackboard model has been used in many AI applications, e.g., understanding images [24], signals [9], and speech [16], to represent possible solutions to a given problem. Blackboard is used here to enhance information extraction from more than one information source. For example, different information sources can be multiple people having multiple conversations at the same place and time, as in the case of Internet chats.

2.3 Context Extraction

Since virtually every application requires the use of context, whether explicitly or implicitly, it is necessary to have means by which to extract it. Bauer and Leake [4] developed WordSieve, an algorithm for automatically extracting information about the context in which documents are consulted during web browsing. Using information extracted from the stream of documents consulted by the user, the WordSieve algorithm automatically builds context profiles that differentiate sets of documents that users tend to access in groups. These profiles are used in a research-aiding system to index documents consulted in the current context and pro-actively suggest them to users in similar future contexts.

Another approach created taxonomies from metadata (in XML/RDF) containing descriptions of learning resources [20]. After the application of basic text normalization techniques, an index was built, observed as a graph with learning resources as nodes connected by arcs labeled by the index words common to their metadata files. A cluster mining algorithm is applied to this graph and then the controlled vocabulary is selected statistically. However, a manual effort is necessary to organize the resulting clusters into hierarchies. When dealing with medium-sized corpora (a few hundred thousand words), the terminological network is too vast for manual analysis, and it is necessary to use data analysis tools for processing. Therefore, Assadi [2] employed a clustering tool that utilizes specialized data analysis functions and clustered the terms in a terminological network to reduce its complexity. These clusters are then manually processed by a domain expert to either edit them or reject them.

We propose the use of a fully automatic contexts recognition algorithm that uses the Internet as a knowledge base and as a basis for clustering.

2.4 Information Seeking

Information seeking is the process in which people turn to information resources to increase their level of knowledge regarding their goals [8]. Although the basic concept of information seeking remains unchanged, the growing need for the automation of the process has called for innovative tools to assign some of the tasks involved in information seeking to the machine level. Thus, techniques for information seeking based on textual information are used, including the ontology tools Text-To-Onto [17], OntoMiner [11], and TexaMiner [12], to name a few, and databases are extensively used for the efficient storage and retrieval of information.

The Internet can be seen as a large database that is constantly being modified and updated. Many information seeking techniques have been developed to retrieve information from the Internet. For example, Valdes-Perez and Pereira [23] developed an algorithm based on the concise all pairs profiling (CAPP) clustering method. This method approximates profiling of large classifications. The use of hierarchical structure was explored for classifying a large, heterogeneous collection of web content [6]. Another method involves checking the frequency of the possible keyphrases of articles using the Internet [22]. However, this method is based on an existing set of keywords and uses the Internet for ranking purposes only.

The present algorithm attempts to automate contexts recognition, based on information seeking techniques, using the Internet as a database for possible multiple contexts. The algorithm differs from previous text analysis techniques by allowing the input to be received from multiple sources, in an unstructured format. In addition, the algorithm utilizes data resources that are independent of the user and are constantly changing to analyze the information.

3 Formal Definition of the Problem

A scenario can be characterized by multiple contexts, each describing a different facet of the situation.

McCarthy [19] formalized context as first class objects with the following basic relation:

ist($\mathcal{C}$, P) meaning that the proposition P is true in the context $\mathcal{C}$.

In this paper, context is defined as any textual description that is most commonly selected by a set of subjects to describe a given situation and multiple contexts are a set of such contexts:

Let $P_1, P_2, ..., P_m$ be a series of textual propositions defining situation S.

Contexts $C_1, C_2, ..., C_k$ are defined as the contexts of situation S if:

$\exists n$ subjects, $n \geqslant 1$ so for the majority of n selected

ist($\mathcal{C}_i$, P_j) $\forall$ i, for a given j

(Contexts $C_1, C_2, ..., C_k$ are true for textual proposition P_j)

For a series of propositions there exists a collection of sets of contexts.

Let $P_1, P_2, ..., P_m$ be a series of textual propositions when $\forall$ P_i there exists a collection of sets of contexts C_{ij} so that:

$\forall$ i, ist($\mathcal{C}_{ij}$, P_i) $\forall$ j meaning that the textual proposition P_i is true in each of the set of contexts $\mathcal{C}_{ij}$. $\mathcal{C}_{ij}$ are not predefined hierarchically in a structure such as a tree. However, hierarchical structures can be built according to a specific set of textual propositions.

The main research problem is formally defined as:

What is the outer context $\mathcal{C}$, the multiple contexts of a scenario, defined by $ist(\mathcal{C}, \bigcap_{i=1}^{m} ist(\mathcal{C}_{ij}, P_i))\ \forall j$.

The number of existing contexts is assumed to be finite and to satisfy $\mathcal{C}, \mathcal{C}_{ij} \subseteqq U_c$ (unity of all existing contexts)

The main research problem can be divided into two sub-problems:

1. Let P be a given single text. What are the possible contexts $\mathcal{C}_i$ that satisfy ist($\mathcal{C}_i$, P) $\forall$ i (for single text P all contexts $\mathcal{C}_i$ are true)
2. Let $P_1, P_2, ..., P_m$ be a set of texts that satisfy the following condition: for each text P_i there exists a set of contexts $\mathcal{C}_{ij}$ so that ist($\mathcal{C}_{ij}$, P_i) $\forall$ j.
   What is the outer context $\mathcal{C}$ so that $ist(\mathcal{C}, \bigcap_{i=1}^{m} ist(\mathcal{C}_{ij}, P_i))\ \forall j$.

The division of the problem into two parts allows a solution of the first part to be acquired by information seeking through the Internet. The second part is addressed using an algorithm that ranks the contexts according to the importance of the information retrieved in the first part. The result of the following algorithm is a list of contexts, the outer context of the situation, which is the multiple contexts of the situation.

4 The Contexts Recognition Algorithm

The research algorithm is based on the streaming in text format of information that represents input from different sources, such as Internet chats. The contexts recognition algorithm output is a set of contexts that attempt to describe the current situation most accurately. The set of contexts is a list of words or phrases, each describing an aspect of the situation. Thus, multiple contexts can be matched to a given situation. The algorithm consists of five major processes:

- Collecting Data - The information from the information sources is decomposed into words and the keywords are extracted from them.
- Selecting Contexts for Each Text (Descriptors) - For each keyword a set of preliminary contexts is extracted from the Internet, which is used as a context database.
- Ranking the Contexts - Each preliminary context is ranked according to the number of references it receives in the context database and the number of appearances it has in the text.
- Identifying the Current Contexts - The preliminary contexts that have significantly higher numbers of references and higher numbers of appearances are included in the current set of contexts.
- Obtaining the Multiple Contexts - The current contexts are examined for synonyms and synonymous contexts are united.

4.1 Collecting Data

The input text was used as is; all misspelled words were left in the text. The text was parsed at the granularity of sentences. Long sentences were parsed according to the

maximum number of words that could be used in a search engine. Each text is decomposed into single words, when words are letter strings separated by spaces, and all punctuation is removed from the text. Then the words are checked according to a set of dictionaries. The first dictionary is a "Stop List", consisting of words that do not add to the understanding of the context, such as I, me, in, are, the. All words that appear in this dictionary are ignored. The next step uses a set of dictionaries according to fields of knowledge to sieve the words that are not related to the specific field of knowledge. If the word appears in the field of knowledge dictionary, then it is added to the list of keywords that are searched in the context database, otherwise it is ignored. This process continues for each word in the text. After each text passes through this module, the algorithm sends a list of words to be checked for a possible set of contexts. Checking against a dictionary can be skipped if the field of knowledge is unknown, but skipping this step may sometimes lead to less accurate results. The difficulty does not lie in finding the possible keywords since the algorithm can always use the whole input corpus.

Algorithm 4.1: COLLECTING DATA($TextualData$)

Parse data according to the granularity of sentences
Replace punctuation with a new line
Eliminate words which appear in "Stop List"
if Field of Knowledge is defined
  then Eliminate words that do not appear in field on knowledge dictionary
if words in line $>$ maximum words for search engine
  then Create new lines according to maximum words
for each new line
  do Activate the next algorithm

4.2 Selecting Contexts for Each Text (Descriptors)

The selection of the current context is based on a search through the database for all relevant documents according to keywords and on the clustering of the results into possible contexts. Once a list of keywords exists, each keyword is searched in the context database - the Internet - and a set of contexts is extracted. This creates a list of preliminary contexts for each keyword. The contexts in this work were represented by words or sets of words, which can be viewed as meta data created for each set of Internet web pages. The Internet can then be viewed as an immense set of words that represent different possible contexts, each associated with its respective web page. Other descriptors can include images appearing on the Internet. The Internet can then be seen as a vast set of descriptors that represent different possible contexts, each associated with its respective web page. The full list of preliminary contexts for all the keywords includes all the possible contexts for this current text.

Any search engine can be used and any Term Frequency / Inverse Document Frequency [10] method for clustering can be implemented. The current application of the algorithm uses the concise all pairs profiling (CAPP) clustering method. [23], as it is applied in the Vivisimo search engine.

The use of the Internet as a context database instead of a precalculated frequencies base has several advantages. The use of the Internet does not require the constant up-

dating and maintenance of a database. The precalculated frequencies base requires the user to work in a limited predefined knowledge domain. The Internet can serve as an unlimited knowledge domain that is continuously being updated.

This step results in a long list of preliminary contexts, many of which are irrelevant to the context. The purpose of the following steps is to minimize the list and identify most relevant contexts of the situation.

Algorithm 4.2: SELECTING CONTEXTS($ListofWords$)

Check Internet search engine with *List of Words*
for each Internet page extracted
  do Activate Ranking Contexts algorithm
Add 1 to *Number of Appearances* for each context identified

4.3 Ranking the Contexts

The algorithm checks the number of appearances in the text for each preliminary context in the set of preliminary contexts. The contexts are also examined for the number of Internet documents that refer to the set of documents. The set of contexts is now ranked according to both the number of references in the text and the number of references in the documents.

These two metrics were selected since the number of appearances in the text represents how many times each preliminary context was mentioned in the situation. The number of references in the Internet represents how important the preliminary context is to the general population that uses the Internet.

New preliminary contexts can now be created according to textual sub-strings of existing preliminary contexts. This step sums up the number of documents referring to the preliminary contexts. Multiple reference pages from similar web sources are counted as one instance. Each document usually refers to multiple contexts, consequently creating a long list of preliminary contexts. The last step involves ranking the set of preliminary contexts according to both the number of references from the documents and the number of appearances in the text. This step maps all the preliminary contexts to a two dimensional graph, allowing the contexts that receive very high ranking in both characterizations to be located, as in Figure 1.

After each session of ranking, the list is used for two purposes - resetting the set of preliminary contexts and identifying the current context. The current list of contexts joins the new preliminary contexts arriving from the continuously streaming text. The lists are united and the ranking process is repeated. In parallel to the repetition of the ranking algorithm, the set of ranked preliminary contexts is forwarded to the next module to determine the current contexts.

Algorithm 4.3: RANKING CONTEXTS($InternetPageExtracted$)

Perform term frequency clustering
for each term
  do { **if** term not in *Preliminary Context List*
    then Add term to *Preliminary Context List*

4.4 Identifying the Current Contexts

The output of the ranking stage is the current context or a set of highest ranking contexts that differ essentially by rank. The algorithm then returns to the first step to collect more texts and feed them again to the database. The set of preliminary contexts that has the top number of references, both in number of Internet pages and in number of appearances in all of the texts, is defined as the highest ranking and is identified to be the current contexts.

The current contexts received from the previous stage can be depicted on a graph according to the number of appearances and the number of references, as in Figure 1.

The algorithm for detecting the current contexts includes the following steps:

Algorithm 4.4: DETECTING CURRENT CONTEXTS($PreliminaryContexts$)

Organize the list of preliminary contexts in descending order according to number of references appearing in the Internet - the Set of Documents.

Find the difference between each value of the number of references and its nearest lower value neighbor, defined as Current References Difference Value (CRDV).

Find the difference between each value of the number of appearances and its nearest lower value neighbor, defined as Current Appearances Difference Value (CADV).

Weight the number of appearances in the text and the number of references in the Internet according to the following formula:
MVR = Maximum Value of References
MVA = Maximum Value of Appearances
WeightedValue = $\sqrt{(\frac{2*CADV*MVR}{3*MVA})^2 + (CRDV)^2}$

Find the maximum value of the Weighted Value. If the maximum Weighted Value is the first value, then continue to the next one, since frequently the first value is too far from its neighbor.

Select all the contexts that appear before the maximum Weighted Value in the list that was organized in the first step as the current contexts. Store current selected contexts.

Erase the selected contexts from the list and repeat the previous two steps.

The Weighted Value can be viewed as the weighted distance to the origin. However, the index of number of references is on a much larger scale than the index of the number of appearances and therefore it is not possible to retain the original proportions and it is

necessary to re-scale the indices. The value is calculated multiplying by the maximum number of references and dividing by the number of maximum number of appearances. A constant of 2/3 was found by experiment to be appropriate for the re-adjustment of the figures.

The first cluster of contexts near the origin includes all the contexts that received low ranking both in number of appearances and in number of references. This group of contexts includes most of the contexts in the list. Since the contexts in this group received low ranking they are eliminated from the list. The remaining contexts are the current contexts.

The process can continue until all the contexts in the list are covered and this will yield all the possible preliminary contexts. However, in most cases the best results were already achieved when the last two steps were performed twice. Further repetitions, which increase the number of results, were unnecessary. The ranking according to number of references and not according to the weighted value also improves the results. This indicates that the number of appearances of the context in the text has less value in the determination of the context than the number of references in the Internet.

During the implementation of the algorithm, there was a problem that required special consideration. The contexts that received lower ranking than the top ranking contexts in the cluster but were not part of the cluster were kept. Namely, these are contexts that receive lower ranking in either the number of appearances or the number of references than the top ranking contexts, but not in both. Running the algorithm showed that these contexts are sometimes relevant and should be kept.

4.5 Obtaining the Multiple Contexts

The current contexts are examined for synonyms using a thesaurus and synonymous contexts are united. Before this step, many of the contexts identified by the algorithm are similar in meaning. The algorithm also looks for semantic similarity. This step enables the algorithm to identify the differing multiple contexts of the situation and thus facilitates the better description of the situation.

Algorithm 4.5: MULTIPLE CONTEXTS CLUSTERING($CurrentContexts$)

for each $x \in$ *CurrentContexts*
do Examine the *CurrentContexts* for synonyms in Thesaurus
Examine the *CurrentContexts* for semantic similarity
if synonyms / semantic similarity found
then Unite contexts and unite their weights
return (United Contexts,Relevant Weights)

The current set of contexts is the output of the algorithm. However, since the algorithm is continuous, the contexts continue updating as long as new textual input continues to be accessed by the algorithm.

4.6 Examples

Example 1. The example presents the implementation of the algorithm on text taken from MSN chat and the results of the algorithm.

Lestat: Question on Linux how much ram can it run
WickedWeekend: like i said im new at this
Xor: ifconfig eth0 192.168.0.1 netmask 255.255.255.0
Xor: lestat virtually any amount of RAM
Xor: if u adjust kernel to it
WickedWeekend: ok Xor
Lestat: well you know my computer 1.53gb of ram
WickedWeekend: to me just looking at it and not being an expert
Xor: lestat it will suffice
Xor: wykd
Xor: yes there are many other command
Xor: but good thing about linux
Xor: is that u cna make aliases to commands
Xor: so instead of ifconfig eth0 192.168.0.1 netmask 255.255.255.0
Xor: u can make an alias
Xor: in form
WickedWeekend: if your ethernet configuration ip is 192.168.0.1 you want it to have a mask of 255.255.255.0 which is a generic one i think instead of broadcasting your native ip?
Xor: dothis
Xor: u cna have any subnet mask
Xor: u want
Xor: u can mak eur own subnetmasks
Xor: 255.255.248.0
Xor: or whatever
WickedWeekend: can they be virtualy anything as long as they are in the correct format?
WickedWeekend: but have to start 255.255.....
Xor: 255.255 not necesserily
Xor: u can have
Xor: 255.248.0.0
Xor: anythign really
WickedWeekend: but the fist one is always 255?
WickedWeekend: or not?
Xor: well yes it should be
Xor: i never saw another format
WickedWeekend: ok

First the input is read one line at a time. Each word is separated by a space. Punctuation marks are eliminated. Each word is checked against the ”Stop List” dictionary. In this case each word was checked in a predefined computer dictionary.

The words that passed the previous stage serve as keywords. After each step (change of speaker) the keywords are sent to the search engine and clustered into a list of preliminary contexts. These steps are repeated 34 times, yielding 222 preliminary contexts that have at least two references in the Internet and are relevant to keywords that appeared at least once in the text.

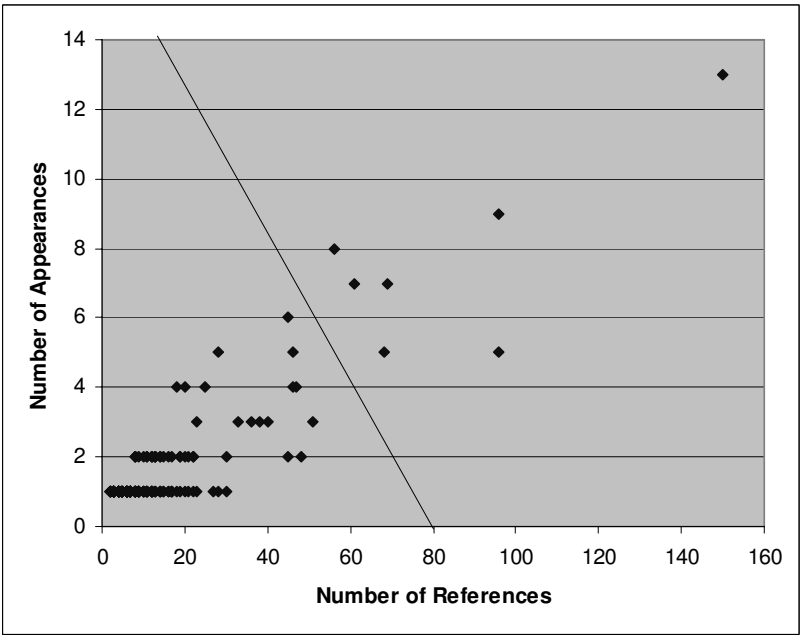

Fig. 1. Identifying the Current Contexts

The algorithm maps all the preliminary contexts to a two dimensional graph, allowing the contexts that receive very high ranking in both characterizations to be located, as in Figure 1.

The set of identified contexts includes: Linux(150,13), Subnet(96,5), Forums(96,9), Review(69,7), FAQ(68,5), Blog(61,7), and Network(56,8). The value in the parentheses includes the number of references and the number of appearances respectively. Figure 1 shows the weighted value calculated for each context. The differences between the weighted values in the figure show that the maximum weighted value is after Network, resulting in the current contexts.

The chat can be viewed as having multiple contexts. The contexts Network and Subnet represent the communication point of view discussed between two participants in the chat. At the same time, a discussion about Linux is being held. Looking from a broader perspective, the contexts of Forums and Frequently Asked Questions (FAQ) can be viewed as the contexts of the conversation.

Example 2. The example presents the implementation of the algorithm on text taken from MSN chat and the results of the algorithm. The algorithm was run without any information about the knowledge domain and did not use any field of knowledge dictionary.

Fox-Fire1: i want to ask some question about hacking
Fox-Fire1: any body help me
simply-crazy: h word is bad

mad-for-computers: what ques about haciking
mad-for-computers: hacking
mad-for-computers: i love hacking
Fox-Fire1: some body hack my id
Demon11: !illegal
Obis-Shadow: Please note: Owner, hosts and participants of this room we don't offer any help with illegal activity. This includes p2p software.(such as Kazaa and Imesh). any discussion of such activities will result in banishment from chat.
Fox-Fire1: and i want to get it back
Fox-Fire1: ohh
Fox-Fire1: hi
Fox-Fire1: what happened
mad-for-computers: nothing
mad-for-computers: there was room named h but it is gone
Fox-Fire1: o k tell me how i can get back my id

The results of algorithm in the example included the contexts of Hacker and Security. This shows that preliminary contexts can also be words that did not appear in the text itself but were a result of the clustering of the web pages, as in the top ranking context of Security. The contexts of Hacker and Security are the different contexts of this chat, representing different complementary facets of the conversation.

Demon11 and Obis-Shadow are software bots that monitor the chat. Eventually the participant Fox-Fire1 was banished from the chat as a result of raising an illegal discussion topic - hacking, although the participants were discussing security measures against hacking.

5 Analysis

The objective was to analyze the different multiple perspectives, or contexts, of a given scenario, with minimal restrictions placed on the human subjects evaluating the scenario so as not to direct them to a single perspective. The contexts recognition algorithm was evaluated using computer-related Internet chats acquired from MSN chats. The chats included several participants and were observed over time. Parts of the chats that dealt with topics concerning computers were copied to files. From these chats sets of files were randomly selected to be analyzed by the algorithm. These files were fed as input into the contexts recognition algorithm. The results were compared with the results given by computer-literate subjects.

The subjects who answered the survey were graduate students with at least basic knowledge of computer terminology. A total of twenty subjects participated in the survey. The participants received a set of three chats. This allowed two groups of participants to be formed, each group with a different set of chats. Each chat had at least nine replies. The maximum number of replies per chat was eleven. An average of ten subjects determined the list of contexts for each chat. A total of six chats in computer related topics were analyzed in this way. The subjects were presented with the above chats and were asked to provide a list of contexts for the chats. The subjects were told that the text was obtained from Internet chats and was presented to them as is, including spelling

mistakes and Internet acronyms. The subjects were asked to write in their own words what they felt were the contexts. These words were not selected from a list. The words were counted to determine the number of times they appeared among all the subjects. Each word was counted once for each subject who mentioned the context. The context was ranked in descending order according to the number of times that the subjects mentioned the context. The list of best-ranked contexts was compared with the results yielded by the contexts recognition algorithm. The other contexts mentioned more than once by the subjects were also compared to check the sensitivity of the algorithm.

Table 1 summarizes the comparison of the results of the algorithm and the results of the subjects. The table also displays the average results of the contexts recognition algorithm for all six cases examined. The second and third sets of contexts, which are contexts mentioned by the subjects but not selected by a majority of the people or only selected by one or two subjects, respectively, are also compared with the second and third reiterations of the algorithm to check the sensitivity of the algorithm.

Table 1. Ranked Contexts (RC) of the Algorithm

Contexts Recognition	Chat 1	Chat 2	Chat 3	Chat 4	Chat 5	Chat 6	Average
Top RC	100%	100%	100%	100%	100%	100%	100%
Top and II RC	100%	84.46%	65.22%	57.15%	75%	100%	80.31%
Top, II, and III RC	86.67%	81.25%	57.15%	40%	76.92%	75%	69.50%

The top ranking contexts mentioned by the subjects were identified as contexts by the algorithm. In most of the cases the contexts that were ranked among the highest by the subjects were also ranked among the highest by the algorithm. Some of the other contexts generated by the algorithm were not selected by the subjects. In addition, a few of the lower ranking contexts mentioned by the subjects were missed by the algorithm. Table 1 shows that for the top ranking contexts the algorithm yields very high results. As more contexts that received lower ranking by the subjects are added, the results of the algorithm degenerate. The algorithm needs improvement in deducting better results in the second and third ranking sets of contexts.

The significance of the results was analyzed using the identical populations test. The test for homogeneity is designed to test the null hypothesis that two or more random samples are drawn from the same population or from different populations, according to some criterion of classification applied to the samples. The Chi-square Pearson Test for Association is a test of statistical significance. The results of the identical populations test comparing the groups containing the algorithm as a subject with the original group consisting only of human subjects showed that they were almost identical populations. In other words, if the computer is part of the group, the context will remain identical. Hence, there is no significant difference in the determination of contexts between the algorithm and the human subjects. Table 2 displays the identical population test results for each of the six chats.

The complexity of the algorithm is $\theta(kn)$ where n represents the number of input cycles such as each line of text or each time that input is received from a different source. The k represents a constant limiting the number of top ranking results from each

Table 2. Identical Populations Test

Chat	χ^2	P-Value
1	0.065199	1
2	0.568693	0.999
3	0.187391	1
4	0.256795	1
5	0.133712	1
6	0.273300	0.998

cycle of the algorithm. This allows different levels for the monitoring of the amount of data the algorithm handles.

6 Applications

6.1 Applicability in Multiple Domains

The contexts recognition algorithm is versatile in terms of its utility in multiple domains of knowledge. The algorithm was extensively analyzed using Internet chats on a wide variety of topics, including health-oriented chats and computer-oriented chats. The algorithm yielded consistently good results in this broad range of topics.

The algorithm can be used without the pre-definition of the field of knowledge. Currently the algorithm is being implemented in the field of medical case studies, with the use of a field-specific dictionary, and in the field of eGovernment services, without any such pre-definition.

In the field of medical case studies, the contexts recognition algorithm is being used to extract information from actual medical cases. The goal is to examine a method for encapsulating a patient's medical history and current situation into keywords - the contexts of the medical case studies - so as to assist the physician in his analysis.

In the field of eGovernment services, the algorithm is currently being examined in TERREGOV and QUALEG, European IST projects. TERREGOV aims at providing territorial governments with flexible and interoperable tools to support the change towards eGovernment services. The purpose is to identify the contexts of documents to enable the revision of ontologies for the optimization of the indexing and search of documents. QUALEG aims at providing local governments with an effective tool for bi-directional communication with citizens. Contexts are used to specify citizen input and then provide services - routing emails to departments, opinion analysis on topics at the forefront of public debates, and identification of new topics on the public agenda.

6.2 Applicability in Multilingual Settings

The contexts recognition algorithm is also versatile in regards to the language of the input text. The algorithm enables the identification and representation of the context in multiple languages. The algorithm is not language dependent, since the Knowledge Base is extracted from the Internet. The algorithm success rate depends on the number of Internet pages existing in each language.

The Web is a multilingual corpus. Xu [25] estimated that 71% of the pages (453 million out of 634 million Web pages indexed by the Excite search engine at that time) were written in English, followed by Japanese (6.8%), German (5.1%), French (1.8%), Chinese (1.5%), Spanish (1.1%), Italian (0.9%), and Swedish (0.7%).

One hundred million words is a large enough corpus for many empirical strategies for learning about language, either for linguists [3] and lexicographers [14] or for technologies that need quantitative information about the behavior of words as input (most notably parsers [5][15]). However, for some purposes, it is not large enough.

Our initial experiments in the QUALEG project show results that coincide with the above data. The previous section displayed the consistently good results of the algorithm in English (See Table 1). Analysis of email contexts yields a high success rate of 84% in the German language as well. However, for the Polish language which has 0.42% of the web pages in the English language [13] the success rate of the algorithm is much lower and thus complementary techniques from Natural Language Processing are currently being integrated to increase effectiveness.

7 Conclusion

Every situation can be characterized by multiple contexts that describe its different aspects and that are necessary for the complete understanding of the different perspectives of the situation. The main idea of the research was to use the Internet as a contexts database to identify the multiple contexts of the given situation. The Internet is a source of information that is constantly increasing and being updated. The use of the Internet as a database for contexts recognition therefore gives a contexts recognition model immediate access to a nearly infinite amount of data in a multiplicity of fields. Hence, the necessity of creating a database for the determination of the contexts is eliminated.

Furthermore, the situations for which the contexts are sought can be independent of the Internet; the Internet is merely the database in which the algorithm searches for the contexts. Thus, for example, the contexts of a conversation between people can be found through the use of the Internet - the algorithm is a tool that allows the computer to determine the contexts by using the Internet as a database and then to pass these contexts back into the real world. The Internet is one possible source of data, but the algorithm holds also for a more restricted database. Intranet data, internally generated textual information about the organization that is stored, can also be used.

Another advantage of the contexts recognition algorithm is that it functions in real-time without needing a period of training or practice. Thus, it extracts the contexts immediately with little previous user intervention.

Tests show that the algorithm also achieves good contexts recognition results without the use of a field of knowledge dictionary, which represents specialized knowledge. Thus the algorithm can be used in diverse areas without predefined knowledge of the field.

The complexity of the algorithm is directly dependent on the size of the input description of a given situation. Thus, online implementation is feasible. Moreover, the algorithm can be implemented in an extensive variety of domains, since it is field independent. Current implementations of the algorithm focus on medical case studies and

online eGovernment applications. These online eGovernment applications show that the algorithm is also language independent and can be implemented in multilingual settings.

The Internet includes many different representations of data, such as text, image, and sound. Therefore, future directions of research include implementing the algorithm to extract contexts in alternative formats of representation. Other directions of research include mapping multiple contexts to ontologies, since contexts and ontologies are complementary disciplines of modeling views.

Acknowledgments

The work of Segev was partially supported by two European Commission 6^{th} Framework IST projects, TerreGov (http://www.terregov.eupm.net) and QUALEG (http://www.qualeg.eupm.net), and the Fund for the Promotion of Research at the Technion.

References

1. In *AAAI'99 (1999) Workshop on Reasoning in Context for AI Applications*, July 19 1999.
2. H. Assadi. Construction of a regional ontology from text and its use within a documentary system. *In Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS-98)*, 1998.
3. C. F. Baker, C. F. Fillmore, and J. B. Lowe. The berkeley framenet project. In *Proceedings of COLING-ACL*, pages 86–90, 1998.
4. T. Bauer and D. Leake. Wordsieve: A method for real-time context extraction. In *CONTEXT 2001*, pages 30–41, 2001.
5. T. Briscoe and J. Carroll. Automatic extraction of subcategorization from corpora. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997.
6. S. Dumais and H. Chen. Hierarchical classification of web content. In *Proceedings of SIGIR, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, 2000.
7. L. Erman, F. Hayes-Roth, V. Lesser, and D. R. Reddy. The Hearsay II speech understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2):213–253, 1980.
8. A. Gal, G. Modica, H.M. Jamil, and A. Eyal. Automatic ontology matching using application semantics. *AI Magazine*, 26(1), 2005.
9. N. Gerard and V. Lesser. *Blackboard Systems for Knowledge-Based Signal Understanding, Symbolic and Knowledge-Based Signal Processing*. Prentice Hall, 1992.
10. S. Gerard. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by a Computer*. Addison-Wesley Publishing Company, Inc., 1989.
11. S. Vadrevu H. Davulcu and S. Nagarajan. Ontominer: Bootstrapping and populating ontologies from domain specific websites. In *Proceedings of the First International Workshop on Semantic Web and Databases*, 2003.
12. V. Kashyap, C. Ramakrishnan, C. Thomas, and A. Sheth. Taxaminer: An experimentation framework for automated taxonomy bootstrapping. *International Journal of Web and Grid Services, Special Issue on Semantic Web and Mining Reasoning*, September 2005. to appear.
13. A. Kilgarriff and G. Grefenstette. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3), 2003.

14. A. Kilgarriff and M. Rundell. Lexical profiling software and its lexicographical applicationsa case study. In *Proceedings of EURALEX 02*, 2002.
15. A. Korhonen. Using semantically motivated estimates to help subcategorization acquisition. In *Proceedings of the Joint Conference on Empirical Methods in NLP and Very Large Corpora*, pages 216–223, 2000.
16. V. Lesser, R. Fennell, L. Erman, and D. R. Reddy. Organization of the hearsay ii speech understanding system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23:11–24, 1975.
17. A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16, 2001.
18. J. McCarthy. Generality in artificial intelligence. *Communication of ACM*, 30:1030–1035, 1987.
19. J. McCarthy. Notes on formalizing context. *In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 1993.
20. C. Papatheodorou, A. Vassiliou, and B. Simon. Discovery of ontologies for learning resources using word-based clustering. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2002)*, pages 1523–1528, 2002.
21. R. M. Turner. Model of explicit context representation and use for intelligent agents. In *1999 International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-99)*, 1999.
22. D. P. Turney. Mining the web for lexical knowledge to improve keyphrase extraction: Learning from labeled and unlabeled data. ERB-1096 NRC #44947, National Research Council, Institute for Information Technology, 2002.
23. R. E. Valdes-Perez and F. Pereira. Concise, intelligible, and approximate profiling of multiple classes. *International Journal of Human-Computer Studies*, pages 411–436, 2000.
24. T. Williams, J. Lowrance, A. Hanson, and E. Riseman. Model-building in the visions system. In *Proceedings of IJCAI-77*, 1977.
25. J. L. Xu. Multilingual search on the world wide web. In *Proceedings of the Hawaii International Conference on System Science (HICSS-33)*, 2000.