Name Discrimination by Clustering Similar Contexts

Ted Pedersen[1], Amruta Purandare[2], and Anagha Kulkarni[1]

[1] University of Minnesota, Duluth, MN 55812, USA
[2] University of Pittsburgh, Pittsburgh, PA 15260, USA
http://senseclusters.sourceforge.net

Abstract. It is relatively common for different people or organizations to share the same name. Given the increasing amount of information available online, this results in the ever growing possibility of finding misleading or incorrect information due to confusion caused by an ambiguous name. This paper presents an unsupervised approach that resolves name ambiguity by clustering the instances of a given name into groups, each of which is associated with a distinct underlying entity. The features we employ to represent the context of an ambiguous name are statistically significant bigrams that occur in the same context as the ambiguous name. From these features we create a co–occurrence matrix where the rows and columns represent the first and second words in bigrams, and the cells contain their log–likelihood scores. Then we represent each of the contexts in which an ambiguous name appears with a second order context vector. This is created by taking the average of the vectors from the co–occurrence matrix associated with the words that make up each context. This creates a high dimensional "instance by word" matrix that is reduced to its most significant dimensions by Singular Value Decomposition (SVD). The different "meanings" of a name are discriminated by clustering these second order context vectors with the method of Repeated Bisections. We evaluate this approach by conflating pairs of names found in a large corpus of text to create ambiguous pseudo-names. We find that our method is significantly more accurate than the majority classifier, and that the best results are obtained by having a small amount of local context to represent the instance, along with a larger amount of context for identifying features, or vice versa.

1 Introduction

The problem of name ambiguity exists in many forms. It is common for different people to share the same name. For example, there is a George Miller who is a prominent Professor of Psychology, another who is a Congressman from California, and two more who are film directors from Australia. Locations may have the same name. For example, Duluth is a city in Minnesota and also a city in Georgia. The acronyms associated with organizations may also be ambiguous. UMD can refer to the University of Michigan – Dearborn, the University of Minnesota, Duluth or the University of Maryland .

The effects of name ambiguity can be seen when carrying out web searches or retrieving articles from an archive of newspaper text. For example, the top 10 hits of a Google search for "George Miller" mention five different people. While it may be clear to a human that the Congressman from California, the Professor from Princeton, and the director of the film *Mad Max* are not the same person, it is difficult for a computer program to make the same distinction. In fact, a human may have a hard time organizing this information such that they find all the material relevant to the particular person they are interested in.

The problem of grouping occurrences of a name based on the underlying entity's identity can be approached using techniques developed for word sense discrimination. This is the process of examining a number of sentences that contain a given polysemous word, and then grouping those instances based on the meaning of that word. Note that this is distinct from word sense disambiguation, which is the process of assigning a sense to a polysemous word from a predefined set of possibilities, usually defined by a dictionary or some other well established resource. However, it is not likely that we will have a complete inventory of the possible identities associated with each name, so our immediate objective is to group the occurrences of a name into clusters based on the underlying identity. We are currently developing methods that will examine the content of each cluster to automatically create a descriptive label that will identify the entity represented. This paper is only concerned with discriminating among the different entities, while the labeling step is an area of ongoing work for us.

Approaches to word sense discrimination generally rely on the strong contextual hypothesis of Miller and Charles [10], who hypothesize that words with similar meanings are often used in similar contexts. This is equally true for names, where a particular entity will likely be mentioned in certain contexts. For example, George Miller the film director may not be mentioned with Princeton University very often, while George Miller the Professor will be. Thus, our approach to name discrimination reduces to the problem of finding classes of similar contexts such that each class represents a distinct entity. In other words, contexts that are grouped together in the same class represent a particular entity.

In this paper we show how the unsupervised word sense discrimination methods of Purandare and Pedersen (e.g., [12], [13]) can be applied to the problem of name discrimination. We begin with a summary of related work on the problem of name discrimination, and then describe our approach, which is based on clustering second-order context vectors whose dimensions have been reduced by Singular Value Decomposition (SVD). We present an evaluation of our approach based on pseudo-names that we create by conflating two related names in a large corpus of newswire text.

2 Related Work

The problem of name discrimination is a natural extension to work that identifies named entities in text. This was shown in early work by Wacholder, et. al. [15], who developed an integrated approach to identifying named entities and resolv-

ing any ambiguities that might be present based on knowledge of co–occurring names in the context, and a database of known names.

Cross document co–reference resolution is closely related to name discrimination, in that it seeks to resolve referents across multiple documents. There are several variations to this problem. For example, there may be multiple forms of the same name (*J. Smith* and *John Smith* and *Mr. Smith*), or there may be titles, pronouns, etc. that refer to an entity (*J.Smith, the President, him*). We focus on the more specific problem of identifying which entities the particular form of a name refer to. For example, *John Smith* may be mentioned in 30 documents. Our objective is to determine how many different individuals this entails. While we do not explicitly find chains of references, in fact this would be easy to reconstruct from our results since each occurrence of a name will appear in a cluster. All of the members of a single cluster can then be considered to form a chain of references.

Bagga and Baldwin [1] propose a method based on creating first order context vectors that represent each instance in which an ambiguous name occurs. Each vector contains exactly the words that occur within a 55 word window around the ambiguous name, and the similarity among names is measured using the cosine measure. In order to evaluate their approach, they created the *John Smith* corpus, which consists of 197 articles from the New York Times that mention 35 different *John Smiths*.

Gooi and Allan [5] present a comparison of Bagga and Baldwin's approach to two variations of their own. They used the *John Smith* Corpus, and created their own corpus which is called the *Person-X* corpus. Since it is rather difficult to obtain large samples of data where the actual identity of a truly ambiguous name is known, the *Person-X* corpus consists of pseudo-names that are ambiguous. These are created by disguising known names as *Person-X*, thereby introducing ambiguities. There are 34,404 mentions of *Person-X*, which refer to 14,767 distinct underlying entitles. Gooi and Allan re–implement Bagga and Baldwin's context vector approach, and compare it to another context vector approach that groups vectors together using agglomerative clustering. They also group instances together based on the Kullback–Liebler Divergence. Their conclusion is that the agglomerative clustering technique works particularly well.

Mann and Yarowsky [9] have proposed an approach for disambiguating personal names using a Web based unsupervised clustering technique. They rely on a rich feature space of biographic facts, such as date or place of birth, occupation, relatives, collegiate information, etc. A seed fact pair (e.g., Mozart, 1776), is queried on the Web and the sentences returned as search results are used to generate the patterns which are than used to extract the biographical information from the data. Once these features are extracted clustering follows. Each instance of an ambiguous name is assigned a vector of extracted features, and at each stage of cluster the two most similar vectors are merged together to produce a new cluster. This step is repeated until all the references to be disambiguated are clustered.

There has also been work on name disambiguation using supervised learning approaches in a number of different domains. These approaches rely on having some number of examples available, where the underlying entity for an ambiguous name is known prior to learning.

For example Han et. al. [6] address the problem of resolving ambiguity in bibliography entries, such as *J. Smith* versus *John Smith* versus *J.Q. Smith*. They rely on the use of co–occurrence relations among the names. For example, if *J.Q. Smith* and *Johnny Smith* both wrote articles with *H. L. Hutton*, then they might conclude that *J.Q.* and *Johnny* are one in the same. They compare the use of Naive Bayesian classifiers and Support Vector Machines, and conclude that both methods are effective in certain circumstances.

Name disambiguation is also a problem in the medical domain. For example, Hatzivassiloglou, et. al. [7] point out that genes and proteins often share the same name, and that it's important to be able to identify which is which. They employ a number of well known word sense disambiguation techniques and achieve excellent results. Ginter, et. al. [4] develop an algorithm for disambiguation of protein names based on weighted features vectors derived from surface lexical features and achieve equally good results.

3 Discrimination by Clustering Similar Contexts

Purandare and Pedersen (e.g., [12], [13]) have developed methods of clustering multiple occurrences of a given word into senses based on their contextual similarity. In this paper we adapt those techniques to the problem of name discrimination.

We begin by collecting some number of instances of an ambiguous name. Each instance consists of approximately 50 words, where the ambiguous name is found in the center of the context.

Then we identify significant bigrams in the contexts to be clustered[1]. A bigram is a sequence of two words that may or may not be adjacent. In our work we generally allow bigrams to be an ordered pair of non–consecutive words and permit one intermediate word between them, or bigrams with a window size of 3. A bigram is judged significant by measuring the log–likelihood ratio between the two words. If that score is greater than 3.814 then the bigram is significant and selected as a feature. Note that we employ a technique known as *OR stop–listing* and remove any bigram that is made up of one or two stop–words. Thus, the bigrams we select are made up of two content words.

We build a matrix based on the set of significant bigrams that we identify. The rows in this matrix represent the first word in the bigram, and the columns represent the second word. Each cell in the matrix contains the log–likelihood ratio associated with the bigram represented by the row and column. Thus, each row of this matrix can be viewed as a word vector made up of log–likelihood

[1] It would be possible to identify these features in a separate large corpus of training data. However, in this work we are identifying the features in the instances that are to be clustered.

ratios, where the word is represented by words with which it co–occurs. Since the bigrams are ordered, this matrix is not symmetric. This matrix is also very sparse, since many words that form bigrams only occur with a small number of other words.

Because of it's large size and sparsity, we employ Singular Value Decomposition (SVD) to reduce the dimensionality. We reduce the matrix to 10% of it's original number of columns, or 300 columns, whichever is least. Thus, any matrix of 3,000 or more columns will be reduced to 300 columns, while those less than 3,000 columns are reduced to 10% of their number of columns. Note that SVD reduces the number of columns, but not the number of rows. The reduction has two effects. First, it acts as a smoothing operation, where the resulting matrix will have very few (if any) zero values. Second, it has the effect of reducing the words that make up the columns from a word level feature space into a concept level semantic space.

The SVD reduced bigram matrix is used to create *second order context vectors* ([14]) that will represent the instances to be clustered. Each word in the context of the ambiguous name that has a row vector in the SVD reduced matrix will be represented by that vector. All the vectors associated with the context that are found in the SVD reduced matrix are averaged together to create an overall representation of the context.

The use of SVD and the averaging of word vectors to create a second order context representation has been employed by Schütze ([14]) in the context of word sense discrimination research, and in Latent Semantic Analysis [8], and Latent Semantic Indexing [2]. Our approach is certainly related to this, although our use of bigram features and the log–likelihood scores makes it somewhat distinct, since the usual technique is to create a word co–occurrence matrix that employs frequency counts.

The general intuition behind the second order representation is that it captures indirect relationships between words. For example, suppose that the word *shoot* forms significant bigrams with the words *murder*, *bullets*, and *weapon*, and that *gun* forms significant bigrams with *fire*, *bullets*, and *murder*. Our intuitive understanding that *shoot* and *gun* are related is confirmed by the shared second order relationships they have with *murder* and *bullets*.

4 Clustering

Once the instances to be discriminated are represented by second order context vectors, they are clustered such that the instances that are similar to each other are placed into the same cluster.

Clustering algorithms are typically classified into three main categories: hierarchical, partitional, and hybrid. It is generally believed that the quality of clustering by partitional algorithms such as k–means is inferior to that of the agglomerative methods such as average link. However, a recent study by Zhao and Karypis [16] has suggested that these conclusions are based on experiments

conducted with smaller data sets, and that with larger data sets partitional algorithms are not only faster but lead to better results.

In particular, Zhao and Karypis recommend a hybrid approach known as Repeated Bisections. This overcomes the main weakness with partitional approaches, which is the instability in clustering solutions due to the choice of the initial random centroids. Repeated Bisections starts with all instances in a single cluster. At each iteration it selects one cluster whose bisection optimizes the given criteria function. The cluster is bisected using standard K-means method with K=2, while the criteria function maximizes the similarity between each instance and the centroid of the cluster to which it is assigned. As such this is a hybrid method that combines a hierarchical divisive approach with partitioning.

5 Experimental Data

Our experimental data is made up of six pairs of pseudo-names that are generated by identifying pairs of names that occur in a large corpus of newswire text. Six pairs of names were selected that represent different frequency distributions and types of names. Once selected, all of the instances associated with each pair were extracted from the corpus and placed in separate files (one file per pair). Each instance consists of approximately 25 words to the left and right of the ambiguous name. After the pairs were extracted, they were conflated in each file by creating an obfuscated form of the name that is used in place of both names. For example, one of our pairs was "David Beckham" and "Ronaldo". All of the instances in the corpus that included either name were extracted, and then all occurrences of both name were replaced with the obfuscated form "RoBeck". Discrimination is then carried out in a completely unsupervised way, meaning that we don't use the knowledge of the correct name until evaluation.

The corpus employed in these experiments is the Agence France Press English Service (AFE) portion of the GigaWord English Corpus, as distributed by the Linguistic Data Consortium. The AFE corpus consists of 170,969,000 words of English text which appeared in the AFE newswire from May 1994 to May 1997, and from December 2001 until June 2002. In all this represents approximately 1.2 GB of text (uncompressed).

The pairs of names we selected and their frequency of occurrence are shown in Table 5. This also shows the combined frequency of the pseudo-name, and the percentage which the more common of the two names occurs in reality. This last value represents the majority class, and is the level of accuracy that a baseline clustering algorithm could achieve by simply placing all instances in one cluster.

These pairs were selected to try and force our methods to make relatively fine grained distinctions between the words/senses that make up the pair. One known drawback of pseudo–words arises when the component words are randomly selected. In such a case, it is very likely that the two senses represented will be quite distinct ([3]). We have adopted a solution to this problem that is somewhat similar to that of Nakov and Hearst [11], who suggest creating psuedo

words of words that are individually unambiguous, and yet still related in some way.

For example, we make distinctions between two soccer players (RoBeck), an ethnic group and a diplomat (JikRol), two computer companies (MSIBM), two political leaders (MonSlo), a nation and a nationality (JorGypt), and two countries (JapAnce). Note that our task has now become finding the original and correct name that was in the corpus before it was obfuscated. In general the names we have selected have only one underlying entity, for example, "David Beckham" always refers to the soccer player, and Microsoft always refers to the software company. However, "Jordan" is an exception. The dominant sense is that of the country (given the nature of the news wire text) but there are also occurrences of the famous American basketball player. This may well have an impact on the results of clustering, which we will discuss in our analysis.

Table 1. Conflated Pairs of Names

Name1	Count1	Name2	Count2	Conflated	Total	Majority
Ronaldo	1,652	David Beckham	740	RoBeck	2,452	69.3%
Tajik	3,002	Rolf Ekeus	1,071	JikRol	4,073	73.7%
Microsoft	3,401	IBM	2,406	MSIBM	5,807	58.6%
Shimon Peres	7,846	Slobodan Milosevic	6,176	MonSlo	13,734	56.0%
Jordan	25,539	Egyptian	21,762	JorGypt	46,431	53.9%
Japan	118,712	France	112,357	JapAnce	231,069	51.4%

Each pair of words is processed separately, so we are making a 2 class distinction in this study. In future work we will conflate larger number of names so that we are making distinctions between more underlying entities.

The two clusters are evaluated by replacing the conflated form of the word with the correct original, and determining which name should be assigned to which cluster in order to maximize accuracy. This can be thought of as similar (but not exactly equivalent) to measuring the purity of the clusters. We can find the maximum accuracy by considering the results (once the known identities are available) as a two-by-two cross classification table, that shows the distribution of names and clusters. An example is shown in Figure 5. Each row represents the distribution of the instances in the clusters as compared to their actual identity, and each column shows the distribution of the actual identities in the clusters. We can find the assignment of clusters to identities that maximizes the accuracy by simply reordering the columns of the matrix such that the main diagonal sum is maximized.

From Figure 5, we can see that the assignment of Peres to C1, and Milosevic to C2, results in an accuracy of 91.4% ((6,573 + 6,012)/13,734), while an assignment of Peres to C2 and Milosevic to C1 results in accuracy of 8.6% (36 + 1,149)/13,734).

We measure the precision and recall based on the maximally accurate assignment of names to clusters. Precision is defined as the number of instances

	Milosevic	Peres	
C1	36	6,537	6,573
C2	6,012	1,149	7,161
	6,048	7,686	13,734

	Peres	Milosevic	
C1	6,573	36	6,573
C2	1,149	6,012	7,161
	7,648	6,048	13,734

Fig. 1. Assigning Cluster to Name

that are clustered correctly divided by the number of instances clustered, while recall is the number of instances clustered correctly over the total number of instances[2]. From these values we compute the F–measure, which is two times the product of precision and recall, divided by the sum of precision and recall.

6 Experimental Methodology

There are several significant issues that determine how accurate this approach can be. First, we must determine the size of the context around the ambiguous name to be clustered. We refer to this as the *test scope*. This size of the test scope determines how many words make up the averaged vector that represents the context. Note that when we set our test scope to a value of N, it means use all of the words within N positions of the target word on both sides that have a row vector associated with them in the SVD reduced bigram matrix.

A small test scope is predicated on the idea that the words nearest the ambiguous name will be the most important indicators of how it should be clustered. For example, in the case of names of people, titles or affiliations might be located in close proximity. However, a larger test scope brings in more context, and allows for more content to be included in the averaged vector, potentially making it possible to make finer grained distinctions.

As there are good arguments in favor of both approaches, we will experiment with test scopes of 5 and 20, where a test scope of N means represent the context with the average of all the vectors found for words within N positions to the left and right of the ambiguous name.

The *training scope* is also a significant factor. This determines how large a context around the ambiguous name will be used for identifying the bigram features. If the training scope is set to N, it means that we restrict consideration of bigrams to those that occur within N positions of the ambiguous name.

A smaller training scope will focus the search for bigrams on those that are near or include the ambiguous name (in the case of one word names). This can result in a small number of very reliable collocational features. However, a larger training scope may find bigrams related to the identity of the ambiguous name that do not necessarily include the name itself.

Again, since there are interesting possibilities with both larger and smaller training scopes, we will run experiments with that scope set to 5 and 30.

[2] The clustering algorithm that we use has the option of not placing an instance in any cluster, which is why precision and recall may differ.

Note that in this experiment the test and training data are the same. We use the training data for feature identification, and then the test data is what we use to determine how we build the second order context representation.

In addition, we hypothesize that the potential role of SVD is unclear. The second order co–occurrence features already help to represent indirect relationships, so it's not clear that the smoothing and identification of principle dimensions done by SVD adds significantly to the results.

In all cases we used bigram features and selected those by taking all bigrams that occurred 5 or more times, and had an associated log-likelihood score of 3.814 or above. We used a standard stop–list of function words, and discard any bigram as a feature if it consists of 1 or 2 stop words.

7 Experimental Results and Discussion

For each of our six pseudo-names, we run eight different experiments. We run all possible combinations of experiments where the test scope is set to 5 and 20, the training scope is set to 5 and 20, and we may or may not use SVD.

We show the results of all eight experiments for each conflated pseudo-word in Table 7. This table shows the F-measure for each combination of settings, and also provides general information about the conflated word such as the total number of instances to be clustered, and the percentage of those that belong to the majority identity. Remember that this value can serve as a lower bound for these approaches, since a method that placed every instance in a single cluster would attain this level of accuracy.

For smaller samples of data, we observe that SVD will at times offer an improvement, but in general does not lead to significant improvements. RoBeck (test=5, training=5) is one case where SVD offers a significant improvement, from 57.3 to 78.4. Given this relatively small amount of data (using small windows for both test and training purposes) the resulting bigram vector is very sparse, and using SVD helps to smooth that out and make it possible to still draw distinctions between contexts.

We note that SVD shows a benefit for those psuedo-names with neither a very large nor a very small number of instances. For example, it results in an improvement for 3 of 4 cases for MonSlo, and all 4 cases for JorGypt. However, JapAnce shows no such improvement, and in fact the overall results are somewhat disappointing in that they are less than the majority sense. We hypothesized that the very large size (more than 200,000 instances) of the data may have had a negative impact, but upon reducing the size of the experiment to 40,000 instances we found essentially identical results. Thus, we believe that this pair might represent a very hard sense distinction to make. While Japan and France are clearly distint geographically and culturally, it may be that they arise in so many different contexts in news text that there are no consistently strong discriminating features that can be identified. This remains an interesting issue for future exploration.

Table 2. F-Measures for Name Discrimination

			test scope 5		test scope 20	
			training scope		training scope	
			5	20	5	20
RoBeck	2,452 to cluster	no SVD	57.3	72.7	**85.9**	64.7
	(69.3) majority	SVD	78.4	71.0	81.9	64.9
JikRol	4,073 to cluster	no SVD	94.7	**96.2**	91.0	90.4
	(73.7) majority	SVD	90.9	93.5	87.2	89.3
MSIBM	5,807 to cluster	no SVD	47.7	51.3	**68.0**	60.0
	(58.6) majority	SVD	52.8	52.6	57.2	58.5
MonSlo	13,734 to cluster	no SVD	62.8	**96.6**	54.6	91.4
	(56.0) majority	SVD	80.0	91.4	82.2	94.2
JorGypt	46.431 to cluster	no SVD	56.6	59.1	57.0	53.0
	(53.9) majority	SVD	56.8	**62.2**	61.5	61.5
JapAnce	231,069 to cluster	no SVD	51.1	51.1	50.3	50.3
	(51.4) majority	SVD	51.1	51.1	50.3	50.3

The effect of the variations in the test and training scope are quite interesting. First, the best results for each pair of words came about by either using a small test scope with a large training scope (test = 5, train = 20) or a large test scope with a small training scope (test=20, train = 5). There was no case where the small scopes or large scopes alone gave the best results. We believe that this shows that the scopes are complementary.

A large test scope means that there are many words in the context that will be used to create the averaged vector. If those words are represented by a feature vector that is derived from a large training scope, then the combination of these two wide scopes leads to overly general information. However, if the training scope is small, then the words that occur in the context vector are all represented relative to words that are known to occur near the ambiguous name in the training data. A similar argument can be made for the case of a small test scope and a large training scope. The small test scope means that the averaged context vector will be made up of a small number of words that occur near the ambiguous noun. The words that make up the contexts may all be fairly distinct, but the co–occurrence information derived from a larger training scope will make it possible to identify them as being similar with other words in the test scope.

8 Future Work

The experiments in this paper all focus on binary distinctions, between two relatively distinct entities. We will extend these experiments in future to make distinctions among a larger number of underlying individuals. Rather than simply using pseudo-names, we will use the *John Smith* corpus as described in the work of Bagga and Baldwin, as well as the data used in the Mann and Yarowsky.

The use of this data will also introduce the other side of the name discrimination problem, that is in identifying two different names that refer to the same person (e.g., *Mr. Smith* and *John Smith*). Fortunately our techniques can be used without modification for this particular problem, and we are optimistic that they will perform well.

We are also developing techniques for looking at the content of the clusters to identify the entity associated with a particular cluster. We have experimented with identifying the most significant features in the clusters of text, and these provide very simple descriptive terms that might describe the entity. However, we wish to improve this approach to the point where it is more analogous to generating a summary of the text in the cluster, and thereby become a tool for knowledge discovery.

9 Conclusions

We have found that the method of Purandare and Pedersen for discriminating word senses by clustering similar contexts performs well in discriminating among ambiguous names. This is an unsupervised approach, so the fact that it nearly always out performs the majority baseline clustering method is significant. We observed that the test and training scopes are complementary, and should be set such that one is small and the other is large in order to get optimal results.

10 Acknowledgments

This research is supported by a National Science Foundation Faculty Early CAREER Development Award (#0092784).

All of the experiments in this paper were carried out with version 0.55 of the SenseClusters package, freely available from the URL shown on the title page.

References

1. A. Bagga and B. Baldwin. Entity–based cross–document co–referencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics*, pages 79–85. Association for Computational Linguistics, 1998.
2. 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*, 41:391–407, 1990.

3. T. Gaustad. Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. In *Companion Volume to the Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL/EACL 2001) – Proceedings of the Student Research Workshop*, pages 61–66, Toulouse, France, 2001.
4. F. Ginter, J. Boberg, J. Jrvine, and T. Salakoski. New techniques for disambiguation in natural language and their application to biological text. *Journal of Machine Learning Research*, 5:605–621, June 2004.
5. C. H. Gooi and J. Allan. Cross-document coreference on a large scale corpus. In S. Dumais, D. Marcu, and S. Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 9–16, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.
6. H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 2004 joint ACM/IEEE conference on Digital libraries*, pages 296–305, 2004.
7. V. Hatzivassiloglou, P. Duboue, and A. Rzhetsky. Disambiguating proteins, genes, and rna in text: A machine learning approach. In *Proceedings of the 9th International Conference on Intelligent Systems for Molecular Biology*, Tivoli Gardens, Denmark, July 2001.
8. T.K. Landauer, P.W. Foltz, and D. Laham. An introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284, 1998.
9. G. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 33–40. Edmonton, Canada, 2003.
10. G.A. Miller and W.G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.
11. P. Nakov and M. Hearst. Category-based pseudowords. In *Companion Volume to the Proceedings of HLT-NAACL 2003 - Short Papers*, pages 67–69, Edmonton, Alberta, Canada, May 27 - June 1 2003.
12. A. Purandare. Discriminating among word senses using McQuitty's similarity analysis. In *Companion Volume to the Proceedings of HLT-NAACL 2003 – Student Research Workshop*, pages 19–24, Edmonton, Alberta, Canada, May 27 - June 1 2003.
13. A. Purandare and T. Pedersen. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48, Boston, MA, 2004.
14. H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.
15. N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of proper names in text. In *Proceedings of the fifth conference on Applied natural language processing*, pages 202–208. Morgan Kaufmann Publishers Inc., 1997.
16. Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the 11th Conference of Information and Knowledge Management (CIKM)*, pages 515–524, 2002.