Using Encyclopedic Knowledge for Named Entity Disambiguation

Razvan Bunescu[*]
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712-0233
razvan@cs.utexas.edu

Marius Paşca
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
mars@google.com

Abstract

We present a new method for detecting and disambiguating named entities in open domain text. A disambiguation SVM kernel is trained to exploit the high coverage and rich structure of the knowledge encoded in an online encyclopedia. The resulting model significantly outperforms a less informed baseline.

1 Introduction

1.1 Motivation

The de-facto web search paradigm defines the result to a user's query as roughly a set of links to the best-matching documents selected out of billions of items available. Whenever the queries search for pinpointed, factual information, the burden of filling the gap between the output granularity (whole documents) and the targeted information (a set of sentences or relevant phrases) stays with the users, by browsing the returned documents in order to find the actually relevant bits of information. A frequent case are queries about named entities, which constitute a significant fraction of popular web queries according to search engine logs. When submitting queries such as *John Williams* or *Python*, search engine users could also be presented with a compilation of facts and specific attributes about those named entities, rather than a set of best-matching web pages. One of the challenges in creating such an alternative search result page is the inherent ambiguity of the queries, as several instances of the same class (e.g., different people) or different classes (e.g., a type of snake, a programming language, or a movie) may share the same name in the query. As an example, the contexts below are part of web documents referring to different people who share the same name *John Williams*:

1. "*John Williams* and the Boston Pops conducted a summer Star Wars concert at Tanglewood."

2. "*John Williams* lost a Taipei death match against his brother, Axl Rotten."

3. "*John Williams* won a Victoria Cross for his actions at the battle of Rorke's Drift."

The effectiveness of the search could be greatly improved if the search results were grouped together according to the corresponding sense, rather than presented as a flat, sense-mixed list of items (whether links to full-length documents, or extracted facts). As an added benefit, users would have easier access to a wider variety of results, whenever the top 10 or so results returned by the largest search engines happen to refer to only one particular (arguably the most popular) sense of the query (e.g., the *programming language* in the case of *Python*), thus submerging or "hiding" documents that refer to other senses of the query.

In various natural language applications, significant performance gains are achieved as a function of data size rather than algorithm complexity, as illustrated by the increasingly popular use of the web as a (very large) corpus (Dale, 2003). It seems therefore natural to try to exploit the web in order to also improve the performance of relation extraction, i.e. the discovery of useful relationships between named entities mentioned in text documents. However, if one wants to combine evidence from multiple web pages, then one needs again to solve the name disambiguation problem.

[*] Work done during a summer internship at Google.

Without solving it, a relation extraction system analyzing the sentences in the above example could mistakenly consider the third as evidence that *John Williams* the composer fought at *Rorke's Drift*.

1.2 Approach

The main goal of the research reported in this paper is to develop a named entity disambiguation method that is intrinsically linked to a dictionary mapping proper names to their possible named entitiy denotations. More exactly, the method:

1. Detects whether a proper name refers to a named entity included in the dictionary (*detection*).

2. Disambiguates between multiple named entities that can be denoted by the same proper name (*disambiguation*).

As a departure from the methodology of previous approaches, the paper exploits a non-traditional web-based resource. Concretely, it takes advantage of some of the human knowledge available in Wikipedia, a free online encyclopedia created through decentralized, collective efforts of thousands of users (Remy, 2002). We show that the structure of Wikipedia lends itself to a set of useful features for the detection and disambiguation of named entities. The remainder of the paper is organized as follows. Section 2 describes Wikipedia, with an emphasis on the features that are most important to the entity disambiguation task. Section 3 describes the extraction of named entity entries (versus other types of entries) from Wikipedia. Section 4 introduces two disambiguation methods, which are evaluated experimentally in Section 5. We conclude with future work and conclusions.

2 Wikipedia – A Wiki Encyclopedia

Wikipedia is a free online encyclopedia written collaboratively by volunteers, using a wiki software that allows almost anyone to add and change articles. It is a multilingual resource - there are about 200 language editions with varying levels of coverage. Wikipedia is a very dynamic and quickly growing resource – articles about newsworthy events are often added within days of their occurrence. As an example, the September 2005 version contains 751,666 articles, around 180,000 more articles than four months earlier. The work in this paper is based on the English version from May 2005, which contains 577,860 articles.

Each article in Wikipedia is uniquely identified by its title – a sequence of words separated by underscores, with the first word always capitalized. Typically, the title is the most common name for the entity described in the article. When the name is ambiguous, it is further qualified with a parenthetical expression. For instance, the article on *John Williams* the composer has the title *John_Williams_(composer)*.

Because each article describes a specific entity or concept, the remainder of the paper sometimes uses the term 'entity' interchangeably to refer to both the article and the corresponding entity. Also, let E denote the entire set of entities from Wikipedia. For any entity $e \in E$, $e.title$ is the title name of the corresponding article, and $e.T$ is the text of the article.

In general, there is a many-to-many correspondence between names and entities. This relation is captured in Wikipedia through *redirect* and *disambiguation* pages, as described in the next two sections.

2.1 Redirect Pages

A *redirect page* exists for each alternative name that can be used to refer to an entity in Wikipedia. The name is transformed (using underscores for spaces) into a title whose article contains a redirect link to the actual article for that entity. For example, *John Towner Williams* is the full name of the composer *John Williams*. It is therefore an alternative name for the composer, and consequently the article with the title *John_Towner_Williams* is just a pointer to the article for *John_Williams_(composer)*. An example entry with a considerably higher number of redirect pages is *United_States*. Its redirect pages correspond to acronyms (*U.S.A.*, *U.S.*, *USA*, *US*), Spanish translations (*Los_Estados_Unidos*, *Estados_Unidos*), misspellings (*Untied_States*) or synonyms (*Yankee_land*).

For any given Wikipedia entity $e \in E$, let $e.R$ be the set of all names that redirect to e.

2.2 Disambiguation Pages

Another useful structure is that of *disambiguation pages*, which are created for ambiguous names, i.e. names that denote two or more entities in Wikipedia. For example, the disambiguation page for the name *John Williams* lists 22 associated

TITLE	REDIRECT	DISAMBIG	CATEGORIES
John Williams (composer)	John Towner Williams	John Williams	Star Wars music, ... Film score composers, 20th century classical composers
John Williams (wrestler)	Ian Rotten	John Williams	Professional wrestlers, People living in Baltimore
John Williams (VC)	*none*	John Williams	British Army soldiers, British Victoria Cross recipients
Boston Pops Orchestra	Boston Pops, The Boston Pops Orchestra	Pops	American orchestras, Massachusetts musicians
United States	US, USA, ... United States of America	US, USA, United States	North American countries, Republics, United States
Venus (planet)	Planet Venus	Venus, Morning Star, Evening Star	Venus *Planets of the Solar System*, *Planets*, *Solar System*, ...

Table 1: Examples of Wikipedia titles, aliases and categories

entities. Therefore, besides the non-ambiguous names that come from redirect pages, additional aliases can be found by looking for all disambiguation pages that list a particular Wikipedia entity. In his philosophical article "On Sense and Reference" (Frege, 1999), Gottlob Frege gave a famous argument to show that sense and reference are distinct. In his example, the planet Venus may be referred to using the phrases "morning star" and "evening star". This theoretical example is nicely captured in practice in Wikipedia by two disambiguation pages, *Morning_Star* and *Evening_Star*, both listing Venus as a potential referent.

For any given Wikipedia entity $e \in E$, let $e.D$ be the set of names whose disambiguation pages contain a link to e.

2.3 Categories

Every article in Wikipedia is required to have at least one category. As shown in Table 1, *John Williams (composer)* is associated with a set of categories, among them *Star Wars music*, *Film score composers*, and *20th century classical composers*. Categories allow articles to be placed into one or more topics. These topics can be further categorized by associating them with one or more parent categories. In Table 1 *Venus* is shown as both an article title and a category. As a category, it has one direct parent *Planets of the Solar System*, which in turn belongs to two more general categories, *Planets* and *Solar System*. Thus, categories form a directed acyclic graph, allowing multiple categorization schemes to co-exist simultaneously. There are in total 59,759 categories in Wikipedia.

For a given Wikipedia entity $e \in E$, let $e.C$ be the set of categories to which e belongs (i.e. e's immediate categories and all their ancestors in the Wikipedia taxonomy).

2.4 Hyperlinks

Articles in Wikipedia often contain mentions of entities that already have a corresponding article. When contributing authors mention an existing Wikipedia entity inside an article, they are required to link at least its first mention to the corresponding article, by using *links* or *piped links*. Both types of links are exemplified in the following wiki source code of a sentence from the article on Italy: *"The [[Vatican City|Vatican]] is now an independent enclave surrounded by [[Rome]]"*. The string from the second link (*"Rome"*) denotes the title of the referenced article. The same string is also used in the display version. If the author wants another string displayed (e.g., *"Vatican"* instead of *"Vatican City"*), then the alternative string is included in a piped link, after the title string. Consequently, the display string for the aforementioned example is: *"The Vatican is now an independent enclave surrounded by Rome"*. As described later in Section 4, the hyperlinks can provide useful training examples for a named entity disambiguator.

3 A Dictionary of Named Entities

We organize all named entities from Wikipedia into a dictionary structure D, where each string entry $d \in D$ is mapped to the set of entities $d.E$ that can be denoted by d in Wikipedia. The first step is to identify named entities, i.e. entities with a proper name title. Because every title in Wikipedia must begin with a capital letter, the decision whether a title is a proper name relies on the following sequence of heuristic steps:

1. If $e.title$ is a multiword title, check the capitalization of all content words, i.e. words other than prepositions, determiners, conjunctions, relative pronouns or negations. Consider e a named entity if and only if all content words are capitalized.

2. If $e.title$ is a one word title that contains at least two capital letters, then e is a named entity. Otherwise, go to step 3.

3. Count how many times $e.title$ occurs in the text of the article, in positions other than at the beginning of sentences. If at least 75% of these occurrences are capitalized, then e is a named entity.

The combined heuristics extract close to half a million named entities from Wikipedia. The second step constructs the actual dictionary D as follows:

- The set of entries in D consists of all strings that may denote a named entity, i.e. if $e \in E$ is a named entity, then its title name $e.title$, its redirect names $e.R$, and its disambiguation names $e.D$ are all added as entries in D.

- Each entry string $d \in D$ is mapped to $d.E$, the set of entities that d may denote in Wikipedia. Consequently, a named entity e is included in $d.E$ if and only if $d = e.title$, $d \in e.R$, or $d \in e.D$.

4 Named Entity Disambiguation

As illustrated in Section 1, the same proper name may refer to more than one named entity. The named entity dictionary from Section 3 and the hyperlinks from Wikipedia articles provide a dataset of disambiguated occurrences of proper names, as described in the following. As shown in Section 2.4, each link contains the title name of an entity, and the proper name (the display string) used to refer to it. We use the term *query* to denote the occurrence of a proper name inside a Wikipedia article. If there is a dictionary entry matching the proper name in the query q such that the set of denoted entities $q.E$ contains at least two entities, one of them the true answer entity $q.e$, then the query q is included in the dataset. More exactly, if $q.E$ contains n named entities e_1, e_2, ..., e_n, then the dataset will be augmented with n pairs $\langle q, e_k \rangle$ represented as follows:

$$\langle q, e_k \rangle = [\delta(e_k, q.e) \mid q.T \mid e_k.title]$$

The field $q.T$ contains all words occurring in a limit length window centered on the proper name. The window size is set to 55, which is the value that was observed to give optimum performance in the related task of cross-document coreference (Gooi and Allan, 2004). The Kronecker delta function $\delta(e_k, q.e)$ is 1 when e_k is the same as the entity $q.e$ referred in the link. Table 2 lists the query pairs created for the three *John Williams* queries from Section 1.1, assuming only three entities in Wikipedia correspond to this name.

δ	**Query Text**	**Entity Title**
1	Boston Pops conduct ...	John_Williams_(composer)
0	Boston Pops conduct ...	John_Williams_(wrestler)
0	Boston Pops conduct ...	John_Williams_(VC)
1	lost Taipei match ...	John_Williams_(wrestler)
0	lost Taipei match ...	John_Williams_(composer)
0	lost Taipei match ...	John_Williams_(VC)
1	won Victoria Cross ...	John_Williams_(VC)
0	won Victoria Cross ...	John_Williams_(composer)
0	won Victoria Cross ...	John_Williams_(wrestler)

Table 2: Disambiguation dataset.

The application of this procedure on Wikipedia results into a dataset of 1,783,868 disambiguated queries.

4.1 Context-Article Similarity

Using the representation from the previous section, the name entity disambiguation problem can be cast as a ranking problem. Assuming that an appropriate scoring function $score(q, e_k)$ is available, the named entity corresponding to query q is defined to be the one with the highest score:

$$\hat{e} = arg \max_{e_k} score(q, e_k) \qquad (1)$$

If $\hat{e} = q.e$ then $\hat{e}$ represents a hit, otherwise $\hat{e}$ is a miss. Disambiguation methods will then differ based on the way they define the scoring function. One ranking function that is evaluated experimentally in this paper is based on the cosine similarity between the context of the query and the text of the article:

$$score(q, e_k) = cos(q.T, e_k.T) = \frac{q.T}{\|q.T\|} \frac{e_k.T}{\|e_k.T\|}$$

The factors $q.T$ and $e_k.T$ are represented in the standard vector space model, where each component corresponds to a term in the vocabulary, and the term weight is the standard $tf \times idf$ score (Baeza-Yates and Ribeiro-Neto, 1999). The vocabulary V is created by reading all Wikipedia

articles and recording, for each word stem w, its document frequency $df(w)$ in Wikipedia. Stopwords and words that are too frequent or too rare are discarded. A generic document d is then represented as a vector of length $|V|$, with a position for each vocabulary word. If $f(w)$ is the frequency of word w in document d, and N is the total number of Wikipedia articles, then the weight of word $w \in V$ in the $tf \times idf$ representation of d is:

$$d_w = f(w) \ln \frac{N}{df(w)} \quad (2)$$

4.2 Taxonomy Kernel

An error analysis of the cosine-based ranking method reveals that, in many cases, the pair $\langle q, e \rangle$ fails to rank first, even though words from the query context unambiguously indicate e as the actual denoted entity. In these cases, cue words from the context do not appear in e's article due to two main reasons:

1. The article may be too short or incomplete.
2. Even though the article captures most of the relevant concepts expressed in the query context, it does this by employing synonymous words or phrases.

The cosine similarity between q and e_k can be seen as an expression of the total degree of correlation between words from the context of query q and a given named entity e_k. When the correlation is too low because the Wikipedia article for named entity e_k does not contain all words that are relevant to e_k, it is worth considering the correlation between context words and the categories to which e_k belongs. For illustration, consider the two queries for the name *John Williams* from Figure 1.

To avoid clutter, Figure 1 depicts only two entities with the name *John Williams* in Wikipedia: the composer and the wrestler. On top of each entity, the figure shows one of their Wikipedia categories (*Film score composers* and *Professional wrestlers* respectively), together with some of their ancestor categories in the Wikipedia taxonomy. The two query contexts are shown at the bottom of the figure. In the context on the left, words such as *conducted* and *concert* denote concepts that are highly correlated with the *Musicians*, *Composers* and *Film score composers* categories. On the other hand, their correlation with other categories in Figure 1 is considerably lower. Consequently, a

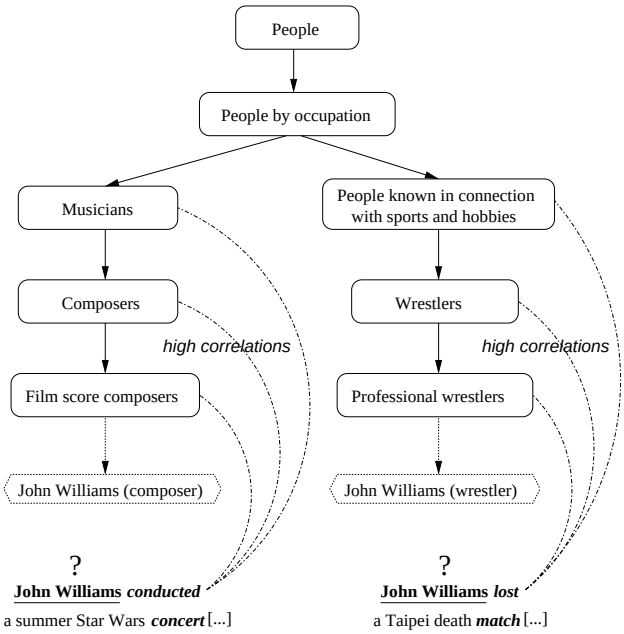

Figure 1: Word-Category correlations.

goal of this paper is to design a disambiguation method that 1) learns the magnitude of these correlations, and 2) uses these correlations in a scoring function, together with the cosine similarity. Our intuition is that, given the query context on the left, such a ranking function has a better chance of ranking the "composer" entity higher than the "wrestler" entity, when compared with the simple cosine similarity baseline.

We consider using a linear ranking function as follows:

$$\hat{e} = arg\max_{e_k} \mathbf{w}\ \Phi(q, e_k) \quad (3)$$

The feature vector $\Phi(q, e_k)$ contains a dedicated feature ϕ_{cos} for cosine similarity, and $|V| \times |C|$ features $\phi_{w,c}$ corresponding to combinations of words w from the Wikipedia vocabulary V and categories c from the Wikipedia taxonomy C:

$$\phi_{cos}(q, e_k) = cos(q.T, e_k.T) \quad (4)$$

$$\phi_{w,c}(q, e_k) = \begin{cases} 1 & \text{if } w \in q.T \text{ and } c \in e_k.C, \\ 0 & \text{otherwise.} \end{cases}$$

The weight vector $\mathbf{w}$ models the magnitude of each word-category correlation, and can be learned by training on the query dataset described at the beginning of Section 4. We used the kernel version of the large-margin ranking approach from (Joachims, 2002) which solves the optimization

problem in Figure 2. The aim of this formulation is to find a weight vector $\mathbf{w}$ such that 1) the number of ranking constraints $\mathbf{w}\ \Phi(q, q.e) \geq \mathbf{w}\ \Phi(q, e_k)$ from the training data that are violated is minimized, and 2) the ranking function $\mathbf{w}\ \Phi(q, e_k)$ generalizes well beyond the training data.

minimize:
$V(w, \xi) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum \xi_{q,k}$
subject to:
$\mathbf{w}\ (\Phi(q, q.e) - \Phi(q, e_k)) \geq 1 - \xi_{q,k}$
$\xi_{q,k} \geq 0$
$\forall q, \forall e_k \in q.E - \{q.e\}$

Figure 2: Optimization problem.

C is a parameter that allows trading-off margin size against training error. The number of linear ranking constraints is $\sum_q (|q.E| - 1)$. As an example, each of the three queries from Table 2 generates two constraints.

The learned $\mathbf{w}$ is a linear combination of the feature vectors $\Phi(q, e_k)$, which makes it possible to use kernels (Vapnik, 1998). It is straightforward to show that the dot product between two feature vectors $\Phi(q, e_k)$ and $\Phi(q', e'_k)$ is equal with the product between the number of common words in the contexts of the two queries and the number of categories common to the two named entities, plus the product of the two cosine similarities. The corresponding ranking kernel is:

$$K(\langle q, e_k\rangle, \langle q', e'_k\rangle) = |q.T \cap q'.T| \cdot |e_k.C \cap e'_k.C| + cos(q.T, e_k.T) \cdot cos(q'.T, e'_k.T)$$

To avoid numerical problems, the first term of the kernel is normalized and the second term is multiplied with a constant $\alpha = 10^8$:

$$K(\langle q, e_k\rangle, \langle q', e'_k\rangle) = \frac{|q.T \cap q'.T|}{\sqrt{|q.T| \cdot |q'.T|}} \cdot \frac{|e_k.C \cap e'_k.C|}{\sqrt{|e_k.C| \cdot |e'_k.C|}} + \alpha \cdot cos(q.T, e_k.T) \cdot cos(q'.T, e'_k.T)$$

In (McCallum et al., 1998), a statistical technique called *shrinkage* is used in order to improve the accuracy of a naive Bayes text classifier. Accordingly, one can take advantage of a hierarchy of classes by combining parameter estimates of parent categories into the parameter estimates of a child category. The taxonomy kernel is very related to the same technique – one can actually regard it as a distribution-free analogue of shrinkage.

4.3 Detecting Out-of-Wikipedia Entities

The two disambiguation methods discussed above (Sections 4.1 and 4.2) implicitly assume that Wikipedia contains all entities that may be denoted by entries from the named entity dictionary. Taking for example the name *John Williams*, both methods assume that in any context, the referred entity is among the 22 entities listed on the disambiguation page in Wikipedia. In practice, there may be contexts where *John Williams* refers to an entity e_{out} that is not covered in Wikipedia, especially when e_{out} is not a popular entity. These *out-of-Wikipedia* entities are accommodated in the ranking approach to disambiguation as follows. A special entity e_{out} is introduced to denote any entity not covered by Wikipedia. Its attributes are set to null values (e.g., the article text $e_{out}.T = \emptyset$, and the set of categories $e_{out}.C = \emptyset$). The ranking in Equation 1 is then updated so that it returns the Wikipedia entity with the highest score, if this score is greater then a fix threshold τ, otherwise it returns e_{out}:

$$e_{max} = arg\max_{e_k} score(q, e_k)$$
$$\hat{e} = \begin{cases} e_{max} & \text{if } score(q, e_{max}) > \tau, \\ e_{out} & \text{otherwise.} \end{cases}$$

If the scoring function is implemented as a weighted combination of feature functions, as in Equation 3, then the modification shown above results into a new feature ϕ_{out}:

$$\phi_{out}(q, e_k) = \delta(e_k, e_{out}) \quad (5)$$

The associated weight τ is learned along with the weights for the other features (as defined in Equation 5).

5 Experimental Evaluation

The taxonomy kernel was trained using the SVM^{light} package (Joachims, 1999). As described in Section 4, through its hyperlinks, Wikipedia provides a dataset of 1,783,868 ambiguous queries that can be used for training a named entity disambiguator. The apparently high number of queries actually corresponds to a moderate size dataset, given that the space of parameters includes one parameter for each word-category combination. However, assuming SVM^{light} does not run out of memory, using the entire dataset for training and testing is extremely

	# CAT.	TRAINING DATASET			TEST DATASET		# SV	TK(A)	Cos(A)
		QUERIES	PAIRS $\langle q, e_k \rangle$	# CONSTR.	QUERIES	PAIRS $\langle q, e_k \rangle$			
S_1	110	12,288	39,880	27,592	48,661	147,165	19,693	**77.2%**	61.5%
S_2	540	17,970	55,452	37,482	70,468	235,290	29,148	**68.4%**	55.8 %
S_3	2,847	21,185	64,560	43,375	75,190	261,723	36,383	**68.0%**	55.4%
S_4	540	38,726	102,553	63,827	80,386	191,227	35,494	**84.8%**	82.3%

Table 3: Scenario statistics and comparative evaluation.

time consuming. Therefore, we decided to evaluate the taxonomy kernel under the following scenarios:

■ **[**S_1**]** The working set of Wikipedia categories C_1 is restricted to only the 110 top level categories under *People by occupation*. The query dataset used for training and testing is reduced to contain only ambiguous queries $\langle q, e_k \rangle$ for which any potential matching entity e_k belongs to at least one of the 110 categories (i.e. $e_k.C \cap C_1 \neq \emptyset$). The set of negative matching entities e_k is also reduced to those that differ from the true answer e in terms of their categories from C_1 (i.e. $e_k.C \cap C_1 \neq e.C \cap C_1$). In other words, this scenario addresses the task of disambiguating between entities with different top-level categories under *People by occupation*.

■ **[**S_2**]** In a slight generalization of **[**S_1**]**, the set of categories C_2 is restricted to all categories under *People by occupation*. Each category must have at least 200 articles to be retained,which results in a total of 540 categories out of the 8202 categories under *People by occupation*. The query dataset is generated as in the first scenario by replacing C_1 with C_2.

■ **[**S_3**]** This scenario is similar with **[**S_2**]**, except that each category has to contain at least 20 articles to be retained, leading to 2847 out of 8202 categories.

■ **[**S_4**]** This scenario uses the same categories as $[S_2]$ (i.e. $C_4 = C_2$). In order to make the task more realistic, all queries from the initial Wikipedia dataset are considered as follows. For each query q, out of all matching entities that do not have a category under *People by occupation*, one is randomly selected as an *out-of-Wikipedia* entity. Then, out of all queries for which the true answer is an *out-of-Wikipedia* entity, a subset is randomly selected such that, in the end, the number of queries with *out-of-Wikipedia* true answers is 10% of the total number of queries. In other words, the scenario assumes the task is to detect if a name denotes an entity belonging to the *People by occupation* taxonomy and, in the positive cases, to disambiguate between multiple entities under *People by occupation* that have the same name.

The dataset for each scenario is split into a training dataset and a testing dataset which are disjoint in terms of the query names used in their examples. For instance, if a query for the name *John Williams* is included in the training dataset, then all other queries with this name are allocated for learning (and consequently excluded from testing). Using a disjoint split is motivated by the fact that many Wikipedia queries that have the same true answer also have similar contexts, containing rare words that are highly correlated, almost exclusively, with the answer. For example, query names that refer to singers often contain album or song names, query names that refer to writers often contain book names, etc. The taxonomy kernel can easily "memorize" these associations, especially when the categories are very fine-grained. In the current framework, the unsupervised method of context-article similarity does not utilize the correlations present in the training data. Therefore, for the sake of comparison, we decided to prohibit the taxonomy kernel from using these correlations by training and testing on a disjoint split. Section 6 describes how the training queries could be used in the computation of the context-article similarity, which has the potential of boosting the accuracy for both disambiguation methods.

Table 3 shows a number of relevant statistics for each scenario: #CAT represents the number of Wikipedia categories, #SV is the number of support vectors, TK(A) and Cos(A) are the accuracy of the Taxonomy Kernel and the Cosine similarity respectively. The training and testing datasets are characterized in terms of the number of queries and query-answer pairs. The number of ranking contraints (as specified in Figure 2) is also included for the training data in column #CONSTR.

The size of the training data is limited so that learning in each scenario takes within three days on a Pentium 4 CPU at 2.6 GHz. Furthermore,

in S_4, the termination error criterion ϵ is changed from its default value of 0.001 to 0.01. Also, the threshold τ for detecting *out-of-Wikipedia* entities when ranking with cosine similarity is set to the value that gives highest accuracy on training data.

As can be seen in the last two columns, the Taxonomy Kernel significantly outperforms the Cosine similarity in the first three scenarios, confirming our intuition that correlations between words from the query context and categories from Wikipedia taxonomy provide useful information for disambiguating named entities. In the last scenario, which combines detection and disambiguation, the gain is not that substantial. Most queries in the corresponding dataset have only two possible answers, one of them an *out-of-Wikipedia* answer, and for these cases the cosine is already doing well at disambiguation. We conjecture that a more significant impact would be observed if the dataset queries were more ambiguous.

6 Future Work

The high number of support vectors – half the number of query-answer pairs in training data – suggests that all scenarios can benefit from more training data. One method for making this feasible is to use the weight vector w explicitely in a linear SVM. Because much of the computation time is spent on evaluating the decision function, using w explicitely may result in a significant speed-up. The dimensionality of w (by default $|V| \times |C|$) can be reduced significantly by considering only word-category pairs whose frequency in the training data is above a predefined threshold.

A complementary way of using the training data is to augment the article of each named entity with the contexts from all queries for which this entity is the true answer. This method has the potential of improving the accuracy of both methods when the training and testing datasets are not disjoint in terms of the proper names used in their queries.

Word-category correlations have been used in (Ciaramita et al., 2003) to improve word sense disambiguation (WSD), although with less substantial gains. There, a separate model was learned for each of the 29 ambiguous nouns from the Senseval 2 lexical sample task. While creating a separate model for each named entity is not feasible – there are 94,875 titles under *People by occupation* – named entity disambiguation can nevertheless benefit from correlations between Wikipedia categories and features traditionally used in WSD such as bigrams and trigrams centered on the proper name occurrence, and syntactic information.

7 Conclusion

We have presented a novel approach to named entity detection and disambiguation that exploited the untapped potential of an online encyclopedia. Experimental results show that using the Wikipedia taxonomy leads to a substantial improvement in accuracy. The application of the new named entity disambiguation method holds the promise of better results to popular web searches.

8 Acknowledgments

We would like to thank Peter Dienes, Thorsten Joachims, and the anonymous reviewers for their helpful comments.

References

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. ACM Press, New York.

Massimiliano Ciaramita, Thomas Hofmann, and Mark Johnson. 2003. Hierarchical semantic classification: Word sense disambiguation with world knowledge. In *The 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.

Robert Dale. 2003. Computational linguistics. *Special Issue on the Web as a Corpus*, 29(3), September.

Gottlob Frege. 1999. On sense and reference. In Maria Baghramian, editor, *Modern Philosophy of Language*, pages 3–25. Counterpoint Press.

Chung Heong Gooi and James Allan. 2004. Cross-document coreference on a large scale corpus. In *Proceedings of Human Language Technology Conference / North American Association for Computational Linguistics Annual Meeting*, Boston, MA.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Andrew McCallum, R. Rosenfeld, Tom Mitchell, and A. Y. Ng. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98)*, pages 359–367, Madison, WI.

M. Remy. 2002. Wikipedia: The free encyclopedia. *Online Information Review*, 26(6):434. www.wikipedia.org.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons.