Placing Search in Context: The Concept Revisited

LEV FINKELSTEIN, EVGENIY GABRILOVICH, YOSSI MATIAS, EHUD RIVLIN, ZACH SOLAN, GADI WOLFMAN, and EYTAN RUPPIN
Zapper Technologies, Inc.

Keyword-based search engines are in widespread use today as a popular means for Web-based information retrieval. Although such systems seem deceptively simple, a considerable amount of skill is required in order to satisfy non-trivial information needs. This paper presents a new conceptual paradigm for performing search in context, that largely automates the search process, providing even non-professional users with highly relevant results. This paradigm is implemented in practice in the IntelliZap system, where search is initiated from a text query marked by the user in a document she views, and is guided by the text surrounding the marked query in that document ("the context"). The context-driven information retrieval process involves semantic keyword extraction and clustering to automatically generate new, augmented queries. The latter are submitted to a host of general and domain-specific search engines. Search results are then semantically reranked, using context. Experimental results testify that using context to guide search, effectively offers even inexperienced users an advanced search tool on the Web.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*dictionaries*, *linguistic processing*, *thesauruses*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering*, *query formulation*, *search process*; I.7.5 [**Document and Text Processing**]: Document Capture—*document analysis*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Search, context, semantic processing, invisible web, statistical natural language processing

1. INTRODUCTION

Given the constantly increasing information overflow of the digital age, the importance of information retrieval has become critical. Web search is one of the most challenging problems of the Internet today, striving to provide users with search results most relevant to their information needs. Internet search engines have evolved through several generations since their inception in 1994, progressing from simple keyword matching to techniques such as link analysis and relevance feedback (achieved through refinement questions or accumulated

This article is a revision of a paper that was presented at the Tenth International World Wide Web Conference (WWW10), Hong Kong, May 2001.
Authors' address: Zapper Technologies, Inc., 3 Azrieli Center, Tel Aviv 67023, Israel; email: gabr@acm.org; {lev, yossi, ehud, zach, gadi, eytan}@zapper.com. E. Gabrilovich Current affiliation: Department of Computer Science, Technion-Israel Institute of Technology, Haifa 32000, Israel; email: gabr@acm.org.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.
© 2002 ACM 1046-8188/02/0100-0116 $5.00

ACM Transactions on Information Systems, Vol. 20, No. 1, January 2002, Pages 116–131.

Petitioner has added number 116 to the first page. Otherwise, it leaves the original page numbering.

MICROSOFT EXHIBIT 1028

personalization information) [Sherman 2000a]. Search engines have now entered their third generation, and current research efforts continue to be aimed at increasing coverage and relevance.

A large number of recently proposed search enhancement tools have utilized the notion of *context*, making it one of the most abused terms in the field, referring to a diverse range of ideas from domain-specific search engines to personalization. We present here a novel search approach that interprets context in its most natural setting, namely, a body of words surrounding a user-selected phrase. We anticipate the growing number of searches that originate while users are reading documents[1] on their computers, and require further information about a particular word or phrase [Microsoft 2001]. Hence, the basic premise underlying our approach is that searches should be processed *in the context of the information surrounding them*, allowing more accurate search results that better reflect the user's actual intensions. For example, a search for the word "Jaguar" should return car-related information if performed from a document on the motoring industry, and should return animal-related information if performed from an Internet website about endangered wildlife. Guiding user's search by the context surrounding the text eliminates possible semantic ambiguity and vagueness.

Our system (named IntelliZap) is based on the client-server paradigm, where a client application running on the user's computer captures the context around the text highlighted by the user. The server-based algorithms analyze the context, selecting the most important words (implicitly performing word sense disambiguation), and then prepare a set of augmented queries for subsequent search. The technology also enables the user to modify the extent to which context guides any given search, by modifying the amount of context considered. Queries resulting from context analysis are dispatched to a number of search engines, performing meta-search. When the context can be reliably classified to a predefined set of domains (such as health, sport or finance), additional queries are dispatched to search engines specializing in these domains. This step can be viewed as referring to the *Invisible Web*, as some of the target domain-specific engines may constitute front-ends to databases that are not otherwise indexed by conventional search engines. A dedicated *reranking* module ultimately reorders the results received from all of the engines, according to semantic proximity between their summaries and the original context. To this end we use a semantic metric that, given a pair of words or phrases, returns a (normalized) score reflecting the degree to which their meanings are related. In fact, IntelliZap performs as an information specialist acting on behalf of the user, which automatically performs the search steps, from query expansion, to search engine selection, to reranking the results.

The significance of the new *context-based* approach lies in the improved relevance of search results even for users not skilled in Web search. We achieve this by applying natural language processing techniques to the captured context in order to guide the subsequent search for user-selected text. Existing approaches

[1]Such documents can be in a variety of formats (MS Word DOC, HTML or plain text to name but a few), and either online (residing on the Internet) or offline (residing on a local machine).

either analyze the entire document the user is working on, or ask the user to supply a category restriction along with search keywords. As opposed to these, our proposed method automatically analyzes the context in the immediate vicinity of the focus text. This allows analyzing just the right amount of background information, without running over the more distant (and less related) topics in the source document. The method also allows collecting contextual information without conducting an explicit dialog with the user.

This paper is organized as follows. The next section reviews related work. Section 3 presents the various features of our context-based search system, explaining how several individual algorithms work in concert to improve the relevance of the search. Section 4 discusses the experimental results. Finally, Section 5 concludes the paper and suggests further research directions.

2. RELATED WORK

Using context for search is not a new idea. A number of existing information retrieval systems utilize the notion of context to some extent. The problem is, however, that everyone defines *context* a little differently. This section surveys a number of approaches to using context in Web search, and is based in part on the elaborate review on the topic by Lawrence [2000].

Explicit context information can be supplied to a search engine in the form of a category restriction.[2] Such a category may considerably disambiguate a query and thus focus the results. For instance, given the search term "jaguar", possible categories are "fauna" or "cars". The Inquirus-2 project [Glover et al. 1999] specifically requests context information in this way.

In contrast to this approach, other tools infer context information automatically by analyzing whole documents displayed on users' screens. The Watson project [Budzik and Hammond 2000] attaches this background information to explicit user queries, while tools like Kenjin[3] automatically suggest Web sites related[4] to the document being worked upon. Such tools encounter difficulties when documents are long and discuss a variety of topics—as the data collected from the entire document reflects all the topics covered, it might not be particularly relevant to the user's current focus (be it an explicit query in the former case, or simply the active part of the document in the latter). The main difference between such tools and IntelliZap is that the latter analyzes the context *in the immediate vicinity* of the user-selected text, thus making the context coherent and focused around a single topic. At the other end of the spectrum, tools like GuruNet (now Atomica)[5] perform database lookup directly from reference sources (dictionaries, encyclopedias etc.). Such tools offer only a limited usage of text, without deep semantic analysis of the enclosing context.

Another interesting document-oriented approach, catering to users' needs to follow up on words or phrases while reading documents, is implemented in

[2]The target engine must obviously support a mechanism for search restriction, so that a category constitutes an integral part of the query.
[3]www.kenjin.com
[4]Note that Kenjin provides related links as opposed to performing conventional search.
[5]www.atomica.com

the Smart Tags mechanism incorporated in Microsoft's new Office XP [2001]. This mechanism dynamically recognizes known terms in documents, labeling them with contextual information. Users can then take relevant actions on the recognized terms, such as navigating to a Web site or looking up a stock symbol, with overall productivity improvement across applications.

There is a family of tools that interprets the notion of context as a set of previous information requests originated by the user. Defined in this way, context search becomes *personalization*, and tools in this category keep track of user's previous queries and/or documents viewed. SearchPad [Bharat 2000] recognizes that many advanced users perform several searches concurrently, and tracks search progress over time. This extension to search engines keeps track of "search context" by following the different search sessions and collecting "useful queries and promising results links" [Bharat 2000].

Xu and Croft [2000] suggested a new query expansion technique based on *local context analysis*. This technique analyzes the concepts found in the top-ranked documents initially retrieved for a given query, and then adds the best scoring concepts to the query. In other words, the query is expanded *in the context* of top-ranked documents retrieved in the first step.

Other ways of incorporating context into search include the usage of domain-specific rather than general-purpose engines [Lawrence 2000]. Databases which belong to the Invisible Web (i.e., whose contents are not indexed by conventional search engines) may be particularly useful as they might contain vast amounts of information within their narrow domain. IntelliZap pursues a similar approach by classifying the topic of the query context, and targeting search engines specializing in the corresponding domain. Note that in this way, the selection of specialized search engines is performed automatically.

Yet another interpretation of *context* belongs to the realm of *link analysis* [Sherman 2000b; Sullivan 2000]. In the quest to expand their coverage, some engines intentionally limit the number of sites they index to make the retrieval efficient, although they can still yield "unindexed" sites in search results. This is achieved by analyzing the *context of links* pointing at these sites, thus deducing information about the contents of the target. Google and Inktomi,[6] among others, employ this technique. Another context-related feature of Google shows up in its search-dependent result summaries. A typical Google summary contains an excerpt from the Web page where the search terms are shown highlighted *in the context of this page* [Google].

Our approach focuses on using the context in its most natural sense—that of the text surrounding the marked query. The limitation of this approach is that it assumes the query is triggered by the need for more information on a term in an existing document. When this is the case, it provides local semantic consistency for the interpretation of this term (i.e., the user-marked query) and yields superior results.

To the best of our knowledge, the GuruNet and Kenjin programs described above are the products most similar to IntelliZap, although they only use context to a limited degree. Since there are no well established benchmarks for

[6]www.google.com and www.inktomi.com, respectively.

evaluating the performance of such tools, and because it is difficult to correlate the functionality of the related links with search *per se* (see also footnote 4), we present, in Section 4, a comparison of IntelliZap to major general-purpose search engines.

3. THE CONTEXT-BASED SEARCH SYSTEM

Current approaches to information retrieval over the Web are based on a scenario in which the user enters a query to a search engine. The search engine then retrieves an ordered set of documents that best match the user's query. We propose an approach that changes the basic settings of the search scene by using the context of the query as an additional input. In this scenario, when the user marks text in a document and submits it for search, the system captures the context surrounding the text, and utilizes it to yield more focused results. The context may include the sentence containing the query word or phrase, a few sentences surrounding the query term, the paragraph in which it resides, or even the whole document.

Using the context to guide the search constitutes a considerable algorithmic challenge. One needs to find ways to extract the amount of context which optimizes the information retrieved, as well as to devise adequate ways to use the extracted context for focusing the response to the user's query.

3.1 System Overview

We have developed a system called IntelliZap[7] that performs context search from documents on users' computers. When viewing a document, the user marks a word or phrase (referred to as *text)* to be submitted to the IntelliZap service (in the example of Figure 1, the marked text is the word "jaguar"). The client application automatically captures the *context* surrounding the marked text, and submits both the text and the context to server-based processing algorithms.

Figure 1 shows a screen shot with the software client invoked on a user document, and Figure 2 shows a part of the results page. Observe that the top part of the results page repeats the user-selected text *in the original context* (only part of which is displayed, as the actually captured context may be quite large).

The IntelliZap system has four main components:

1. Context capturing (performed by client-side software).
2. Extracting keywords from the captured text and context.
3. High-level classification of the query to a small set of predefined domains.
4. Reranking the results obtained from different search engines.

The three latter components are based on the semantic network explained in the next section. Figure 3 gives a schematic overview of the IntelliZap system, while the following sections explain its individual components.

[7]The IntelliZap client application may be obtained from www.zapper.com. The Web site also features a Web-based IntelliZap, which does not require client download, but rather allows the user to copy-and-paste both search terms and context into appropriate fields of an HTML form. The latter feature is available at http://www.intellizap.com.

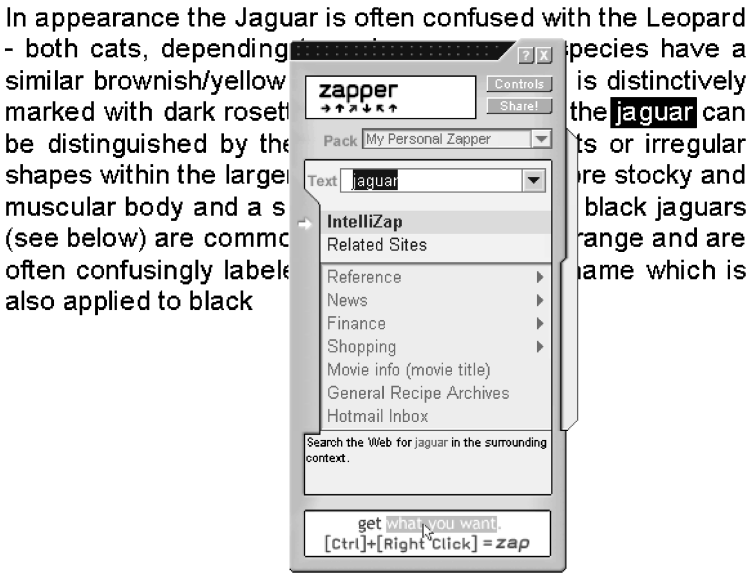

Fig. 1. IntelliZap client invocation on a document.

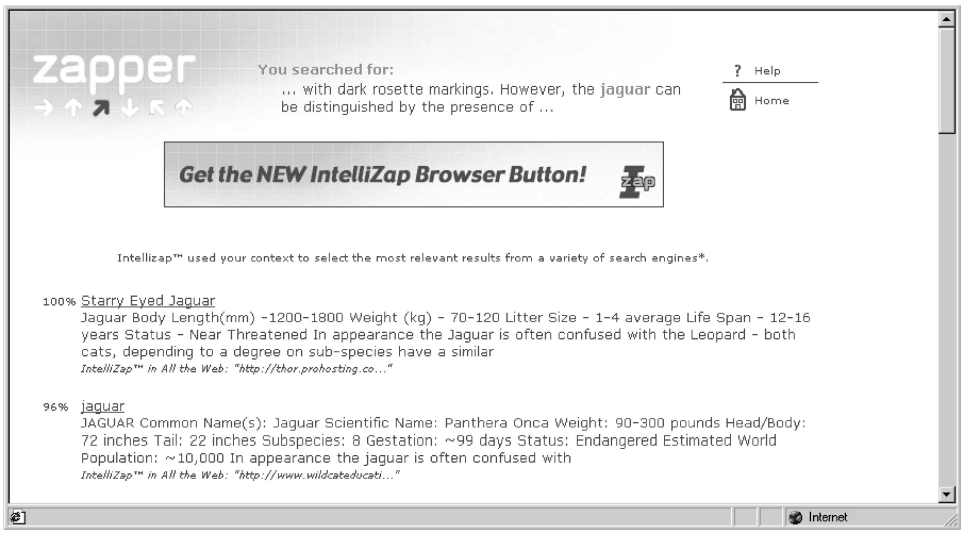

Fig. 2. IntelliZap search results.

3.2 The Core Semantic Network

The core of IntelliZap technology is a semantic network, which provides a metric for measuring distances between pairs of words. The basic semantic network is implemented using a vector-based approach, where each word is represented as a vector in multi-dimensional space. To assign each word a vector representation, we first identified 27 knowledge domains (such as computers, business and entertainment) roughly partitioning the whole variety of topics.

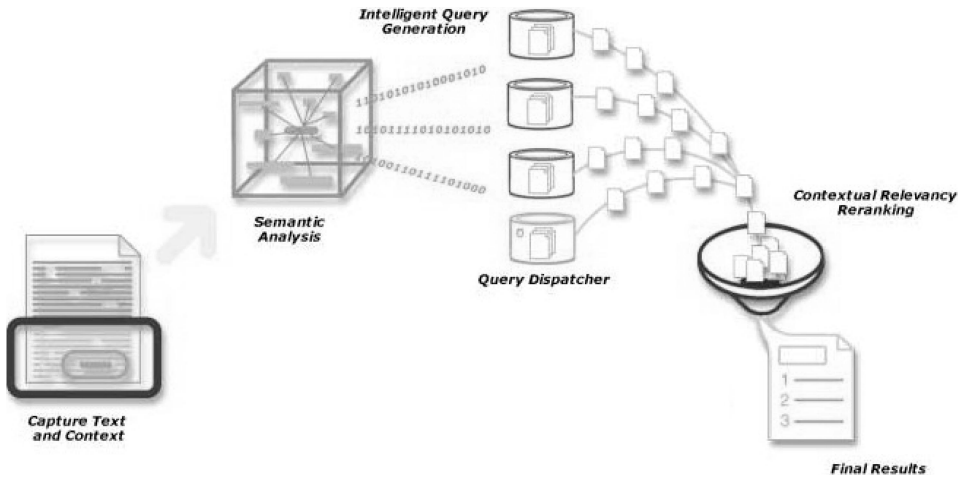

Fig. 3. IntelliZap system overview: information and processing flow (from left to right)

We then sampled a large set of documents in these domains on the Internet.[8] Word vectors[9] were obtained by recording the frequencies of each word in each knowledge domain. Each domain can therefore be viewed as an axis in the multi-dimensional space. The distance measure between word vectors is computed using a correlation-based metric:

$$sim_{VB}(w_1, w_2) = \sum \frac{(\vec{w}_1 - \bar{w}_1)(\vec{w}_2 - \bar{w}_2)}{\sigma_1 \sigma_2},$$

where $\vec{w}_1$ and $\vec{w}_2$ are vectors corresponding to words w_1 and w_2, and $\bar{w}_i$ and σ_i are estimates of their mean and standard deviation, respectively. Although such a metric does not possess all the distance properties (observe that the triangle inequality does not hold), it has strong intuitive grounds—if two words are used in different domains in a similar way, these words are most probably semantically related.

We further enhance the statistically based semantic network described above, using linguistic information, available through the WordNet electronic dictionary [Fellbaum 1998]. Since some relations between words (like hypernymy/hyponymy and meronymy/holonymy) cannot be captured using purely statistical data, we use WordNet dictionary to correct the correlation metric. A WordNet-based metric was developed using an information content criterion similar to Resnik [1999], and the final metric was chosen as a linear combination between the vector-based correlation metric and the WordNet-based metric:

$$sim(w_1, w_2) = \alpha \cdot sim_{VB}(w_1, w_2) + \beta \cdot sim_{WN}(w_1, w_2),$$

where $sim_{VB}(\cdot, \cdot)$ and $sim_{WN}(\cdot, \cdot)$ are the vector-based and the WordNet-based metrics, respectively. Optimal values for α and β were obtained from the training set of word pairs (see below), and verified using a cross-validation technique.

[8]Approximately 10,000 documents have been sampled in each domain.
[9]Each word vector has 27 dimensions, as the number of different domains.

Unfortunately, there are no accepted procedures for evaluating performance of semantic metrics. Following Resnik [1999], we evaluated different metrics by computing correlation between their scores and human-assigned scores for a list of word pairs. The intuition behind this approach is that a good metric should approximate human judgments well. While Resnik used a list of 30 noun pairs from Miller and Charles [1991], we opted for a more comprehensive evaluation. To this end, we prepared a diverse list of 350 noun pairs representing various degrees of similarity,[10] and employed 16 subjects to estimate the "relatedness" of the words in pairs on a scale from 0 (totally unrelated words) to 10 (very much related or identical words). The vector-based metric achieved 41% correlation with averaged human scores, and the WordNet-based metric achieved 39% correlation[11,12] A linear combination of the two metrics achieved 55% correlation with human scores.

Currently, our semantic network is defined for the English language, though the technology can be adapted for other languages with minimal effort. This would require training the network using textual data for the desired language, properly partitioned into domains. Linguistic information can be added, subject to the availability of adequate tools for the target language (e.g., EuroWordNet for European languages [Euro WordNet] or EDR for Japanese [Yokoi 1995]).

3.3 Keyword Extraction Algorithm

The algorithm utilizes the semantic network to extract keywords from the context surrounding the user-selected text. These keywords are added to the text to form an augmented query, leading to context-guided information retrieval.

The algorithm for keyword extraction belongs to a family of clustering algorithms. However, a straightforward application of such algorithms (e.g., K-means [Duda and Hart 1973; Fukunaga 1990]) is not feasible due to a large amount of noise and a small amount of information available—usually we have about 50 context words represented in 27-dimensional space, which makes the clustering problem very difficult. Observe also, that application of a clustering algorithm would require that the semantic network be able to handle non-words

[10]Our list included, among others, all the 30 noun pairs from Miller and Charles [1991]. The correlation between our subjects' scores and those reported by Miller and Charles is consistently high—95%.

[11]Resnik [1999] reports 79% correlation with humans for the metric that uses the information content criterion. Although we replicated this result for Miller and Charles' word list [Miller and Charles 1991] with a high degree of confidence (obtaining 75% correlation with human scores), for a longer list of 350 word pairs the WordNet metric only achieves 39% correlation. For the sake of comparison, a metric based on latent semantic analysis (LSA) [Landauer et al. 1998] achieves 56% correlation for this longer list (in this experiment we used the implementation of LSA available online at http://lsa.colorado.edu).

[12]Our list of 350 word pairs contained 82 in which at least one word was not found in WordNet. When these pairs are disregarded, the correlation between the WordNet-based metric and humans rises to 47%. We can also observe here the synergy between the two components of the semantic metric—while WordNet reflects word relations that cannot be captured statistically, the vector-based component handles statistical word cooccurrence and contains words not found in the electronic dictionary.

(centroids of multi-dimensional clusters), and this requirement is problematic for the WordNet-based metric. In order to overcome these problems we used a special-purpose clustering algorithm (similar to Opher et al. [1999]) that performs recurrent clustering analysis and then refines the results statistically. To this end, we first perform 100 iterations of the K-means algorithm, and build an adjacency matrix A, so that $A(i, j)$ contains the number of iterations when words i and j were assigned to the same cluster. During this stage, only the vector-based semantic metric is used, as it can easily represent any vector, not necessarily corresponding to an existing word. We then modify the values of A according to the distances between words estimated by the WordNet-based metric. Specifically, we increase the value $A(i, j)$ if the combined semantic metric considers words i and j more related than the vector-based metric alone (this effectively reflects the similarity score produced by the WordNet-based metric), and decrease it otherwise. Finally, we reconstruct word clusters from the resultant matrix by identifying *strongly-connected components*, that is, groups of words for which the value of $A(i, j)$ is above some empirically estimated threshold value (pairwise).

For a typical query of 50 words (one to three words in the text, and the rest in the context), the keyword extraction algorithm usually returns three or four clusters. The rationale of the clustering process is to identify clusters of words that represent different semantic aspects of the query. Keywords in clusters are ordered by their semantic distance from the text, so that the most important keywords appear first. Cluster-specific queries are then built by combining the text words with several of the most important keywords of each cluster. Responding to such queries, search engines yield results covering most of the semantic aspects of the original context, while the reranking algorithm filters out irrelevant results.

3.4 Search Engine Selection

The queries created as explained above are dispatched to a number of general-purpose search engines. In addition, the system classifies the captured context in order to select domain-specific engines that stand a good chance for providing more specialized results. The classification algorithm based on probabilistic analysis classifies the context to a limited number of high-level domains[13] (e.g., medicine or law) by determining the amount of similarity between predefined domain "signatures" and the query context. In order to compute the domain signature, a corpus of approximately 100,000 words is sampled for each domain. As in the semantic metric (see Section 3.2), each word is represented by a vector that reflects its occurrence frequencies across the domains. The probability of a domain given a particular text query, $P(Domain_j \mid Text)$, can be represented according to Bayes' rule as follows:

$$P(Domain_j \mid Text) = \frac{P(Text \mid Domain_j)P(Domain_j)}{P(Text)}$$

[13]Currently, 9 of the 27 domains used in the semantic metric are employed for search engine selection. The *a priori* assignment of search engines to domains is performed offline, while each domain is mapped to two or three search engines.

The probability $P(Text)$ is constant, and we assume the prior probabilities of domains to be equal; therefore, only $P(Text \mid Domain_j)$ needs to be computed. The probability of the text query given a domain, $P(Text \mid Domain_j)$, is modeled as the product of probabilities[14] of all the text words w_i given this domain:

$$P(Text \mid Domain_j) = \prod_{w_i \in Text} P(w_i \mid Domain_j).$$

We ultimately select the search engines which correspond to the domain j that maximizes the value of $P(Domain_j \mid Text)$.

Some of the search engines (such as AltaVista[15]) allow limiting the search to a specific category. In such cases, categorizing the query in order to further constrain the search usually yields superior results.

3.5 Reranking

After queries are sent to the targeted search engines, a relatively long list of results is obtained. Each search engine orders the results using its proprietary ranking algorithm, which can be based on word frequency (inverse document frequency), link analysis, popularity data, priority listing, and so forth. Therefore, it is necessary to devise an algorithm that would allow us to combine the results of different engines and put the most relevant ones first.

At first, this problem may seem misleadingly simple—after all, humans usually select relevant links by quickly scanning the list of results summaries. Automating such an analysis can, however, be very demanding. To this end, we again make use of the semantic network, in order to estimate the relatedness of search results to the query context.

The reranking algorithm reorders the merged list of results by comparing them semantically with both the text query and the context surrounding it. The algorithm computes semantic distances between the words of text and context on the one hand, and the words of results' titles and summaries on the other hand. Text, context, titles, and summaries are treated as sets (bags) of words. The (asymmetric) distance between a pair of such sets is canonically defined as an average distance from the words of the first set to the contents of the second set:

$$dist(S_1, S_2) = \frac{1}{|S_1|} \sum_{w \in s_1} dist(w, S_2),$$

where the distance between a word and a set of words is defined as the shortest distance between this word and the set (i.e., the distance to the nearest word of the set):

$$dist(w, S) = \min_{w' \in s} dist(w, w').$$

The distance measure used in these computations is exactly the semantic metric defined in Section 3.2 above. The final ranking score is given by weighting

[14]In order to prevent computation underflow, we actually use a sum of probability logarithms rather than a product of raw probability values.

[15]www.altavista.com

the distances between text and summary, context and summary, summary and text, and summary and context. Search results are sorted in decreasing order of their scores, and the newly built results list is displayed to the user.

An important feature of the algorithm is that the distances computed between sets of words are *not* symmetric—specifically, the distances from text and context to summaries are taken with larger weights than their reciprocals. Observe that the text (and, incidentally, the context) is selected by the user, while the summaries are somewhat more arbitrary in their nature. According to the above formulae, computing the distance from the text to a summary considers all the text words, but not necessarily all the summary words. Thus, giving extra weight to distances from text and context to summaries effectively realizes the higher importance of text and context words.

4. EXPERIMENTAL RESULTS

In this section we discuss a series of experiments conducted on the IntelliZap system. The results achieved allow us to claim that using the context effectively provides even inexperienced users with advanced abilities for searching the Web.

4.1 Context vs. Keywords: A Quantitative Measure

A survey conducted by the NEC Research Institute shows that about 70% of Web users typically use only a single keyword or search term [Butler 2000]. The survey further shows that even among the staff of the NEC Research Institute itself, about 50% of users use one keyword, an additional 30%—two keywords, about 15%—three keywords, while only 5% of users actually use four keywords or more. The goal of the experiment described below was to determine what number of keywords in a conventional search scenario with a keyword-based search engine is equivalent to using the context with the IntelliZap system.

Twenty-two subjects recruited by an external agency participated in this study. Conditions for participation included college-level acquaintance with the Internet and high level command of English. Other than that, the subjects had no explicit demographic biases, and comprised a fairly unbiased sample of Israeli population versed in Internet search. Each subject was presented with three short texts and was asked to find (in three separate stages of the test) information relevant to the text using IntelliZap and each of the following search engines: Google, Yahoo, AltaVista, and Northern Light.[16]

The texts were composed of a number of short paragraphs (about four to seven lines long), each focused on a specific topic selected from the Encarta Encyclopedia. The subjects were told that the study compares the utility of a variety of engines, and had no prior knowledge of the topics discussed in the texts. At no point were they informed that the comparison between IntelliZap specifically, and the other engines, was the focus of the study. The subjects were asked to search for relevant information using one, two and three keywords

[16]www.google.com, www.yahoo.com, www.altavista.com, and www.northernlight.com, respectively.

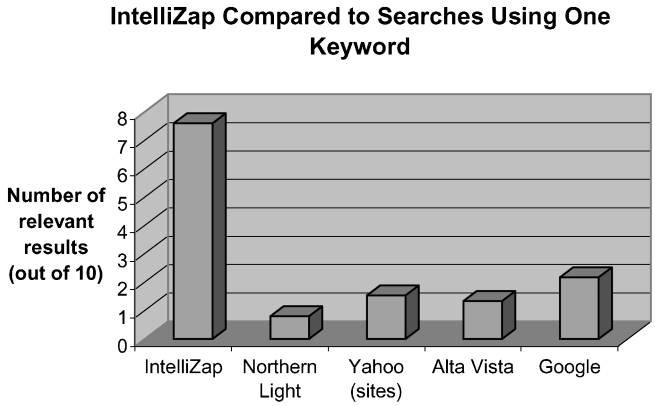

Fig. 4. IntelliZap compared to searches using one keyword.
Statistical significance of the difference between IntelliZap and Google:
$\chi^2 - \mathrm{p} = 0.004$, K-S $- \mathrm{p} = 2 \cdot 10^{-7}$.

using each of the search engines. They were not limited to the keywords used in the source texts and could come up with any keyword they saw fit. Moreover, they were free to use any search operators they wished, and did not receive any explicit guidance in this regard. The instructions for using IntelliZap remained the same through all stages—to capture any word or phrase from the text, as the subjects deemed appropriate.

Relevancy[17] was rated for the first ten results returned. The rating system was defined as follows: 0 for irrelevant results, 0.5 for results relevant only to the general subject of the text, and 1 for results relevant to the specific subject of the text. Dead links and results in languages other than English were assigned the score of 0. The cumulative score for each search was defined as the sum of individual scores for the first ten results. Figures 4, 5, and 6 show the results for one, two, and three keyword queries, respectively. The non-monotonic behavior of the number of relevant results among the stages is due to the usage of different texts in different stages of the experiment.

When the search engines are probed with a single keyword (Figure 4), the superiority of IntelliZap is very distinct. In order to verify the statistical significance of this difference, we used two tests: chi-square (χ^2) and Kolmogorov-Smirnov (K-S) [Press et al. 1992]. When IntelliZap is compared to the closest engine—Google—the p-value computed according to χ^2 is $\mathrm{p} = 0.004$, and according to K-S $- \mathrm{p} = 2 \cdot 10^{-7}$. As follows from Figures 5 and 6, using context efficiently enables IntelliZap to outperform other engines even when the latter are probed with two- and three-keyword queries, although in these cases the difference is not statistically significant.

[17]The notion of *relevancy* was obviously subjectively interpreted by each tester. Here we report the cumulative results for all the participants of the experiment.

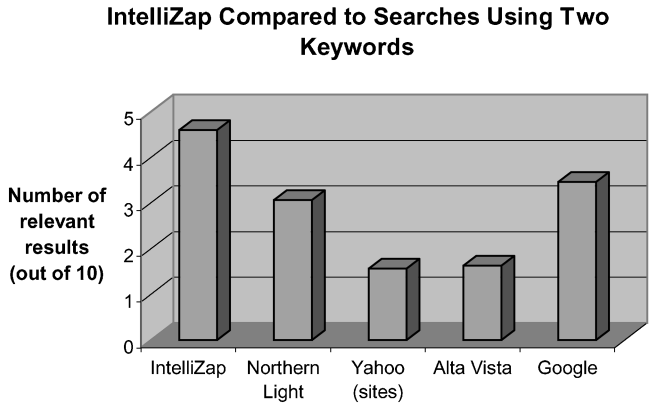

Fig. 5. IntelliZap compared to searches using two keywords.

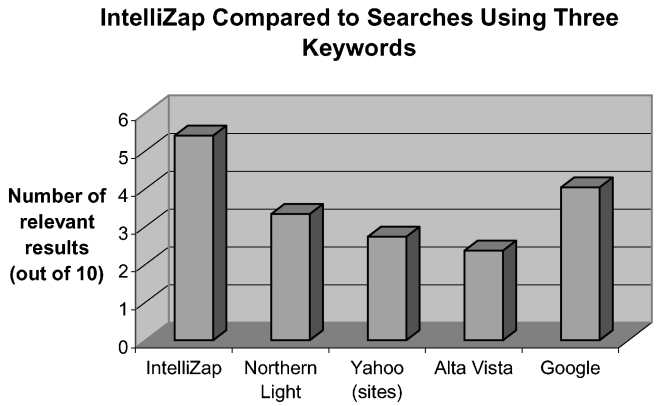

Fig. 6. IntelliZap compared to searches using three keywords.

4.2 IntelliZap vs. Other Search Engines: An Unconstrained Example

In order to validate the IntelliZap performance, we compared it with a number of major search engines: Google, Excite, AltaVista, and Northern Light.[18] Twelve subjects recruited by an external agency were tested. As before, the subjects were required to have some acquaintance with the Internet and high level command of English. At no point throughout the study were the subjects explicitly informed that the comparison between IntelliZap specifically, and the other engines, was the focus of the study.

Each subject was presented with five randomly selected short texts. For each text the subject was asked to conduct *one* search in order to find information relevant to the text using a randomly assigned search engine. The subjects were given no instructions or limitations regarding how to search. This is because the aim of this part of the test was to compare IntelliZap to other search engines when users employed their natural search strategies. In particular, the

[18]www.google.com, www.excite.com, www.altavista.com, www.northernlight.com, respectively.

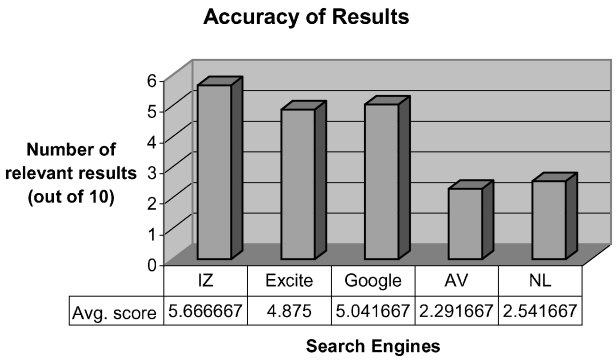

Fig. 7. IntelliZap vs. other search engines—accuracy of results.

users were allowed to use boolean operators and other advanced search features as they saw fit. The IntelliZap system used in this experiment utilized Google, Excite, Infoseek[19] (currently GO network search) and Raging Search[20] as underlying general-purpose engines. A number of domain-specific search engines (such as WebMD and FindLaw[21]) were also used in cases when the high-level classification succeeded in classifying the domain of the query. The subjects were required to estimate the quality of search by counting the number of relevant links in the first ten results returned by each engine. The relevancy rating system was identical to the one described in the previous experiment.

As can be seen from the comparison chart in Figure 7, IntelliZap achieves a level of performance comparable to major search engines, but does so without any human guidance (apart from marking the text to search for, in order to commence the process). Note that the above test measures only the precision of search, as it is very difficult to measure the recall rate when operating Web search engines. However, the precision rate appears to be highly correlated with user satisfaction from the search results.

4.3 Response Time

In the client-server architecture of IntelliZap, client-captured text and context are sent to the server for processing. Server-side processing includes query preparation based on context analysis, query dispatch, merging of search results, and delivering the top reranked results to the user. The cumulative server-side processing time per user query is less than 200 milliseconds, measured on a Pentium III 600 MHz processor. In contrast to the conventional scenario, in which users access search engines directly, our scheme involves two connection links, namely, between the user's computer and the server, and between the server and search engines (that are contacted in parallel). Therefore, actual

[19] www.go.com
[20] www.raging.com
[21] www.webmd.com and www.findlaw.com, respectively.

response time of IntelliZap depends on the slowest search engine employed. Thanks to the high-speed Internet connection of the server, the proposed scheme delivers the results to the end user in less than 10 seconds.

This response time is considerably slower than that of the conventional search engines, because of the overhead involved in metasearching. Observe, however, that the time elapsed from when the query is submitted until the results are available, is only a small fraction of the overall time users spend in the search process. In fact, users spend most of the time to formulate a good query and to analyze the search results, while the former task is performed by the IntelliZap system semi-automatically in an almost instantaneous manner.

5. DISCUSSION

This paper describes a novel algorithm and system for processing queries in their context. Our approach caters to the growing need of users to search directly from items of interest they encounter in the documents they view.[22] Using the context surrounding the marked queries, the system enables even inexperienced web searchers to obtain satisfactory results. This is done by automatically generating augmented queries and selecting pertinent search engine sites to which the queries are targeted. The experimental results we have presented testify to the significant potential of the approach.

This work can be extended in the future in a number of ways. First, context can be utilized to expand the augmented queries in a disambiguated manner to include new terms. This disambiguation process could be used to concomitantly determine the extent of the context which is most relevant for processing the specific query in hand. Second, more work could be done on specifically tailoring the generic approach shown here, for maximizing the context-guided capabilities of individual search engines. In summary, harnessing context to guide search from documents offers a new and promising way to focus information retrieval and counteract the "flood of information" so characteristic of the World Wide Web.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewer for the comments and suggestions that greatly improved this paper.

REFERENCES

BHARAT, K. 2000. SearchPad: Explicit capture of search context to support web search. In *Proceedings of the 9th International World Wide Web Conference, WWW9* (Amsterdam, May).

BUDZIK, J. AND HAMMOND, K. J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces* (New Orleans, Louisiana) ACM Press, New York, NY 44–51.

[22]This particularly applies to users who are professionals in their respective fields. For example, software developed by LexisNexis, one of the early adopters of Microsoft's Smart Tags technology (see Section 2 above), allows legal professionals to look up various terms found in documents, to locate public records, news, and other legal information [Lexis Nexis 2001; Microsoft Office XP 2001].

BUTLER, D. 2000. Souped-up search engines. *Nature,* Vol. 405 (May), 112–115.

DUDA, R. O. AND HART, P. E. 1973. *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York, NY.

EUROWORDNET. http://www.hum.uva.nl/~ewn/

FELLBAUM, C., ED. 1998. WordNet—An Electronic Lexical Database. MIT Press, The WordNet database is available online at http://www.cogsci.princeton.edu/~wn.

FUKUNAGA, K. 1990. *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, CA.

GLOVER, E. ET AL. 1999. Architecture of a meta search engine that supports user information needs. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, *CIKM 99*, (Kansas City, MO, November) 210–216.

GOOGLE. The basics of Google search. http://www.google.com/help/basics.html

LANDAUER, T. K., FOLTZ, P. W., AND LAHAM, D. 1998. Introduction to Latent Semantic Analysis. *Dis. Proc. 25,* 2 & 3, 259–284.

LAWRENCE, S. 2000. Context in web search. *Data Engineering*, IEEE Comput. Soc, *23*, 3, (September) 25–32. http://www.research.microsoft.com/research/db/debull/A00sept/lawrence.ps

LEXISNEXIS. 2001. LexisNexis delivers Smart Tags for Microsoft Office XP http://www.lexisnexis.com/lncc/about/newsreleases/0412.html.

MICROSOFT OFFICE XP. 2001. At a glance: Smart Tags http://www.microsoft.com/Partner/BusinessDevelopment/SalesResources/factsheets/SmartTags.asp.

MILLER, G. A., AND CHARLES, W. G. 1991. Contextual correlates of semantic similarity. *Lang. Cog. Pro. 6*, 1, 1–28.

OPHER, I., HORN, D., AND QUENET, B. 1999. Clustering with Spiking Neurons. In *Proceedings of the International Conference on Artificial Neural Networks* (*ICANN' 99*, Edinburgh, Scotland, September) 485–490.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1992. *Numerical Recipes in C*: *The Art of Scientific Computing*, 2nd ed. Cambridge University Press, Section 14.3, 620–628.

RESNIK, P. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. J. Art. Intell. *Re.*, *11*, 95–130.

SHERMAN, C. 2000a. Inktomi inside. http://websearch.about.com/internet/websearch/library/weekly/aa041900a.htm.

SHERMAN, C. 2000b. Link building strategies. http://websearch.about.com/internet/websearch/library/weekly/aa082300a.htm.

SULLIVAN, D. 2000. Numbers, numbers—but what do they mean? *The Search Engine Report* (March 3). http://searchenginewatch.com/sereport/00/03-numbers.html.

XU, J. AND CROFT, W. B. 2000. Improving the effectiveness of information retrieval with local context analysis. *ACM TOIS 18*, 1 (January) 79–112.

YOKOI, T. 1995. The EDR electronic dictionary. *Comm. ACM, 38*, 11 (November) ACM Press, New York, NY, 42–44.

Received April 2001; revised August 2001 and September 2001; accepted September 2001