

Multi-Resolution Disambiguation of Term Occurrences

Einat Amitay*, Rani Nelken*, Wayne Niblack**, Ron Sivan*, Aya Soffer*

*IBM Haifa Research Lab, Mount Carmel, Haifa 31905, Israel

**IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120

{einat,rani,ayas,rsivan}@il.ibm.com; niblack@us.ibm.com

ABSTRACT

We describe a system for extracting mentions of terms such as company and product names, in a large and noisy corpus of documents, such as the World Wide Web. Since natural language terms are highly ambiguous, a significant challenge in this task is disambiguating which occurrences of each term are truly related to the right meaning, and which are not. We describe our approach for disambiguation, and show that it achieves very high accuracy with only limited training. This serves as a necessary first step for applications that strive to do analytics on term mentions.

Categories and Subject Descriptors

H.3.3 Information Systems, Information Search and Retrieval

General Terms

Disambiguation

Keywords

Information Retrieval, Text Mining, Natural Language Processing

1. INTRODUCTION

In recent years, the Web's importance as a primary knowledge source has continually increased. Due to its wide availability and distributed structure, it allows a large population of users to express various opinions on an unbounded range of topics and issues. Consequently, the Web today contains a treasure trove of information about *subjects* such as people, companies, organizations, products, etc. that may be of wide interest.

A first step toward any Web-based text mining effort would be to collect a significant number of Web mentions of a subject. However, due to the infamous ambiguity of natural language, many subjects have several meanings. This is particularly true for brand names, which are often derived from names of real word objects. Thus, the challenge becomes not only to find all the subject occurrences, but also to consider only those that have the desired meaning.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.

Copyright 2003 ACM 1-58113-723-0/03/0011...\$5.00.

The easiest method of finding the set of mentions of a subject is to use a search engine. This is feasible for relatively rarely-occurring subjects, such as researchers searching for the pages containing their name, or references to their work, but quickly becomes impractical for commonly occurring subjects such as brand names. For example, consider the term *Santana*. “*Santana*” is of potentially significant commercial value for Sony Music which is the owner of Columbia Records, Epic Records & Legacy Recordings (all published at least one *Santana* record). In order to track what people are saying about their music and about *Santana* in particular on the Web it is necessary to first collect a large number of Web pages that refer to *Santana*. This would presumably be a first step in a larger application that would also apply sophisticated text-mining analyses to these references. However, even this first step is problematic due to the ambiguity of natural language and its use on the Web. Many Web pages refer to *Santana* the flower, the high school, the keelboat, the Cycles, the motor agency of Suzuki in Spain, the NFL player, and to many other *Santan*as, rather than to *Santana* the band or to its guitarist *Carlos Santana*. A Google search for *Santana* yields over 1.3 million hits. Clearly, the average user, who typically examines just the first 1-2 pages of search results, would not be able to go over a large number of them. Even the highly motivated product or brand manager cannot be expected to filter these results without automation.

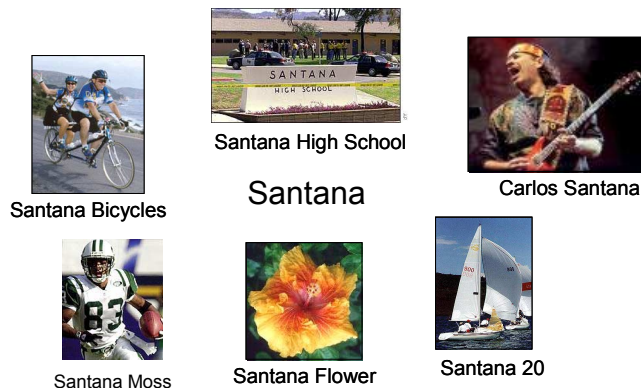


Figure 1 Example for several meanings of the term “Santana” found on the Web

In fact, the problem is even harder, since we are actually interested in filtering subjects at the resolution of individual hits. For instance, a *Carlos Santana*'s Greatest Hits album was recently sold on eBay next to a jersey of the NFL player *Santana Moss*. Both items were on display two sentences apart.

In this paper we present a fully functional system that separates the *on-topic* occurrences and filters them from the potential multitude of *off-topic* references to unrelated entities. For the example above, the system would ideally be able to find

occurrences of *Santana* only when they refer to *Carlos Santana* or his band. Our main contribution lies in presenting a working solution of high accuracy that is scalable to the Web. The algorithm is based on first creating a list of on-topic and off-topic terms that are representative of the domain, and then using the co-occurrence of these terms with the subject as positive or negative evidence for the subject occurrence being on-topic. We explore several methods for acquiring on-topic and off-topic terms. Our evaluation studies show that with limited training and set-up, the system can achieve up to 97% precision with 95% recall on a Web crawl of 2 million pages, and a diverse set of topics.

As a teaser, consider the following excerpt from a biography of the singer *Pink*, which contains 3 occurrences of the term *Pink*. The challenge is to distinguish which of these refers to the singer and which refers to the color name.

Along the way, Alicia Moore earned her nickname, first for her complexion as a child, then for the color of her face when embarrassed and finally after the outgoing Mr. Pink in Quentin Tarantino's Reservoir Dogs. Her hair had been tie-dyed, tinted blue, cornrowed, etc. She thought it would be funny if Pink also had pink hair.

This paper is structured as follows. Section 2 gives a brief survey of related work. Section 3 describes our system. Section 4 presents our evaluation studies, while Section 5 presents a discussion of our contribution and the implications for further research. Throughout this paper we will use the pop music domain as a running example to illustrate the system.

2. RELATED WORK

A well-known strategy for enhancing search accuracy is the use of *query refinement* (See e.g. [13],[10]). In a sense, query refinement can be viewed as a form of disambiguation. By adding on-topic terms to the original query, we may get more precise search results. However, this approach is not particularly helpful for the text-mining problem. The difference is that in search we are typically interested in only the top k out of n results for $k \ll n$. Adding different sets of terms may lead to different sets of top results. However, for text mining, we are interested in a much larger proportion of the results. It is therefore unlikely that a single list of query terms that would be added would yield this. In addition, unlike search, which works at the page granularity, we are interested in the granularity of single subject occurrences.

Another strategy that one can apply to filter out off-topic occurrences is to limit the set of source pages, for instance using focused crawling [2]. By mining just pages that are a-priori likely to be related to a particular domain, we increase the chances that subject occurrences will be on-topic. Unfortunately, this approach is not accurate enough, since many highly ambiguous subject names may appear in the wrong sense even in pages that are related to the domain. For instance, just knowing that a particular page belongs to the music domain does not guarantee that every occurrence of *Next* or *Filter*, both of which are band names as well as dictionary words, is automatically on-topic. Conversely, we may be interested in subject occurrences in atypical pages. Especially for brand management, we may be interested in consumer comments that may originate in pages that do not belong to the set of industry-related pages.

The Natural Language Processing community has focused considerable attention on the problem of Word Sense

Disambiguation (WSD). See [5] for a survey as well as the recent SENSEVAL workshops [6]. While seemingly closely related, the problem definition is in fact different than the one we are trying to solve. Most WSD work assumes each word has a known fixed set of a-priori senses, and the problem is determine the correct sense for each occurrence of a word. For instance, consider the term *Pink*. The Webster dictionary lists several different senses of the word. The WSD task requires a system to tag each occurrence of *pink* in a corpus according to which of these senses it belongs. Due to this problem definition, WSD systems almost invariably rely on a resource that provides different sense entries, such as a dictionary or WordNet [4] as a primary information source. In our case however, we are interested in whether word occurrences belong to a particular domain, and not a particular sense. Unfortunately, we have no domain-differentiating resource akin to a dictionary at our disposal, which makes most WSD techniques irrelevant.

There do exist *unsupervised* approaches to WSD [16] which do not assume a pre-determined set of senses. These approaches work using machine learning techniques to cluster occurrences according to their senses, and then learn these clusters using feature sets extracted from the text. However, it is not clear whether the same approach would work for distinguishing between domains rather than senses. Moreover, these approaches have yet to scale-up convincingly to unrestricted Web-data. In particular, a subject may be mentioned in documents belonging to many different genres or sources. A clustering approach would need to learn the domains while abstracting away from the genres and sources.

A related problem that has been studied is *named entity recognition* [15]. The challenge there is to identify proper names of persons, places, etc. in text. Even a perfect implementation of this approach would be able to solve only part of the problem, e.g. distinguish between *Next* the band name and *next* the adjective/adverb, but it would not be able to help for a name such as *Madonna*, for which all the occurrences are likely to be names, but of different persons (e.g. singer vs. religious icon).

3. THE DISAMBIGUATION PROCESS

Our disambiguation system is based on the classical idea that disambiguation can be achieved by relying on the presence or absence of additional terms that appear in the context of a subject. The basic premise is that the user is interested in a particular domain, which may be identified by a particular vocabulary of *on-topic* terms and *off-topic* terms. We use three different types of terms: single words, multiple word phrases and lexical affinities (LAs) [7]. LAs are pairs of terms that appear together within a fixed-size window of words, in any order.

Disambiguation is done on a particular data-set, which consists of a set of source Web-pages, a set of subjects and a set of on/off topic terms for disambiguation. Data-sets can be defined at varying granularity levels, ranging from the very narrow, e.g. the domain of a particular product or brand, to the very wide, e.g. the domain of a whole industry or set of industry-related topics. We explore several strategies for generating lists of on/off topic terms, ranging from the completely manual to the completely automated.

Once a data-set is defined, first a *Spotter* module searches for subject and on/off-topic term occurrences and tags them. Then, a *Disambiguator* module determines which terms appear in both the

local and global contexts surrounding each occurrence of a subject. The Disambiguator scores these occurrences based on a *tf*idf* measure, and then determines which occurrences are on topic and which are not, using a threshold-based computation with several additional heuristics. The algorithm is implemented as part of a full application which allows easy set-up, viewing the results and fine-tuning.

3.1 The application infrastructure

Our system is implemented on top of a highly scalable and robust application framework for extracting Web-based text analytics developed at the IBM Almaden Research Center. We use a distributed continuous crawler, described in detail in [3], to fetch and update a fresh local copy of Web-pages. Downloaded pages are placed in a dedicated repository, which stores both the original pages and meta-data annotations of them. For the music data-set, we use a collection of 2 million Web pages extracted from a manually generated list of seed site URLs¹. Both the Spotter and Disambiguator are instances of “*miners*”, which are loosely-coupled components that sequentially traverse the pages in the repository. Using a simple network-based API, they read and write annotations of Web-pages. This mechanism allows disparate miners to communicate in an efficient and modular manner. The system is driven by a web-based graphical user interface, allowing users to define subject sets for which to search and disambiguation terms for them.

Source	On Topic Terms	Off Topic Terms
Manual	guy ritchie, like a virgin, die*day, louise, veronica, ciccone, michael jackson, britney spears, evita, music, erotica, who’s that girl, music, pap, groove, dick tracy,	jesus, hospital, university, terry*madonna
KA	madonna*fan, madonna*song, madonna*lyric evita*madonna, light*ray, fan*site, madonna*song, bon*jovi, britney*spears, michael*jackson, musical*single	
Supervised	music*review, exclusive, video, ciccone, vocal, album*music, girl, izine, album*rate, album*review, music* review, live*tv, ritchie girl*material, song, awards, tickets, entertainment*news	church*doors, grieving, vasaris, altar, jesus, calabria*traveler, marble, procession,

Table 1 Example for on/off -topic terms for *Madonna*

3.2 Acquiring on/off-topic terms

Our disambiguation procedure depends on a set of high-quality on/off-topic terms. We’ve experimented with several methods of acquiring such terms, including manual setup, extracting terms from domain-related Web-pages using the Knowledge Agents (KA) system [1], and supervised learning of terms. See Table 1 for a flavour of some of the top terms, phrases, and LAs of the *Madonna* data-set according to source. In the figure, terms are

¹ By crawling particular sites, we are making a first step toward disambiguation. However, as discussed in section 3.2, even more sophisticated strategies of focused crawling are insufficient in themselves to accurately complete the disambiguation task.

separated by commas. LAs are denoted by the pair of their constituents, separated by a “*”.

3.2.1 Manual setup

One method we experimented with to get high-accuracy on/off-topic terms is by relying on the user to provide them. Our experience has shown that it may be quite difficult for a user to come up with a set of terms “out of the blue”. We therefore provide an iterative setup process, in which the user first inputs a set of terms, initiates a mining process, evaluates partial results, and reiterates, after modifying the set of terms, until results are satisfactory. Our system ensures that iterations are fast, requiring only seconds before results begin to appear. Partial mine results are displayed as a table of term occurrences, as shown in Table 2. For instance, a couple of on-topic/off topic examples for the context of “*Madonna*” are shown in Table 3 and Table 4.

URL	Title	Subject	Context	On topic	Evidence
Page URL	Page title	Ambiguous word / subject	Context	Yes/ No	Disambiguation terms

Table 2 Format of output for mine results

URL	www.apple.com.au/documents ... madonnamusic.html
Title	Warner Bros., Maverick and Apple Bring Madonna’s “Music” to the Web
Subject	Madonna
Context	“Music”, released on Maverick/Warner Bros., is <i>Madonna</i> ’s eighth studio album and is co-produced by the artist herself and French dance sensation Mirwais, William Orbit (at the helm of her 1998 smash, the Grammy-winning “Ray of Light”), Spike Stent and Guy Sigsworth
On topic	Yes
Evidence	+music, +album

Table 3 An on-topic term occurrence

URL	www.tnr.com/100900/soskis100900.html
Title	TNR Online A Tale of Two Cities by Benjamin Soskis
Subject	Madonna
Context	Even so, in July, according to one poll, 70 percent of Pennsylvanians still hadn’t heard of him. Says G. Terry <i>Madonna</i> , a pollster from Millersville University: “This race is about Santorum, who is the incumbent, and Klink has not made a case that Santorum is unworthy of reelection, because he hasn’t had the money.”
On topic	No
Evidence	-university, -terry*Madonna

Table 4 An off-topic term occurrence

The number of iterations depends on several factors including the level of complexity of the data-set (i.e. the entropy of the data-set, the diversity of the crawl sources, etc.) and the level of accuracy desired. We have found that 3-5 iterations are sufficient for simple data-sets, whereas complex ones may require 10-20 iterations or even more. In fact, since we have found this iterative setup process to be so useful, we also use it to refine sets of terms acquired using the more automated methods. This process allows the user to progressively refine the set of terms to the utmost

accuracy possible using this method. For the results reported in this paper, we have done only a handful of iterations.

3.2.2 Automatically extracting terms from domain-related Web-pages

As one way of automating the process of acquiring on/off-topic terms we have experimented with extracting terms from Web-pages that are related to the domain of interest. This process reduces the problem of finding on-topic terms to that of finding Web pages that are related to the domain, and profiling them to extract frequently occurring terms and LAs. The rationale is that terms that occur frequently in Web pages belonging to the domain of interest are likely to be indicative of the domain.

To obtain a collection of domain-related pages, we used the Knowledge Agents (KA) system [1]. The idea behind the KA system is to provide an intermediary level between a user and search engines. The system allows users to semi-automatically define domains of interest and focus search results on those particular domains. To define a new domain, the user gives the KA system a set of textual queries, plus optionally a set of sample URLs. The queries are submitted to a commercial search engine, and the set of search results plus sample URLs are expanded by following forward and backwards links to create a set of representative pages for the domain of interest. Pages are scored using a combination of a link-based score and a text-based score. From the set of representative pages, the KA extracts the terms with the top *tf*idf* measures, which it then uses for domain-specific query refinement. We have used the KA architecture to define domains and extract representative terms. From the generated lists of terms, we automatically prune a set of stop-words, as well as the subject names².

In general, we found that this approach yields high-quality on-topic terms and LAs, but also a multitude of seemingly irrelevant terms, which just happen to appear in the pages. In addition, this approach does not yield off-topic terms. Consequently, disambiguation based on the KA has a tendency to introduce many false positives. In particular, a simple strategy of selecting the top several hundred LAs and terms tended to give a classifier that would almost always make a positive decision. To overcome this problem, we chose to use just the LAs and not the other terms taken from the KA. Results are reported in Section 4.

As an alternative method of acquiring domain-specific Web pages, we turned to the Open Directory Project (ODP) [11], which is a human-edited directory of Web-pages. Many natural domains of interest correspond to a directory entry in the ODP. For instance, Madonna's entry in the ODP currently consists of 121 pages. We profiled these pages to extract frequently occurring terms and LAs from them, and used them as on-topic terms. However, this approach failed to yield on-topic terms of good discriminating value.

3.2.3 Supervised learning of terms

Another method we used to acquire on/off-topic terms is by supervised learning. In this method, we run the Spotter on a

² The subject names themselves are likely to appear frequently in the set of domain-related pages, but cannot be taken as on-topic terms, since they would bias the decision to be on-topic for any occurrence of the subject.

collection of pages and produce results as in Table 2 of Section 3.2.1. Rather than using the Disambiguator to decide whether the spots are on or off topic, the user marks this information. We then use this training data to automatically learn on/off topic terms. We profile the different contexts of the spots tagged by the user, extracting frequently occurring terms and LAs from them, removing any stop words. We then take the top terms learned from the positively marked contexts as on-topic terms, and the top terms from the negatively marked contexts as off-topic terms, and use them for disambiguation.

3.3 The Spotter

The Spotter is a general purpose miner for identifying occurrences of arbitrary terms or phrases within documents. The logical split between the Spotter and the Disambiguator allows the Spotter to quickly traverse the content of the repository pages and identify term occurrences. The Spotter is given a list of terms to seek and it tags the documents that contain them with tokens specifying where the terms appear in the document.

We use the Spotter to search both for subjects and for disambiguation (on/off topic) terms, according to the set of subjects and terms specified by the user via the graphical user interface. Subject terms are grouped into synonym sets, for instance, when searching for the subject *Madonna*, the user can configure the Spotter to also look for her real name, *Louise Veronica Ciccone*. The rationale is that when analysing occurrences of a subject, we wish to count all the different variations on a subject name together. Likewise, the Spotter allows some flexibility as to the exact form of the subject name along several dimensions, each separately configurable per subject. These include case-insensitivity, plurals, possessives, and stop words. When these options are turned on³, the Spotter accepts terms, even when they appear with these variations, i.e. in lower or upper case (e.g. the band name *Ace of Base* can be defined using the term *ace of base*), with plural or possessive suffixes (e.g. accept *ace of bases*, *Ace of Base's*) with the omission of stop words (e.g. accept *Ace Base*). While searching for phrases, the Spotter always looks for the longest possible match, ignoring sub-phrases. This feature makes it possible to use off-topic phrases to automatically filter out subject spots when they appear as part of a longer phrase. For instance, by defining *Next record* as an off-topic term, the Spotter would not report occurrences of *Next* followed by *record* as subject spots. We refer to subject occurrences identified by the Spotter as *spots*.

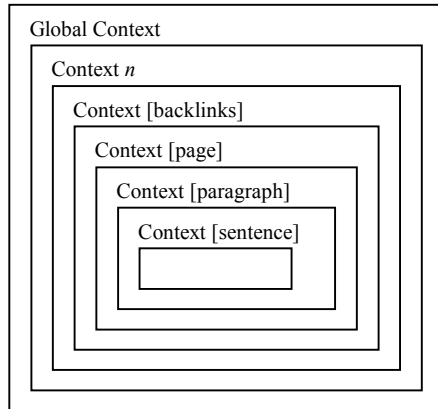
3.4 The Disambiguator

Pages processed by the Spotter are passed to the Disambiguator. The pages are tagged by the spotter with the identified subjects as well as with all on/off topic terms found in the page. For each spot, the Disambiguator collects information about all the terms located within a *context*. Contexts can be of varying resolution, ranging from a window of words surrounding the spot to the entire site containing the page itself, as shown in Figure 1. The Disambiguator collects information for each context and uses this information to disambiguate each spot. The *global context* by default spans over all contexts. The use of multi-resolution contexts is the key to allowing the Disambiguator to work at the

³ By default, case-insensitivity, plurals and possessives are turned on.

granularity level of a single spot, while also taking into account information that appears in varying proximity. For instance, the Disambiguator can determine that a particular spot is on topic based primarily on the sentence-level or paragraph-level contexts even if the more remote global context contains little or no supporting evidence.

For simplicity reasons we describe here a system based on two levels of context, where we use a window of 10 words to each side of the ambiguous term (local context level) and a whole page



referred to in a single URL (global context).

3.4.1 General method

On topic spots are determined using a scoring function on the occurrences of on/off topic terms in the local and global context of the subject occurrence. For a context C , we denote this scoring function by $S(C)$. We first compute the score for the global context, GC . If $S(GC)$ exceeds a *global threshold*, then all occurrences within the global context are considered on topic, and every spot is deemed on topic. Otherwise, we compute $S(C)$ for each nested context. If $S(C) + S(GC)$ exceeds a *local threshold*, the spot is deemed on topic. Otherwise, it is deemed off topic. Global and local thresholds are configurable, and have been determined empirically to maximize accuracy. In addition to this scoring algorithm, the details of which we present in Section 3.4.2, we also use a set of additional heuristics, as described in Section 3.4.3. When using additional context levels, each level will receive a separate score and has separate thresholds associated with it.

The Disambiguator reports its decision per spot by indicating whether it is on or off topic and providing the evidence that led to the decision. The evidence consists of the on/off-topic terms found in the local and global contexts. This information is then displayed to the user as described in Section 3.2.1.

3.4.2 Scoring

For each context C , the Disambiguator computes a score, $S(C)$ based on a variant of $tf*idf$ measures for the on/off topic-terms, as follows:

$$S(C) = \sum_{t \in C} W_t \cdot t_f \cdot idf_t$$

In this formula, t is a term, phrase, or lexical affinity. W_t denotes t 's weight, which depends on two factors. The weight's sign depends on whether the term is defined as being on topic or off topic, where on topic terms are positive, and off-topic terms are negative. The weight's absolute value depends on whether the term is a single word, a longer phrase or a lexical affinity, where phrases and lexical affinities are assigned a higher value, to increase their relative contribution. $tf*idf$ is measured per context (local and global), where the $tf*idf$ of a term t , denoted $tf*idf_t$ is computed using the formula:

$$tf_t = \sqrt{N_t}$$

where N_t denotes the number of occurrences of a term (phrase / LA) t in the given context. The rationale behind using the square root function rather than the more common algorithm is to amplify the impact of recurring terms. This is especially important for the nested contexts which can be very small. Hence the expected number of occurrences in a local context is too small for a logarithmic scale to confer sufficient significance on recurring terms. Note that this makes the occurrence of multiple different terms within a context more significant than the same number of occurrences of a single term.

For $tf*idf$, we use an approximation based on TREC Web-Track data. For the sake of simplicity, we compute $tf*idf$ for single terms only, as getting accurate information for phrases and LAs for general Web pages is much more complex. As a smoothing method, terms that do not have approximate $tf*idf$ s in our collection, including phrases and LAs, are given a default $tf*idf$ representing a small number of occurrences.

3.4.3 Heuristics

We apply a set of heuristics to augment the scoring method described above. These include:

- **Always-on-topic subjects:** Some subject names are unambiguous, and can be manually labelled as always-on-topic. For instance, any occurrence of *Linkin' Park* is almost certainly an unambiguous occurrence of the band name. By defining it as always-on-topic, the Disambiguator automatically decides that it's on topic, without even examining the context.
- **Always on/off topic terms:** Particular terms can be tagged as being always-on (off) topic, signifying that their occurrence within a local context automatically makes the subjects in the context on (off) topic. Again, this heuristic bypasses the regular scoring method. Always-on-topic subjects are automatically tagged as always on-topic terms as well.
- **Majority rules:** In special case where over two thirds of the spots on the page context are determined to be on (off) topic by the regular scoring rules, then all the spots on the page are set to be on (off) topic. This heuristic is for pages that contain a large number of spots. In some cases, several of the occurrences have sufficient (positive or negative) evidence in their smaller contexts to classify them correctly, while others do not. In such cases, it is usually fair to assume that if a large majority of the spots are classified in one direction, then the rest of spots should also be classified in the same way.

4. EXAMPLES AND EVALUATION

We now describe our evaluation of the accuracy of the Disambiguator’s results. For our experiments, we used three music-related data-sets:

- **A *Madonna* data-set.** This represents a case in which most spots are expected to be on-topic.
- **A *Pink* data-set,** for which most spots are expected to be off-topic.
- **A combined collection of band names** including: *187, Babel Fish, Binocular, Camus, Ivy, The Doors, The Hives, Train*. This illustrates a data-set in which we are interested in a collection of several different subjects together. It is somewhat artificial since the subjects do not form a coherent set, but were chosen due to their high level of ambiguity.

We first describe the evaluation methodology in Section 4.1 and then turn to actual experimental results in Section 4.2. We give performance measurements in Section 4.3.

4.1 Evaluation methodology

Evaluating the Disambiguator’s accuracy is done by comparing the Disambiguator’s results to a manually determined gold set (or ground truth) standard.

4.1.1 Creating a gold set standard

To create a gold set standard, we first used the Spotter to identify subject occurrences. We then asked a judge to manually traverse a set of 1000 spot contexts in the format described in Section 3.2.1, for each data-set. The judge marked each spot as on or off topic. Interestingly, the human judge could almost always confidently determine whether the spot was on topic or not. For some very rare cases, it remained virtually impossible to disambiguate the spot⁴. We have found a single judge to be sufficient to create the gold set since our experiments show an almost absolute level of agreement between different judges. We have been careful to separate the training and evaluation data-sets. In particular, for both the manual and supervised learning approach, both of which require processing examples of on/off-topic spots, we have used a separate training set than the gold set of results.

4.1.2 Accuracy measures

To gauge the Disambiguator’s accuracy, we used the standard measures of *Precision*, P , and *Recall*, R . Precision denotes the ratio of spots correctly identified by the Disambiguator as on topic out of all the spots which the Disambiguator reports as being on topic. Recall denotes the ratio of spots correctly identified as on topic out of all the spots that actually are on topic.

As usual, there is a trade-off between precision and recall. For the problem at hand, we have a preference for precision over recall. In other words, we prefer false negatives over false positives. The rationale behind this preference lies with the intended use of the

⁴ For instance, one such case involved a large corporation offering travel packages. Since the corporation involved has diverse business offerings, it was non-trivial to determine whether the company offering the travel package was really the corporation or some other company of the same name. Another example involved news coverage of the painting of a Madonna mural, for which it is quite difficult to determine just from the text which Madonna is depicted.

data. Since there are many spots in any case, it is acceptable to miss a few. However, false positives are worse, since they introduce noise into most analytic measures that we may wish to apply to the data. For instance, assume we would collect terms that co-occur with a particular subject. Having false negatives would just decrease our sample size, while having false positives would introduce co-occurring terms that are in fact unrelated to our subject. Moreover, if we provide the user with a means to perform drill-down on on-topic hits, we would not want to drill down on incorrectly identified spots. To factor-in this preference, we use the E Evaluation measure of Van Rijsbergen [14] (which is essentially $1 -$ a weighted *harmonic mean*). E is thus defined as follows:

$$E = 1 - \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}}$$

where R denotes recall and P denotes precision. b is a parameter, such that values of $b < 1$ reflect that we are more interested in recall than precision. We use a value of $b=0.5$. Accuracy increases as E approaches 0.

4.1.3 Confidence intervals

It is important to note that our evaluation of the Disambiguator’s results on the gold set provides only an approximation of the true accuracy. After all, we evaluated our algorithm on a gold set of 1000 spots out of the 2,000,000 pages. A natural question is therefore how good this approximation is. In other words, what is our confidence level that the accuracy measures on the gold set reflect true accuracy on the full set. This problem is resolved in the context of machine learning by Mitchell [9]. For a collection C , define $error_C$ to be the ratio of false positive and false negatives in the algorithm’s results on C . Our evaluation of a sample G gives us $error_G$. Mitchell makes the assumption that the probability of having a particular ratio of errors is approximated by a normally distributed random variable with mean $error_G$ and standard deviation:

$$\sqrt{\frac{error_G(1 - error_G)}{|G|}}$$

where $|G|$ is the sample size.⁵ In other words, the true error can be viewed as laying in a bell-curve centred on the observed error. Therefore with probability $N\%$, $error_C$ is within z_N standard deviations of $error_G$, where z_N is the z -value. In particular, there is a 95% chance that $error_C$ is within 1.96 standard deviation from $error_G$. For instance, for an observed error ratio of 10% on the gold set, there is a 95% chance that the error on the full collection is 7.5%-12.5%.

4.2 Experimental results

We now report the Disambiguator’s results on the data-sets. As a baseline, we use a Bernoulli decision process that with probability

⁵ Note that for our gold sample of 1000 spots, it would be inaccurate to set $|G|$ at 1000, since the assumption is that samples are drawn independently, which is not true for spots on the same page. Hence, it is more accurate to take $|G| \approx 600$ (reflecting the number of independent pages in the sample).

p determines the spot as being on topic, and with probability $1-p$ determines it to be off-topic.

Results for the *Madonna* data-set are given in Figure 2 and Figure 3. We found the probability of being on topic for this data-set to be 0.866. Consequently, a simple-minded disambiguation scheme that would always report a positive answer is therefore very likely to be right in the majority of cases. In fact, it will have a perfect recall value, with a precision value that is exactly the probability of being on-topic. This is shown in Figure 2 as a Bernoulli process with $p=1$.

As another baseline, we used a Bernoulli process with $p=0.866$. The KA yields a precision of over 95%. As discussed above, to achieve this high level of precision, we used just the LAs extracted from the KA. As a result, the recall is quite low. Both the supervised learning and the manual approaches show extremely high precision: 97-99%. This comes at a recall level of 80-85%. Note that for the manual training, we only did a handful of iterations. In principle, it is possible to further refine the list of terms until accuracy is much higher.

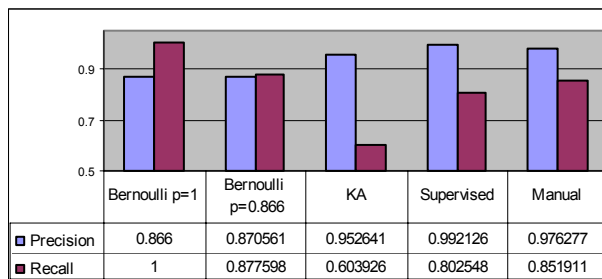


Figure 2 – Precision and Recall for *Madonna*

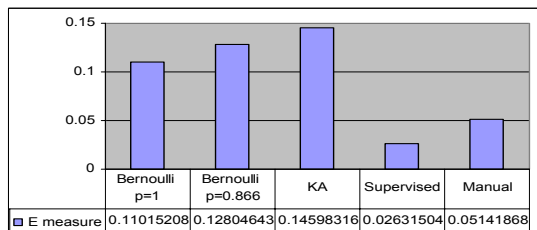


Figure 3 – E measure of Van Rijbergen for *Madonna*

Results for the *Pink* data-set are shown in Figure 4 and Figure 5. For *Pink*, the percentage of on-topic occurrences in our gold data-set was 0.267. Intuitively, this makes *Pink* a tougher case to disambiguate. For instance, simply saying “yes” continues to yield a perfect recall value, but the precision is much degraded (again reflecting the ratio of on-topic occurrences in the gold set). Lowering p does not change the precision, while harming recall. Both the KA and supervised learning are able to achieve perfect precision but at lower levels of recall⁶. Finally, the results from the manual run illustrate the inherent difficulty of finding good on/off topic terms, and require more domain-specific knowledge.

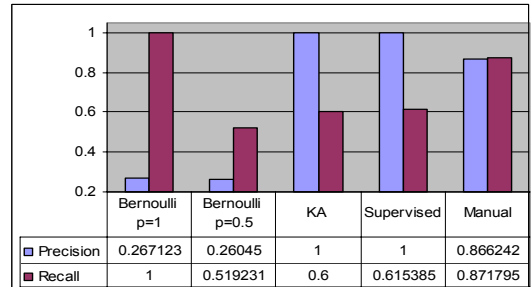


Figure 4 – Precision and Recall for *Pink*

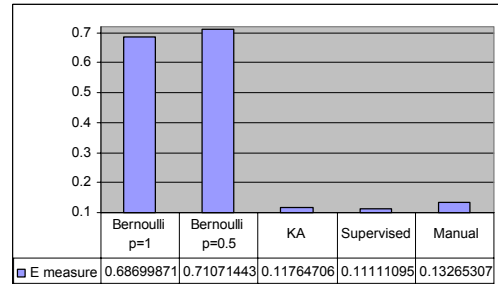


Figure 5 – E measure of Van Rijbergen for *Pink*

Results for the combined collection of bands are shown in Figure 6 and Figure 7. In a sense, this data-set is different from the preceding ones in that it combines several diverse subjects, which are disambiguated together. This raises interesting issues of interaction. For instance, there may be cases where terms provide positive evidence for one subject, while simultaneously providing negative evidence for another. A case in point is the phrase *Long Distance*, which happens to be the name of an album by *Hy*, but an off-topic term for the subject *Train*.

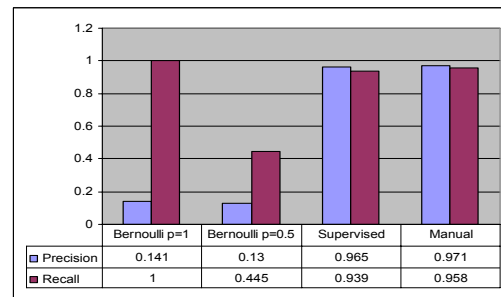


Figure 6 – Precision and Recall for *Bands*

For the gold set, the ratio of on-topic hits was 0.141. Similarly to *Pink*, this means that the Bernoulli measures give very poor precision. We were unable to create a KA that gave meaningful results for this data-set, since it cannot be viewed as a coherent domain. It is therefore not surprising that it is quite hard to create a single KA that captures a representative set of pages (and consequently terms) that distinguish this set. Both the supervised learning and manual terms approach yield excellent results.

⁶ The extraordinary result for precision is achieved as follows: we collected a gold set of just 584 hits, of which 96 were correctly classified as on-topic.

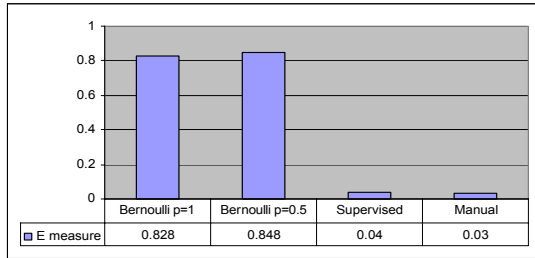


Figure 7 - E measure of Van Rijsbergen for Bands

4.3 Performance

Our experimental setup consisted of a single machine running the miners communicating with another machine which ran the repository of pages. The machines used were 1 Gigabyte Pentium-3 machines running Linux. Processing 2 million pages takes just over 11 hours on an average of 20 milliseconds per page.

5. CONCLUSION

Our experiments show that the basic premise of locating on/off topic terms in the context of spotted subjects is extremely helpful for disambiguation. As we have seen, this approach is able to achieve high levels of accuracy with limited training. Thus, the main challenge lies in generating highly accurately discriminating terms. All the techniques we have experimented with require some amount of training or teaching from the user. There is a trade-off between the amount of labour required and the degree of accuracy required. It is possible, and indeed may be worthwhile to fine-tune a set of disambiguation terms until results are extremely accurate. Our system provides a convenient method for doing so. More automated techniques tend to introduce considerable noise, and thus cannot be used in isolation.

The fact that we are disambiguating Web pages has many repercussions, especially with respect to the scalability and noise. An interesting question is whether the Web's structure can aid in the disambiguation decision. In particular, would taking into account the pages that link to or are linked from the page help disambiguation decisions? There is some anecdotal positive evidence that this indeed is the case. For instance, we encountered a page that contains a court ruling in a case related to a domain name dispute. The page cited a precedent regarding a similar dispute on the *Madonna.com* domain name. While the page itself contained very limited *Madonna*-related terms, one of the pages linking to it is a less formal news report that mentions *Madonna* by name, and provides a lengthy description of her with sufficient on-topic terms. However, in many other cases, linked/linking pages provided "more of the same" language cf. [8]. These pages were likely to be disambiguated in the same way as the original page (if they contained a spot at all), and thus were not particularly helpful.

The main challenge for future work thus remains exploring methods of introducing more automation into the process of disambiguation. In particular, we believe it may be worthwhile to examine methods that start off with the techniques that give good precision results such as the supervised learning and KA (for coherent domains), and then use these to automatically learn more terms so as to increase recall.

6. ACKNOWLEDGMENTS

We wish to thank David C. Smith, Donald S. Bell, Jasmine Novak, Zengyan Zhang and Jan H. Pieper for their part in the implementation aspects of the system.

7. REFERENCES

- [1] Aridor Y., Carmel D., Lempel R., Maarek Y., and Soffer A. (2000). Knowledge agents on the web. In Proceedings of the 4th International Workshop on Cooperative Information Agents, CIA 2000, LNAI 1860, pages 15--26.
- [2] Chakrabarti S., Dom B., and van den Berg M. (1999). Focused crawling: A new approach to topic-specific web resource discovery. *WWW8 / Computer Networks*, 31(11-16):1623-1640.
- [3] Edwards J., McCurley K.S., and Tomlin J. (2001). An adaptive model for optimizing performance of an incremental web crawler. In Proceedings of the 10th International World Wide Web Conference (WWW10), pp 106-113.
- [4] Fellbaum C., editor. (1988). *WordNet: An On-Line Lexical Database and Some of Its Applications*. MIT Press, Cambridge, Mass., 1988.
- [5] Ide N. and Véronis J. (1998). Word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1-40.
- [6] Kilgariff A. (1998). Senseval: An exercise in evaluating word sense disambiguation programs. In LREC, Granada, May 1998, pages 581--588.
- [7] Maarek Y. and Smadja F. (1989). Full text indexing based on lexical relations, an application: Software libraries. In Proceedings of SIGIR '89, pp. 198-206.
- [8] Menczer F. (2002). Links tell us about lexical and semantic web content. Unpublished manuscript, university of Iowa, 2002.
- [9] Mitchell T. (1997). *Machine Learning*. McGraw Hill.
- [10] Mitra M., Singhal A., and Buckley C. (1998). Improving automatic query expansion. In Proceedings of SIGIR '98, pp. 206-214.
- [11] <http://www.dmoz.org>
- [12] Pedersen T. and Bruce R. (1998). Knowledge lean word-sense disambiguation. In Proceedings of AAAI/IAAI, pp. 800-805.
- [13] Rocchio J.J. (1971). Relevance feedback in information retrieval. In "The SMART Retrieval System: Experiments in Automatic Document Processing", pp. 313-323, Englewood Cliffs, NJ, Prentice-Hall Inc.
- [14] van Rijsbergen C.J. (1979). *Information Retrieval*. Butterworths, 1979.
- [15] Wacholder N, Ravin Y., and Choi M. (1997). Disambiguation of proper names in text. In Proceedings of the 5th Conference on Applied Natural Language Processing, pp. 202-208.
- [16] Yarowsky D. (1995). Unsupervised word sense disambiguation rivalling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pp. 189-196.