# The Infocious Web Search Engine: Improving Web Searching Through Linguistic Analysis

Alexandros Ntoulas
Infocious Inc.
ntoulas@infocious.com

Gerald Chao
Infocious Inc.
gerald@infocious.com

Junghoo Cho
UCLA Computer Science
cho@cs.ucla.edu

## ABSTRACT

In this paper we present the Infocious Web search engine [23]. Our goal in creating Infocious is to improve the way people find information on the Web by resolving ambiguities present in natural language text. This is achieved by performing linguistic analysis on the content of the Web pages we index, which is a departure from existing Web search engines that return results mainly based on keyword matching. This additional step of linguistic processing gives Infocious two main advantages. First, Infocious gains a *deeper understanding* of the content of Web pages so it can better match users' queries with indexed documents and therefore can improve relevancy of the returned results. Second, based on its linguistic processing, Infocious can *organize* and *present* the results to the user in more intuitive ways. In this paper we present the linguistic processing technologies that we incorporated in Infocious and how they are applied in helping users find information on the Web more efficiently. We discuss the various components in the architecture of Infocious and how each of them benefits from the added linguistic processing. Finally, we experimentally evaluate the performance of a component which leverages linguistic information in order to categorize Web pages.

## Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Content Analysis and Indexing;
H.3.3 [**Information Systems**]: Information Search and Retrieval;
H.3.7 [**Information Systems**]: Digital Libraries;
C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems

## Keywords

Web search engine, Web searching, information retrieval, crawling, indexing, language analysis, linguistic analysis of Web text, natural language processing, part-of-speech tagging, word sense disambiguation, phrase identification, concept extraction.

## 1. INTRODUCTION

Millions of users today use Web search engines as the primary (and sometimes the sole) means for locating information. They rely on search engines to satisfy a broad variety of information needs, ranging from researching medical conditions to locating a convenience store to comparing available products and services. The most popular of the search engines today (e.g. Google [21], Yahoo! [46], MSNSearch [34], AskJeeves [3], etc.) maintain a fresh local repository of the ever-increasing Web. Once a user issues a query, search engines go through their enormous repository and identify the most relevant documents to the user's query.

In all of the current major Web search engines, the process of identifying the relevant documents typically involves matching the keywords present in the user's query with the documents in the local repository. That is, for a Web page to be considered relevant to a query it simply has to contain the query keywords. Until now this particular approach has worked very well in serving most of the users' needs. However, there are queries for which simple keyword matching will not suffice.

For example, consider the single-keyword query *jaguar* that a user might issue to a search engine. As of the time of this writing, the major search engines return results that deal with at least three disjoint issues: (1) Jaguar - the car brand name, (2) Jaguar - the latest version of MacOS X, (3) jaguar - the animal. As one can imagine, it is highly unlikely that a user is interested in all three of the above at the same time. Therefore, the query *jaguar* is an example of an ambiguous query because it is associated with multiple senses, each one pertaining to a different topic of interest. As a consequence, Web pages that have distinct topics but all share the same keywords are considered relevant and presented to the user. Resolving such ambiguities has long been the study of a field called Natural Language Processing (NLP), which we will briefly review in the next section.

In this paper we present Infocious, a new Web search engine built with the primary goal of improving the users' search experience by reducing ambiguities through linguistic analysis. Infocious applies linguistic analysis in two major ways:

1. First, through language analysis Infocious resolves ambiguities within the content of Web pages. Currently Infocious focuses on three types of ambiguity: (1) part-of-speech ambiguity, (2) phrasal ambiguity, and (3) topical ambiguity. As more ambiguities are resolved, Infocious provides for more precise searching and enables the users to locate the information they seek more quickly and more accurately.

2. Second, the language analysis done contributes to the ranking of the results that are presented to the user. At a high level, one can see this as rating the coherence, or quality, of the text. This is used to improve the ranking of search results, by promoting well-written, content-rich documents while conversely demoting lower quality text.

While building Infocious we encountered various issues and challenges in applying NLP towards Web searching, including scalability, efficiency, usability, and robustness. In the following sections, we discuss our lessons and experiences from applying language analysis towards helping people find information on the Web.

*WWW 2005*, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

# 2. BENEFITS OF NLP TOWARDS WEB SEARCHING

The main goal of the Natural Language Processing (NLP) field is to understand the process of information exchange between humans when they communicate using natural languages. A better understanding of this process would allow computers to extract and operate on information and knowledge represented using natural languages in a more reliable fashion. The field of NLP has a long and rich history, encompassing linguistics, computer science, psychology and even neuroscience. Over the years, a variety of approaches and methodologies have been used to better resolve ambiguities so to extract the semantics within natural language text.

Our goal is to build upon NLP and create a better Web search experience for the users. For our task, we focus on statistical, data-driven techniques, which have experienced a surge of progress in the last decade (e.g., [31] and [27]). The reason for this choice is threefold. First, data-driven NLP requires minimal human effort for creating statistical models. This is particularly important in the case of Web-scale processing because of the voluminous and dynamic nature of Web content. Second, statistical models are very robust in the sense that they will generate an interpretation regardless of the input. This is of paramount importance when coping with the heterogeneous and unpredictable nature of the Web. Third, statistical models are currently the most accurate in resolving ambiguities within natural language text, which is the primary task but also the main challenge of NLP.

We present the most common types of language ambiguity, and show how a statistical NLP approach can be used in resolving each. In addition, we discuss how Web search can benefit from resolving such ambiguities.

## 2.1 Part-of-speech disambiguation

Consider a Web page that contains the two words *house plants*. Depending on the context around it, this phrase may have multiple interpretations. For example, the Web page may be about *plants for inside the house* or it may be about *objects or methods to house plants*. The difference in the meaning comes from the fact that in the first case the word *house* is used as a noun, while in the second case it is used as a verb. A search engine based on keyword matching would not be able to distinguish between the two cases and the returned results might contain a mix of both uses.

Part-of-speech (POS) disambiguation is the process of assigning a part-of-speech tag (such as *noun, verb, adjective*, etc.) to each word in a sentence. By assigning POS tags to each word, we can determine how the word functions within its context. In doing so, we can determine whether *house* is used as a noun or as a verb in the previous example. A search engine can exploit this POS tagging information by restricting the use of the query keywords to a particular POS tag, thus providing results that are more specific to the desired meaning of a query.

## 2.2 Word sense disambiguation

In many cases, words take on a multitude of different meanings (or senses). Such words are called *polysemous*. For example, the word *jaguar* may refer to a car brand-name, an operating system[1] or an animal.[2] The task of distinguishing between the meanings of a polysemous word, is called *word sense disambiguation (WSD)*. Having the word senses disambiguated would allow the users to search for a specific sense of a word, thus eliminating documents containing the same keyword but that are semantically irrelevant.

## 2.3 Phrase identification

Multiple words are typically grouped into phrases to describe a concept more precisely. As individual words are overloaded with multiple usages and meanings, phrases are important for describing a concept more precisely. For example, in the phrases *motor oil* and *cooking oil* the words *motor* and *cooking* are used to describe a more specific type of oil. Phrases, however, are not simply words occurring next to each other. Take for example the sentence *"In the profession of cooking oil is the most important ingredient"*, where *cooking* and *oil* do not form a phrase. Thus a search engine should not consider this sentence relevant to the phrase *cooking oil*. In general, in order to properly identify phrases it is necessary to perform linguistic analysis at a broader context.

## 2.4 Named entity recognition

Named entities refer to names of people, companies, locations, dates, and others. Recognizing the difference between *Jordan* being a person versus a country is the process of *named entity recognition* (NER). A search engine capable of distinguishing different types of name entities would enable users to search specifically for the person or for the country, for example. NER can also be used to extract particular named entities of interest to the user, such as all the companies or locations mentioned in a business article, or all the people mentioned in a newsletter.

## 2.5 Full sentential parsing

Parsing is the process of decomposing a sentence into smaller units, as well as identifying the grammatical role of each and its relationship to other units. Parsing is a well studied problem with many grammar formalisms and parsing algorithms. Parsing is very important for extracting the semantics of sentences precisely.

Consider as an example the sentence *the man who bought shares of Apple fell*. In this case, a parser would be able to determine that *who bought shares of Apple* is a modifier for *the man*, and that it is the man who fell. In the case of simple keyword matching this article may have been returned as a result for the query *shares of Apple fell*. Additionally, parsing can enable very precise searches, since it would allow the user to specify queries based on subjects (e.g., only *Apple* as the subject), main verbs (e.g., only *bought* as the main verb), or even combinations of these linguistic units. This is especially powerful since many structural constructs can be used to express the same semantics, such as *the man, who owns some Apple shares, fell*.

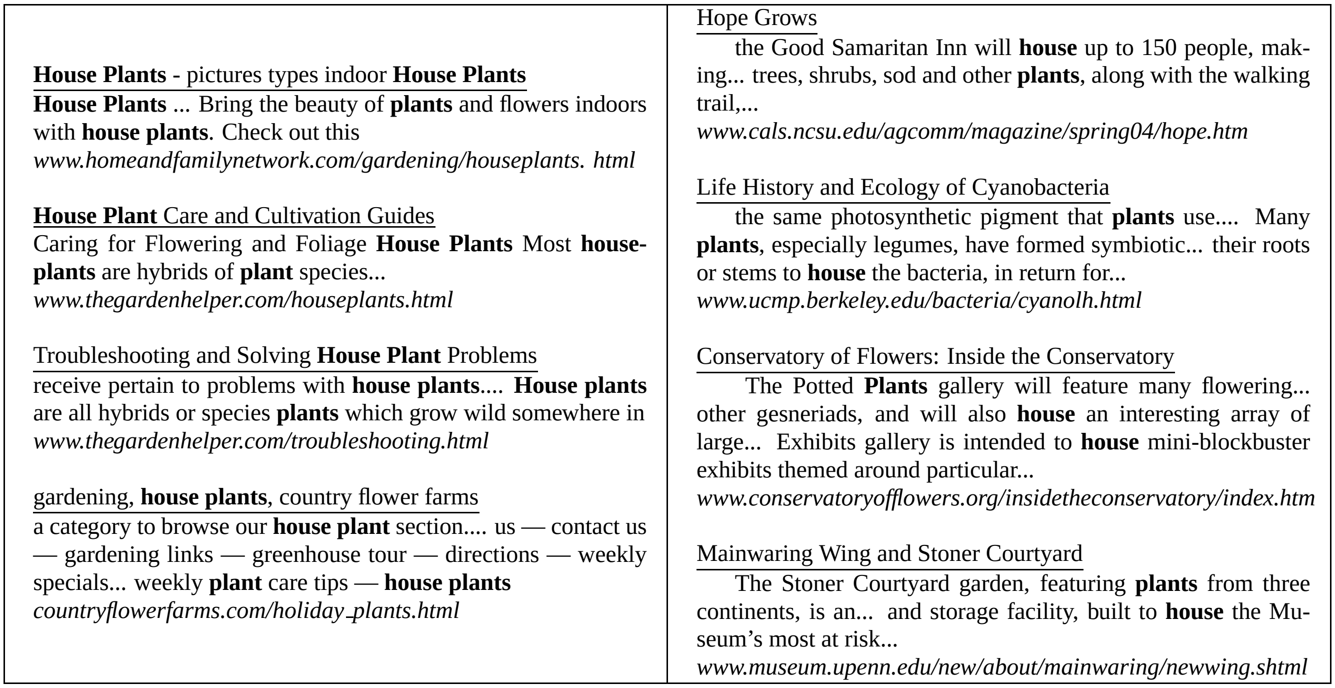
We have presented a very brief overview of the most common types of language ambiguities. Interested readers may refer to [31] and [27] for a more comprehensive treatment of the subject. In the next section, we present the approach that we have taken in Infocious to address each one of the ambiguities just discussed.

# 3. LINGUISTIC ANALYSIS OF WEB TEXT

The task of applying linguistic analysis to improve Web searching involves two main challenges. The first comes from the massive scale and diversity of Web content, making the issues of efficiency and robustness paramount. The second is how to exploit this linguistic analysis to best benefit the user. That is, given that we have resolved various ambiguities through linguistic analysis, how can this improve the way users find information, while making the system simple and intuitive to use? In this section we discuss the first challenge of Web-scale linguistic analysis, and in Section 4 we address the second challenge: how Infocious leverages this analysis to best benefit the user.

[1] MacOS version X.

[2] Scientific name: *Panthera onca*.



**House Plants** - pictures types indoor **House Plants**
**House Plants** ... Bring the beauty of **plants** and flowers indoors with **house plants**. Check out this
*www.homeandfamilynetwork.com/gardening/houseplants. html*

**House Plant** Care and Cultivation Guides
Caring for Flowering and Foliage **House Plants** Most **houseplants** are hybrids of **plant** species...
*www.thegardenhelper.com/houseplants.html*

Troubleshooting and Solving **House Plant** Problems
receive pertain to problems with **house plants**.... **House plants** are all hybrids or species **plants** which grow wild somewhere in
*www.thegardenhelper.com/troubleshooting.html*

gardening, **house plants**, country flower farms
a category to browse our **house plant** section.... us — contact us — gardening links — greenhouse tour — directions — weekly specials... weekly **plant** care tips — **house plants**
*countryflowerfarms.com/holiday_plants.html*

Hope Grows
the Good Samaritan Inn will **house** up to 150 people, making... trees, shrubs, sod and other **plants**, along with the walking trail,...
*www.cals.ncsu.edu/agcomm/magazine/spring04/hope.htm*

Life History and Ecology of Cyanobacteria
the same photosynthetic pigment that **plants** use.... Many **plants**, especially legumes, have formed symbiotic... their roots or stems to **house** the bacteria, in return for...
*www.ucmp.berkeley.edu/bacteria/cyanolh.html*

Conservatory of Flowers: Inside the Conservatory
The Potted **Plants** gallery will feature many flowering... other gesneriads, and will also **house** an interesting array of large... Exhibits gallery is intended to **house** mini-blockbuster exhibits themed around particular...
*www.conservatoryofflowers.org/insidetheconservatory/index.htm*

Mainwaring Wing and Stoner Courtyard
The Stoner Courtyard garden, featuring **plants** from three continents, is an... and storage facility, built to **house** the Museum's most at risk...
*www.museum.upenn.edu/new/about/mainwaring/newwing.shtml*

**Figure 1: Sample search results from Infocious for the query *house plants*, with the default results on the left and the results for *house* used as verb, done via the query *V:house plants* on the right.**

We should stress that our focus of linguistic analysis is placed on the *content* of Web documents and less on the queries. This is because most queries are too short to provide a meaningful context for reliable disambiguation. Instead, ambiguities in the query terms are resolved through examining the *results of queries*, a process described in Section 4.6.1.

## 3.1 Part-of-speech tagging

We treat POS tagging as a probabilistic classification task, i.e.,

$$\tilde{T} = T_{best}(S) = arg\ max_T P(T|S),$$

where $S$ is the input sentence, and $T$ is the set of POS tags assigned to each word of the sentence. In this formulation, the POS assignment for each word $w_i$ in the sentence is treated as a random variable $T_i$. Each variable can take on the values $\{1, \ldots, |N|\}$, where $|N|$ is the number of different POS tags. Therefore, the task is to determine the instantiations of $T$ such that $P(T|S)$ is maximized.

POS tagging is one of the most well-studied problems in NLP and is also one of the most accurate (e.g., [7], [38], and [42]). In Infocious, POS tagging is the first step in the linguistic analysis of every Web page. Our state-of-the-art statistical POS tagger [8] was implemented with efficiency in mind such that it operates at crawling speed. A pre-compiled dictionary is used to improve efficiency. If a word does not appear in the dictionary, we calculate its POS tags based on its prefix or suffix. The Viterbi algorithm [44] is used to determine the most probable tag assignments across a sentence, and this probability is recorded for each sentence in every Web page.

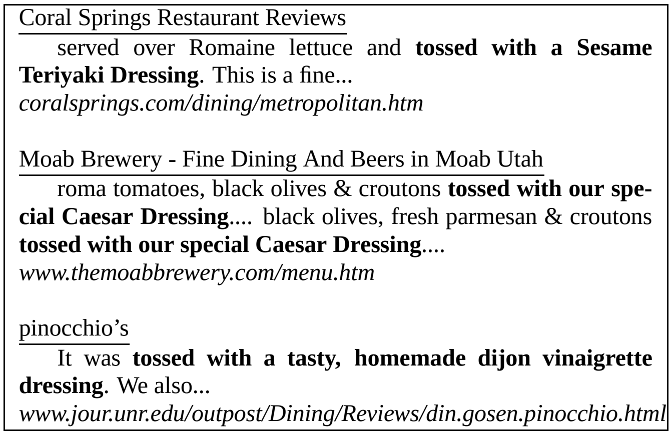
By assigning POS tags for each keyword in the Web pages that it indexes, Infocious can offer its users the choice between different word classes (nouns, verbs, and adjectives) of their ambiguous query words. An example comparison of the search results for *house plants* is shown in Figure 1, with and without distinguishing the POS for the word *house*. On the left side of the figure, results that match *any* POS for the words *house plants* are returned, while on the right side of the figure, the user can restrict the word *house* to be only verb by prepending the *V:* directive before it. This directive is a shortcut for experienced users, since knowing and specifying the POS tag for a query keyword may be burdensome for the average user. Because of this reason, Infocious provides illustrative textual prompts to let the users select the POS tag of interest via hyperlinks, as we will show in Sections 4.6.1 and 4.7.

## 3.2 Phrase identification

Infocious performs phrase identification (also called chunking in the NLP literature) right after POS tagging. Our statistical chunker also treats this task as a probabilistic classification problem, i.e., it assigns a phrase tag for every word (e.g. whether a word is the start of a noun phrase or the end of a verb phrase), so that the overall probability is maximized across the sentence. For each sentence, this outcome probability is combined with the one from POS tagging to reflect the confidence of both disambiguation processes. For an introduction and additional details on chunking and POS tagging, please see [43] and [8].

Based on the chunker's outputs, we extract what we refer to as "concepts" by combining phrases via a set of rules, such as noun-preposition-noun phrases (e.g., *United States of America*), verb-noun phrases, (e.g., *build relationships*), and verb-preposition-noun phrases (e.g., *tossed with salad dressing*). The rules can be specified either manually or can be automatically extracted from annotated collections of text.

We refer to these constructs as concepts because the phrases are reduced to their main components only, i.e., they are stripped of any intervening modifiers or quantifiers. For example, the set of phrases *lightly tossed with oil and vinegar dressing* is reduced to the *tossed with dressing* concept. Similarly, the set of phrases *tossed immediately with blue-cheese dressing* is converted to the same concept. Therefore, a user would be able to find all documents describing the



Coral Springs Restaurant Reviews

served over Romaine lettuce and **tossed with a Sesame Teriyaki Dressing**. This is a fine...
*coralsprings.com/dining/metropolitan.htm*

Moab Brewery - Fine Dining And Beers in Moab Utah

roma tomatoes, black olives & croutons **tossed with our special Caesar Dressing**.... black olives, fresh parmesan & croutons **tossed with our special Caesar Dressing**....
*www.themoabbrewery.com/menu.htm*

pinocchio's

It was **tossed with a tasty, homemade dijon vinaigrette dressing**. We also...
*www.jour.unr.edu/outpost/Dining/Reviews/din.gosen.pinocchio.html*

**Figure 2: Sample search results from Infocious for the concept *tossed with dressing*.**

concept of *tossed salads*, irrespective of the dressing used. A sample of Infocious' search results for this concept is shown in Figure 2.[3]

In effect, this concept extraction process compresses a document into a list of linguistically sound units. This list of concepts is created for every Web page and is used in two ways. First, it is used *externally* as a search aid for the users. We will show how the extracted concepts blend with Infocious' user interface in Section 4.7. Second, the list of concepts is used *internally* to improve the accuracy of determining the topic of a Web page and to detect pages with very similar (or identical) content.

## 3.3 Named entity recognition

Based on the phrases extracted by the chunker, NER is largely a classification task of labeling the noun phrases in a document. This task is again modeled as a statistical tagging problem, calculating $P(E|p)$, where $E$ is the entity tag given a phrase $p$. A gazette, which is an entity dictionary that maps a phrase to its entity type $E$, is compiled from the Web and is used to simplify the NER task. For each proper noun and noun phrase in a document, the NER classifier computes this probability and the class with the maximum probability is chosen as the correct named entity tag.

## 3.4 Word sense disambiguation and page classification

We experimented with a statistical WSD model with state-of-the-art accuracy rates. While our model is sufficiently efficient for Web-scale disambiguation, there were two problems that we encountered: the accuracy of determining the correct sense of a word and the presentation of a word's different meanings to the user. Although our model's accuracy [9] is comparable to the current best, this accuracy remains relatively low compared to other NLP tasks. Additionally, in the hypothetical case that one could perform WSD correctly, there is still the challenge of how to engage users into specifying which sense they are interested in. Our feeling is that users would not be inclined to read a list of definitions before choosing the desired sense for each of their ambiguous query words. Due to these two issues, we decided to put WSD on hold for the moment. Instead, we opted for an intermediate solution for distinguishing between keyword senses through the use of automatic text categorization.

We use classification as a way to organize results and hide the complexities involved with various ambiguities. That is, instead of prompting the user with a list of definitions, Infocious simply organizes the results into categories. Therefore, in the example case of the *jaguar* query, pages about Jaguar cars would fall under the *Automobile* category, whereas pages about the software would be under the *Computers* category. The users can then choose one of these categories to narrow their search.

This feature is made possible by classifying every page within Infocious' index into categories prior to querying. To train our classifier, we have used the category hierarchy from the DMOZ directory [17], along with the documents organized into each of the categories. The classification process is described in more detail in Section 5.

## 3.5 Parsing

From our prior experience with statistical, lexicalized parsers, we believe that full sentential parsing remains too expensive for Web-scale deployment. Having a complexity of $O(n^3 \cdot |G|)$, where $n$ is the number of words in a sentence and $|G|$ is the number of rules in a grammar,[4] one can see that for sentences of non-trivial length, parsing can be quite expensive. While parsing can provide useful information to improve ranking of results, we believe that at present the computational cost does not justify the benefits. Furthermore, parsing also presents the issue of user interface, in that in order to tap into the preciseness of searching parsed data, users may have to master a query language. These are interesting challenges we wish to address in the near future.

## 3.6 Calculating text quality

As Infocious processes the text probabilistically, the resulting probabilities are combined and saved for each sentence. These probabilities are then factored into an overall score for the entire document, which we refer to as the textual quality (or *TextQuality*) of the page. This probability is used during ranking to promote pages with high-quality textual content, as well as during indexing to weigh the relative importance of anchor texts.

In Figure 3 we illustrate the influence of our TextQuality measure on the ranking of search results based on the textual portion of the documents. On the left of Figure 3 we show the results for the query *britney spears* without the TextQuality metric. As seen from the summaries of these results, these pages are mainly composed of secondary phrases containing popular keywords. On the right of Figure 3 we show the results for the same query (i.e., *britney spears*) but we factor the TextQuality metric into the ranking. In this case, the results presented are considered to contain more coherent text which we believe the users would find more informative and useful.

[3] Concept-based searching in Infocious is not identical to traditional phrase searching. Concept-based searching is designed to help the user better organize and navigate search results via our user interface, which is described in Section 4.7. The results shown in Figure 2 may be reproduced via the following URL: `http://search.infocious.com/q?s=%60tossed+with+dressing%60&c0=cab81178c`

[4] A grammar is a set of rules describing the legal construct of the sentences in a given language. One example rule for English is that verbs follow subjects.

**Britney Spears** Pictures - **britney spears** pictures,...
picture of **britney spears**, hot pictures of **britney spears**...
*britney-spears-pictures.hotyoungstars.com/nude/*

**Britney Spears** Breasts - **britney spears** breasts, pics...
breast implant, pictures of **britney spears** breasts, **britney**...
*britney-spears-breasts.hotyoungstars.com/nude/index2.html*

**Britney Spears** Photos - **britney spears** photos,...
**spears, britney spears** nude photos, nude photos of...
*britney-spears-photos.hotyoungstars.com/nude/*

Hot **Britney Spears** Pics - hot **britney spears** pics,...
**britney spears**, new hot pics of **britney spears**,...
*hot-britney-spears-pics.hotyoungstars.com/nude/*

Is **Britney Spears** over the edge?
Is **Britney Spears** over the Edge?... **Britney Spears** is a singer....
*azwestern.edu/modern_lang/esl/cjones/mag/spring2004/britney.htm*

Best Pictures Of **Britney Spears** + wallpapers, facts and funny...
**Britney Spears** comes to us from... **Britney** was a performer since a...
*keanu-reeves.best-pictures.com/spears/britney.html*

IMPERSONATORS - **BRITNEY SPEARS**
Is Proud To Present! Contact: Gary Shortall Back...
*www.impersonators.com/brittany/brit.htm*

**Britney Spears'** Coke Habit
**Britney Spears'** Coke Habit Destroys Her...
*www.emptyv.org/britney_spears.htm*

**Figure 3: Sample search results for the query *britney spears*, comparing the ranking *without* our TextQuality measure on the left and the ranking when it is *included* on the right.**

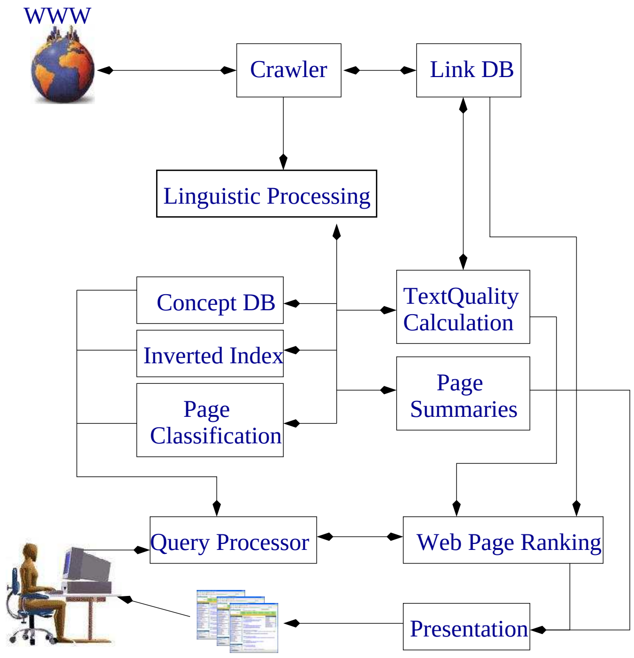


**Figure 4: The architecture of Infocious.**

# 4. THE ARCHITECTURE OF THE INFOCIOUS SEARCH ENGINE

We now describe the overall architecture of Infocious and how our linguistic analysis is used throughout the system to improve Web searching. An overview of Infocious' main modules and their interaction is shown on Figure 4.

## 4.1 Crawling the Web

The crawler is the part of a search engine that handles the task of retrieving pages from the Web and storing them locally for processing. Our distributed crawler behaves like other typical crawlers in the sense that it discovers and follows the links inside a Web page in order to download other Web pages. However, we have extended our crawler with recent research results such that it can provide us with a fresh subset of the Web with a minimal overhead [14], as well as retrieve pages from the so-called Hidden Web [36]. Crawling is a broad topic and readers interested in the topic please refer to [13, 12, 11].

Once the crawler downloads a page from the Web it performs two tasks. First it extracts all the links from the page and sends them to the link database, described next. Second, it hands the page off to the linguistic analysis component.

## 4.2 The link database

The link database performs two functions. First it manages the task of assigning a globally unique ID for every link that the crawler has identified. The second functionality is to store various static properties of the URLs that Infocious is aware of. Such information includes the number of incoming links to a Web page, the number of outgoing links, a content signature, a concept list signature, and the quality of the text described earlier. This information is used during the ranking of results, as well as for rescheduling crawls.

## 4.3 Linguistic processing

This module performs the linguistic analysis that we described in Section 3 for every page that the crawler downloads. We have developed the linguistic analysis module in such a way so as to keep pace with the crawling speed. This module, the heart of Infocious, resolves language ambiguities, appends the linguistic information to the content of a Web page, and sends documents to other modules for processing.

## 4.4 Inverted indexes

An inverted index is a typical data structure used for retrieving documents that are relevant to one or more particular keywords. Given a collection of Web pages, the inverted index is essentially a set of lists (called inverted lists), one for each term[5] found in the collection. For every term, a record (also called a posting) is maintained at its corresponding list. Every posting stores the docu-

[5] Term is used loosely in this context. It can refer to either a single word, a phrase or a concept. In its inverted indexes, Infocious keeps all three kinds of terms.

ment ID that contains a particular term. Additionally, every posting contains the number of occurrences of the term in the document, along with a list of positional information of the term. If a term appears too frequently within the same document, only the first $n$ occurrences are saved, where $n$ was empirically determined after analyzing the typical distributions of words in documents.

Along with every positional record, we maintain information regarding the formating of a term (i.e., whether the term should be rendered in bold, what is the font size, the color, etc.) Furthermore, the index stores functional attributes such as whether the term appears in the URL of the document, whether it appears in the title, etc.

Finally, for every term occurrence in a document we store in the index any associated NLP annotations as identified by the linguistic analysis component. This records any ambiguities resolved by the linguistic analysis module, such as whether a term is used as a noun or a verb. This enables Infocious to return only the documents with the correct meaning to the user.

## 4.5 Page summaries

This module stores and retrieves the NLP-annotated version of the Web pages that Infocious indexes. The module takes as input the data from the linguistic processing modules and stores the pages in a compressed format. Upon returning a document as a search result, the document's summary is retrieved from this module to display the context around the query words.

Additionally, this module stores and retrieves the list of concepts extracted by the NLP module for every document. These concepts are used as navigational aids for users, as well as for improving text categorization, described later.

## 4.6 Answering a query

Infocious supports the standard keyword and phrase searching, as well as searching based on the concepts described earlier. Furthermore, a mixture of keywords, phrases, concepts, and categories is supported, including the ability to exclude concepts or categories deemed undesirable by the users. For example, a user searching for *jaguar* the animal can either select the *Animals* category, or choose to exclude the *Computer* category instead. In addition, the user can specify the part-of-speech tag for any query keyword. For example, the query *V:house plants* will only match documents where the word *house* is used as a verb. On the other hand the query *N:house plants* will retrieve documents where *house* is used as a noun. We should note that the default query semantics in our search engine is the ANDing of the keywords. That is, we return documents which must contain all of the keywords that the user specified.

Given a list of documents that contain the user's keywords and any additional directives (e.g., exclusion or POS tags), Infocious ranks and sorts the result list so that the most relevant documents are displayed first. Ranking is probably the single most important component of a search engine and is also the most challenging. It is also an ongoing process that needs to be constantly tuned and tailored to the dynamic nature of the Web.

With Infocious we take a variety of factors into account for ranking the results. Such factors include the frequency of the keyword in a document, whether the keyword appears in the URL of a page, whether it appears in the title of the page, its relative font size to the rest of the document, etc. We also incorporate link-based properties of the Web pages. That is, pages which are highly linked are (in most cases) considered more important than pages with fewer incoming links.

In addition to the above, we leverage our NLP technology to return pages of greater quality to the user. More specifically, we incorporate the probabilities that the NLP module has calculated during disambiguation into our ranking algorithm. The main idea is that if a page is composed of well-written textual content, it will be promoted, while the opposite will happen for a page with poor textual content.

### 4.6.1 Automatic Query Disambiguation

We also utilize the NLP annotation stored in our index to perform a form of automatic query disambiguation, which is then used to dynamically rank documents according the most likely meaning of a keyword for which the user is querying.

Instead of performing linguistic analysis on the query strings, which are usually too short to establish a reliable context, we instead use the result documents themselves. That is, by gathering statistics on how the query terms are used in context within complete documents, Infocious can disambiguate the query terms based on how people use these query words within the same context.

For example, we can establish that in a majority of documents where the words *train* and *engines* are discussed, *train* is most often used as a noun. We then rank the results based on this meaning of the query word, i.e., promoting documents with the noun usage of *train*. The same principal applies for the opposite case, such as for the query *train pets*, where the verb sense would more likely be used.

Taking this example a step further, consider a more ambiguous query *train chiefs* or a seemingly non-sense query *train grass*. In these cases, there might not be enough evidence in the documents as to decide which of the two senses the word *train* refers to. In such cases Infocious does not assume a particular meaning. Instead, it presents the user with intuitive examples of different usages so he or she can choose the desired meaning.

We conjecture that our method of query disambiguation is more reliable because it draws upon the great number of instances of Web documents where the query words are used in context. On the other hand, directly performing disambiguation on the user's query cannot be as reliable since the context that the user provides is typically very limited. Note that our method of disambiguation comes nearly for free because the NLP analysis is performed and stored in the index ahead of querying.

## 4.7 User interface

When Infocious presents the results to the user, we again tap into our NLP technology to further help users navigate and manage search results. An example of our user interface is shown in Figure 5 for the query *lesson plans*. This is how the search results are presented to the user (along the center), plus any additional search and navigational aids designed to help users in their search quests. We briefly describe each of these aids: [6]

- Infocious presents, along the top and above the search results (Figure 5-2), the categories that the current search results fall into. In our particular case for the query *lesson plans*, these categories include *Education/Educators*, *Education/K through 12*, etc. By hovering over these categories the user can see in real time what category each of the results falls into. The user also has the option of excluding a category from the search results by clicking on the "X" button to the left of the category. In this case, Infocious removes the pages from the excluded category from the results and re-ranks the list.

[6] For more detailed information on the Infocious' user interface, please visit `http://search.infocious.com/about/`.

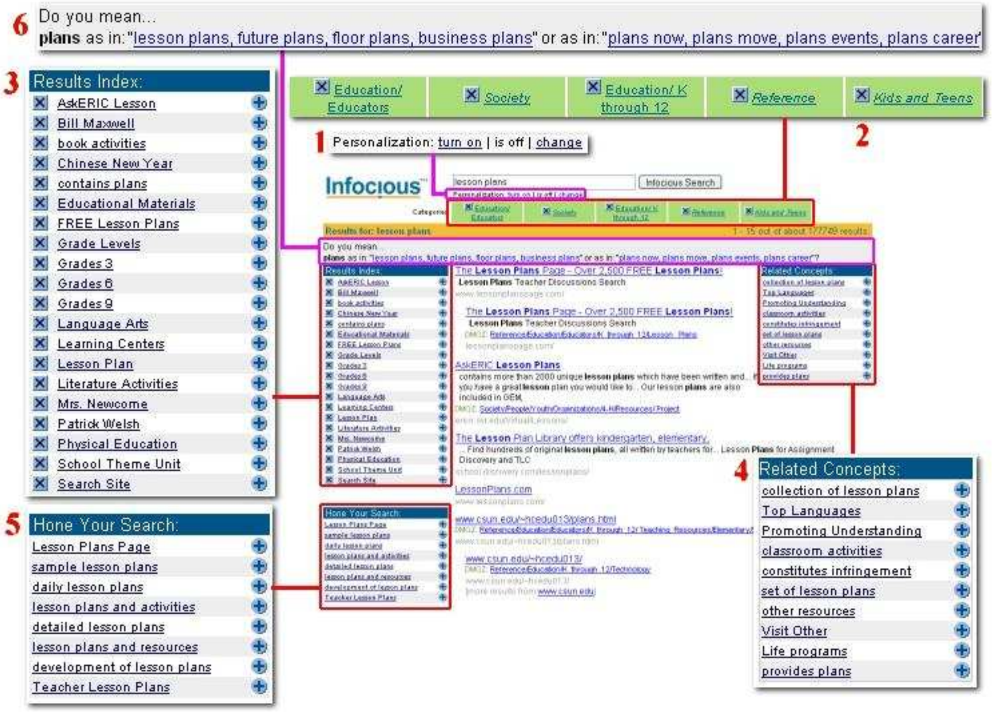


**Figure 5: The Infocious user interface for the query *lesson plans*. Each section of the interface is zoomed in and numbered: (1) Personalization, (2) Categories of the results, (3) Results Index, (4) Related Concepts, (5) Hone Your Search, and (6) Disambiguation suggestion.**

- In this example, the word *plans* is ambiguous and can be a verb or a noun. Because for this query both meanings of *plans* are deemed as probable, Infocious provides the user with links to more precisely specify which meaning of *plans* they are interested in. This is shown on Figure 5-6.

- On the left side of the search results (Figure 5-3), the user can find the "Results Index" list. This list presents the concepts culled from the Web pages during the NLP stage. The "Results Index" is similar to an index at the end of a book, listing the important concepts along with their location in the text. This list provides users with a quick overview of the important concepts within search results, and gives them context in advance about these results before having to visit them. Similar to the Categories feature, the user can exclude any concept on the Results Index list and hence eliminate the corresponding Web Pages from the retrieved results.

- Under the "Results Index" is the "Hone Your Search" list (Figure 5-5). This particular list contains suggestions of longer queries that, when employed, would make the current search more specific. For the current example, Infocious suggests queries such as *sample lesson plans* and *daily lesson plans*, which will help the user further hone their original search.

- To the right of the search results, there is a list of "Related Concepts" (Figure 5-4), which is compiled during the concept extraction phase. This list is provided to help the user expand their search in case they are unfamiliar with the topic they are researching on. For this example Infocious suggests concepts such as *Promoting Understanding* or *classroom activities*. We have found this feature to be particularly useful when the user wants to explore some generic area of interest such as *fuzzy logic*, *Moore's law*, *San Diego*, etc.

- Finally, because Infocious classifies every Web page into categories, it is capable of offering the user the ability to personalize their search results and tailor them to their particular interest. Right below the search box (Figure 5-1) the user can enable or disable personalization. For example, the user might be an avid connoisseur of arts and is not interested at all in sports. By using personalization, users can restrict the results to be within the categories that are of interest to them.

We have presented the major features of Infocious and how they are intended to provide the users with a better, faster, and easier experience in finding what they are looking for. As one can see, most of these features are either enabled by ("Results Index", and "Do You Mean"), made better ("Related Concepts" and "Hone Your Search"), or made more accurate (Categories and Personalization) because of our NLP technology. We further support this claim by demonstrating the benefits of NLP analysis in improving classification accuracy in the next section.

## 5. EVALUATION OF AUTOMATIC CATEGORIZATION

To better address the word-sense disambiguation problem, one of our goals is to automatically classify, as accurately as possible, every Web document into a pre-defined category hierarchy such as the DMOZ directory. In doing so, Infocious enables users to narrow

| Category | Number of documents | Avg # of sentences |
|---|---:|---:|
| Arts | 48,568 | 88 |
| Business | 93,936 | 71 |
| Computers | 43,293 | 104 |
| Games | 8,499 | 92 |
| Health | 18,759 | 115 |
| Home | 6,031 | 116 |
| News | 2,859 | 151 |
| Recreation | 23,239 | 95 |
| Reference | 15,204 | 146 |
| Regional | 258,543 | 75 |
| Science | 23,029 | 136 |
| Shopping | 43,284 | 83 |
| Society | 52,178 | 114 |
| Sports | 23,223 | 94 |
| World | 308,975 | 49 |
| Adult | 8,969 | 69 |
| Kids and Teens | 3,301 | 86 |
| Total | 981,890 | |

**Table 1: Statistics on the collection of Web pages used for evaluating classification accuracy.**

their search to a particular topic, or to personalize the ranking of search results to better match their interests.

What Infocious has in addition to other text classification methods is its large repository of NLP annotated Web pages. In this section, we illustrate through a classification experiment that the additional information that NLP provides can actually improve classification accuracy and therefore can help Infocious to better organize search results.

The text classification problem can be stated simply as follows: given an input document $d$, find the class $c$ it belongs to. A probabilistic formulation of the problem can be: $max_{c \in C} P(c|d)$. However, because DMOZ has close to $600,000$ categories (i.e., $|C| \approx 600,000$), Infocious uses a hierarchical algorithm that employs a different classifier for every parent node in the DMOZ hierarchy. While our algorithm goes beyond the basic Bayesian classifier to improve accuracy, we simplify the experiment here so we can best evaluate and compare the influence of NLP annotations on classification accuracy. Specifically, we will focus on classifying Web pages into one of the top-level categories.

## 5.1 Experimental Setup

Through our crawler and NLP processing module we have the NLP annotated version of most of the web pages that are listed in the DMOZ directory. This data is used as our training corpus to evaluate classification accuracy, i.e., to reproduce the classification done by the DMOZ volunteers given a new Web document.

In the DMOZ directory there are 17 top-level categories. Since DMOZ is organized hierarchically, we not only include documents listed within each top-level category, but also pages from all of its sub-categories. Table 1 shows these categories and the number of documents we collected for this evaluation.

For each document, our preprocessor first discards all formatting elements, tokenizes the document, and detects sentence boundaries. The NLP module then performs POS tagging and phrase detection, and appends the tagging to each word. Lastly, concepts for each document are extracted, sorted based on their *tfidf* values [39], and the top 50 concepts are added to the documents.

| Classifier | Accuracy | stdev |
|---|---|---|
| (1) words only | 64.9% | 0.03% |
| (2) words plus POS tags | 66.1% | 0.03% |
| (3) words plus extracted concepts | 66.3% | 0.04% |
| (4) words plus POS and extracted concepts | 67.6% | 0.04% |

**Table 2: Accuracy results from four classifiers trained on varying amounts of NLP annotations.**

| Category | Classifier 1 | Classifier 4 | % Error Reduction |
|---|---:|---:|---:|
| Arts | 52.01% | 59.59% | 15.8% |
| Business | 56.65% | 60.58% | 9.1% |
| Computers | 58.14% | 61.03% | 6.9% |
| Games | 61.92% | 62.54% | 1.6% |
| Health | 62.10% | 67.23% | 13.6% |
| Home | 32.24% | 35.88% | 5.4% |
| News | 46.35% | 46.74% | 0.7% |
| Recreation | 46.75% | 51.57% | 9.1% |
| Reference | 60.75% | 65.52% | 12.2% |
| Regional | 51.16% | 52.64% | 3.0% |
| Science | 39.89% | 45.64% | 9.6% |
| Shopping | 58.79% | 64.00% | 12.6% |
| Society | 45.14% | 51.20% | 11.1% |
| Sports | 64.38% | 69.80% | 15.2% |
| World | 91.37% | 92.24% | 10.1% |
| Adult | 62.44% | 63.27% | 2.2% |
| Kids and Teens | 11.40% | 13.86% | 2.8% |

**Table 3: Comparison of average accuracy rates and reductions in error rates between individual categories for the classifiers without (Classifier 1) and with NLP annotations (Classifier 4).**

For each experiment we performed 10-fold cross-validation to generate the accuracy rates, with 90/10 split of training and testing data. For classification, all tokens are converted to lower case and words that occur less than five times are replaced with the "unknown" token.

For our experiment here, we chose the Naive Bayes classifier [18] because of its efficiency, important for Web scale processing, and for its accuracy. We compared Naive Bayes to maximum entropy, expectation maximization, and tfidf on a subset of our collection and Naive Bayes was either comparable to or more accurate than the other classifiers.[7] We have also found that Support Vector Machines [15], well known for their classification accuracy, are too computationally expensive for our task.

## 5.2 Results

We trained four classifiers with increasing amount of NLP annotations: (1) words only (i.e., no NLP information), (2) words plus POS tagging, (3) words plus extracted concepts, and (4) words plus POS tagging and extracted concepts. The first classifier serves as our baseline since it does not rely on any NLP information, whereas the last combines two additional annotations. The overall accuracy results are shown in Table 2, whereas in Table 3 the accuracy rates for individual categories are compared between Classifiers 1 and 4.

[7] We plan to report on a detailed study comparing the performance of different classifiers in a future work.

### 5.3 Discussion

The overall accuracy results show that POS tags and extracted concepts individually improved classification accuracy, and by combining both the accuracy improved by 2.7%, i.e., we observed a 7.7% reduction in error rate. While this improvement is modest, we demonstrated that NLP annotations do provide valuable context for improving text classification accuracy.

Table 3 shows the accuracy rates of each top-level DMOZ category. The most accurate category is *World*, which benefits from the English/non-English distinction. The worst is *Kids and Teens*, a relatively recent addition to DMOZ that has a limited number of documents. When comparing between Classifier 1 and 4, one can see a uniform improvement of classification accuracy, with the *Arts* category benefiting from NLP annotations the most.

While these accuracy rates leave room for improvement, it is worth mentioning that the baseline accuracy is comparable to other large-scale text classification studies with a complete set of categories [10, 22, 35, 29].

Inside Infocious, we store both the classification outcomes and their corresponding probabilities in our indexes. Upon ranking of results, pages with higher classification confidence are prioritized over more ambiguous pages, thus reducing the likelihood of erroneous categorization appearing early in the results. This is an example of how Infocious copes with disambiguation errors to minimize the negative impact on the end user.

## 6. RELATED WORK

Some of the earliest research into searching textual information is in the field of information retrieval [39, 45, 5]. Certain approaches proposed by the information retrieval field have been incorporated into the modern Web search engines. One promising approach is the so-called *latent semantic indexing* (LSI) [19, 16], which is capable of locating semantically similar documents in a textual collection. Unfortunately, at present LSI is a very expensive technique to be applied to the scale of the Web.

Web search engines have made significant progress in the last few years. Arguably the very first search engine on the Web was the World Wide Web Worm [32]. The paradigm of Web searching was followed by a variety of search engines such as Altavista [2], Lycos [30], Excite [20], etc. In the last few years, an innovative approach to ranking of the Web pages was introduced by Google [37] and the area of Web searching has advanced even further. At present, besides Google, there is a variety of other popular search engines (e.g., Yahoo! [46], MSNSearch [34], Teoma [41], etc.) All of the aforementioned search engines answer the users' queries by performing keyword matching. In our approach however, we include linguistic analysis in order to improve the search results.

There are also companies such as Autonomy [4], Inquira [24], Inxight [25] and iPhrase [26] that aim to improve information retrieval through the use of language analysis. Although these companies employ some type of linguistic processing in one form or another,[8] they mainly focus on enterprise textual collections. Such collections are typically smaller and more homogeneous than the information available on the Web. Furthermore, their user base and information needs are quite different from the general Web population.

A different approach to combining linguistic analysis with the information on the Web is one that aims at creating an *answer-engine* [1, 28]. That is, given a user's query that is given in the form of a question, the engine tries to come up with a few authoritative answers. Examples of such an approach was the first version of Ask.com [3], the START answering system at MIT [40], and BrainBoost [6]. Although such approaches have potential, we believe that in most cases full sentential parsing is necessary in order to provide a truly reliable service. Other issues include inferencing, the need for common-sense knowledge, and identifying out-liars, all of which are very tough challenges that remain to be solved.

## 7. CONCLUSIONS AND FUTURE WORK

We presented Infocious, a Web search system designed to help users find information more easily and precisely by resolving ambiguities in natural language text. In realizing Infocious, we analyzed current natural language technologies (e.g. POS-tagging, concept extraction, etc.) for their benefits and trade-offs in applying them to Web-scale information retrieval. Equally important are the considerations for enabling the user to exploit the power of these technologies intuitively and transparently.

We believe that Infocious is but the first step in the promising path of realizing the many benefits NLP can have in improving information retrieval, one of the most important tasks performed on the Web today. In its first incarnation described in this paper, Infocious incorporates only but a few of the available NLP technologies, with great opportunities for improvement still left unexplored. It is this challenge that excites and motivates us to further bridge the gap between NLP research and Web searching. Here are some of the challenges we are currently exploring.

Word sense disambiguation: WSD accuracy suffers from the lack of training data. Fortunately, innovative approaches have been proposed to generate them automatically, such as one based on search engines [33]. Since Infocious has amassed large amounts of NLP annotated text, this resource can be used to generate training data for improving WSD models. With reliable word senses Infocious can index directly on word meanings, thus enabling users to search for a specific meaning of polysemous word, such as *living plants* versus *manufacturing plants*.

Full sentential parsing: While time complexity still remains an issue for parsing, the questions of how to represent, index, and query parsed text at the Web scale are largely left unanswered. Nevertheless, the potential benefits for parsing are great, for it can provide for very precise searching, improved text summarization, question answering, machine translation, and others. Finding the best way to bring these benefits to the end user also poses many interesting challenges.

Text classification: More studies are needed to compare different classification algorithms and to better understand the dynamics of categorization errors. For example, examining categorization errors for queries with topical ambiguity, i.e., when Infocious' Categories feature is the most useful to the user, may be more important than aiming for absolute categorization accuracy.

Robustness to disambiguation errors: Even with humans, natural language disambiguation is not perfect. Hence, systems that utilize NLP information need to be robust against errors. We have taken initial steps in Infocious by maintaining probabilities from the NLP disambiguation, but more work is needed to study the impact of NLP errors on search quality, and better ways to cope with them.

Many more possibilities exist for applying our NLP annotated repository to improve other NLP tasks, such as machine translation, text summarization, and question answering. Additionally, we would like to explore the potentials of our NLP technologies to better connect businesses with potential customers. That is, we plan to investigate how Infocious can improve the relevance of advertisements through our better understanding of what users are searching for.

[8] Unfortunately detailed information on the technology of these companies is not publicly available.

## 8. REFERENCES

[1] E. Agichtein, S. Lawrence, and L. Gravano. Learning to find answers to questions on the web. *ACM Trans. Inter. Tech.*, 4(2):129–162, 2004.

[2] Altavista Inc. http://www.altavista.com.

[3] Ask Jeeves Inc. http://www.ask.com.

[4] Autonomy Inc. http://www.autonomy.com.

[5] R. Baeza-Yates and G. Navarro. *Modern Information Retrieval*. Addison-Wesley, 1999.

[6] Brainboost. http://www.brainboost.com.

[7] E. Brill. Some advances in rule-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, Washington, 1994.

[8] G. Chao. *A Probabilistic and Integrative Approach for Accurate Natural Language Disambiguation*. PhD thesis, University of California, Los Angeles, July 2003. http://www.infocious.com/papers/chao-2003.pdf.

[9] G. Chao and M. G. Dyer. Maximum entropy models for word sense disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan, August 2002.

[10] C. Chekuri, M. Goldwasser, P. Raghavan, and E. Upfal. Web search using automatic classification. In *Proceedings of WWW-96, 6th International Conference on the World Wide Web*, San Jose, US, 1996.

[11] J. Cho. *Crawling the Web: Discovery and Maintenance of a Large-Scale Web Data*. PhD thesis, Stanford University, January 2002.

[12] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proceedings of the Twenty-sixth International Conference on Very Large Databases (VLDB)*, September 2000.

[13] J. Cho and H. Garcia-Molina. Synchronizing a database to improve freshness. In *Proc. of SIGMOD conf.*, May 2000.

[14] J. Cho and A. Ntoulas. Effective change detection using sampling. In *Proceedings of the Twenty-eighth International Conference on Very Large Databases (VLDB)*, August 2002.

[15] N. Cristianini and J. Shawe-Taylor. *An Introduction To Support Vector Machines*. Cambridge University Press, 2000.

[16] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, September 1990.

[17] The open directory project. http://www.dmoz.org.

[18] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[19] S. T. Dumais. Latent semantic indexing (LSI) and TREC-2. In *The Second Text Retrieval Conference (TREC-2)*, 1994.

[20] Excite Inc. http://www.excite.com.

[21] Google Incorporated. http://www.google.com.

[22] C.-C. Huang, S.-L. Chuang, and L.-F. Chien. Liveclassifier: creating hierarchical text classifiers through web corpora. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*. ACM Press, 2004.

[23] Infocious Incorporated. http://www.infocious.com.

[24] Inquira Inc. http://www.inquira.com.

[25] Inxight Inc. http://www.inxight.com.

[26] iPhrase Inc. http://www.iphrase.com.

[27] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, N.J., 2000.

[28] B. Katz, J. Lin, D. Loreto, W. Hildebrandt, M. Bilotti, S. Felshin, A. Fernandes, G. Marton, and F. Mora. Integrating web-based and corpus-based techniques for question answering, November 2003.

[29] C. Li, J.-R. Wen, and H. Li. Text classification using stochastic keyword generation. In *Twentieth International Conference on Machine Learning (ICML)*, pages 464–471, 2003.

[30] Lycos Inc. http://www.lycos.com.

[31] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[32] O. A. McBryan. GENVL and WWWW: Tools for taming the web. In *First International Conference on the World Wide Web*, CERN, Geneva, Switzerland, May 1994.

[33] R. Mihalcea. Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Language Resources and Evaluations (LREC)*, Las Palmas, Spain, May 2002.

[34] MSNSearch. http://www.msnsearch.com.

[35] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[36] A. Ntoulas, P. Zerfos, and J. Cho. Downloading hidden web content. Technical report, UCLA, 2004. Available at http://oak.cs.ucla.edu/˜ntoulas/pubs/ntoulas_hidden_web_extended.pdf.

[37] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Database Group, Computer Science Department, Stanford University, November 1999. http://dbpubs.stanford.edu/pub/1999-66.

[38] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the First Conference on Empirical Methods in Natural Language Processing*, pages 133–142, 1996.

[39] G. Salton and M. J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, first edition, 1983.

[40] START natural language question answering system. http://www.ai.mit.edu/projects/infolab/.

[41] Teoma. http://www.teoma.com.

[42] S. M. Thede and M. P. Harper. A second-order hidden markov model for part-of-speech tagging. In *Proceedings of ACL-99*, 1999.

[43] E. F. Tjong Kim Sang and S. Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132, Lisbon, Portugal, 2000.

[44] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–267, 1967.

[45] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kauffman Publishing, San Francisco, 2nd edition, 1999.

[46] Yahoo! Inc. http://www.yahoo.com.