Discovering Identities in Web Contexts with Unsupervised Clustering

Ted Pedersen
Department of Computer Science
University of Minnesota
Duluth, MN 55812 USA
tpederse@d.umn.edu

Anagha Kulkarni
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
anaghak@cs.cmu.edu

Abstract

We describe the application of unsupervised clustering methodologies to the problem of discriminating among ambiguous names found in short passages of text that appear on Web pages. We show how to tailor these methods to handle the very noisy data that we typically find on the Web. We experiment with several variations in feature selection, two methods that automatically determine the number of clusters in the data, two different representations of the contexts to be discriminated, and with dimensionality reduction. Our evaluation is carried out using Web contexts for five different ambiguous names that were manually disambiguated to use as a gold standard.

1 Introduction

In this paper we evaluate the effectiveness of unsupervised methodologies that cluster short contexts of text based on their similarity. We apply these techniques to the problem of discriminating among named entities as found in Web pages. As the Web increases in coverage, there is a growing problem of ambiguity on Web pages, since different people or organizations can share the same name.

These techniques are based on the Distributional Hypothesis (e.g., [Harris, 1968], [Miller and Charles, 1991]) which holds that words that occur in similar contexts will tend to have similar meanings. Our approach is to cluster Web contexts that contain an ambiguous name such that each resulting cluster represents a particular entity. These contexts are approximately 100 word–long passages of text taken from Web pages, where an ambiguous name is located in the middle of the context.

These methods have previously been applied to discriminating among the meanings of ambiguous names or words, or grouping short contexts based on their topic. Specific examples of where these methods have already been applied include word sense discrimination (e.g., [Schütze, 1998],[Purandare and Pedersen, 2004]), email clustering (e.g., [Kulkarni and Pedersen, 2005]), and named entity discrimination (e.g.,[Pedersen *et al.*, 2005]).

These techniques are language independent (c.f., [Pedersen *et al.*, 2006]) and as such only rely on lexical features that can be identified in text. They do not incorporate any syntactic or linguistic information, nor do they utilize any manually created or maintained knowledge sources. As such they are ideal for Web contexts, which are often not well formed and include many strings that are not typically a part of knowledge bases or dictionaries. While our evaluation is done with English language texts, these methods can be applied to Web contexts in any other language.

The specific approaches we take are first order methods based on [Pedersen and Bruce, 1997], and second order methods that follow [Schütze, 1998]). These have been been implemented and extended in the freely available SenseClusters[1] system. In general, first order methods create vectors that represent the features that occur in the contexts to be clustered. Second order methods create word co-occurrence vectors that are used to represent the words in the contexts to be clustered, where each context is represented by the average of all of its word vectors.

While there are existing approaches that carry out word sense discrimination (e.g., [Schütze, 1998], [Purandare and Pedersen, 2004], [Levin *et al.*, 2006]), these have required that the user specify in advance the number of clusters to be discovered. This is a significant limitation, since in general a user will not know this number, and in fact discovering it might be a goal of the experiment in the first place. [Pedersen and Kulkarni, 2006] present a number of methods for automatically determining the number of clusters in short contexts. We explore the use of the Gap Statistic [Tibshirani *et al.*, 2001] and the PK2 measure for determining the number of clusters in Web contexts. Both of these measures are included in the SenseClusters system.

While we are using Web contexts as the basis of our experiments, our intent is that this serve as an example of very noisy data. Thus, our focus is on reducing the effect of such noise on an unsupervised discrimination process in any kind of text, and not specifically or exclusively on Web data. As such we do not take advantage of information that is unique to Web data, such as domain names, click–through behavior by users, or iteratively refined searches. There are approaches to carrying out name discrimination on the Web that do utilize such information to good effect (e.g., [Mann and Yarowsky, 2003]).

[1] http://senseclusters.sourceforge.net

This paper continues with an overview of the lexical features that we utilize to represent the Web contexts. We then describe the first and second order representations of context, and how dimensionality reduction may be employed. We go on to discuss how those contexts are clustered and evaluated. We introduce a corpus of Web contexts that have been manually disambiguated in order to evaluate our method, and then we describe the results of our experiments.

2 Lexical Features

At the heart of language–independent unsupervised approaches are lexical features. We identify our features from the Web contexts that are to be clustered. We also make use of stop–lists to exclude certain words from serving as features. In general a stop–word is a low–content word that occurs very frequently in the contexts, to the point where it does not provide any useful information in discriminating between the different identities.

The features we employ are either unigrams or bigrams. Unigrams are single words that occur more than a given number of times in the Web contexts, and are not present in the stop–list. Bigrams are ordered pairs of words that occur adjacent to each other in the Web contexts, where neither word may be a stop–word. Bigrams are selected by using measures of association that identify when these pairs of words have occurred more than would be expected by chance. In this paper we describe both Pointwise Mutual Information (PMI) and the Log-Likelihood Ratio[Dunning, 1993].

2.1 Pointwise Mutual Information

Pointwise Mutual Information is defined in Equation 1.

$$PMI = \log \frac{n_{11}}{m_{11}} \tag{1}$$

n_{11} is the count of the number of times a candidate bigram occurs in the Web context data, and m_{11} is the expected count for this bigram, based on the assumption that the two words are independent of each other (meaning that they will only occur together by chance). The expected value m_{11} can be estimated by taking the product of the count of the number of bigrams in the Web contexts that start with the first word (and are followed by any other word), and the count of the bigrams that finish with the second word in the candidate. This product is divided by the total number of bigrams in the sample.

If the observed value is much greater than the expected value, this means that the bigram has occurred more often than would be expected by chance, and that the pair of words is strongly associated and should be considered a feature. PMI suffers from a well known bias towards bigrams that are made up of words that only occur with each other, and in fact gives this highest score to any bigram that only occurs 1 time or to bigrams where the words that make it up only occur in that bigram.

While this is not desirable behavior in general, when dealing with noisy data this can actually be a positive characteristic. In many cases the distribution of identities in ambiguous Web names is very skewed, and the features associated with one name may dominate to the point where the features of the other name can not even be recognized. However, if there is very distinct bigram that occurs with a low frequency name, it can still be identified by PMI since it will rise to the top even with relatively low frequency (assuming that the words in the bigram only occur with each other, or nearly so).

In selecting features with PMI, we require that the bigram occur at least 5 times or 10 times in the Web contexts, and that the PMI score be at least 5 or above, meaning (intuitively) that the bigram has occurred at a rate 5 times expected by chance.

2.2 Log-likelihood Ratio

The Log–Likelihood Ratio (G^2) is defined as shown in Equation 2. G^2 is a score assigned to each candidate bigram that indicates the degree to which the occurrence of that bigram deviates from what would be expected by chance, that is if the words that make up the candidate bigram are independent.

$$G^2 = 2 \sum_{i,j} n_{ij} \log \frac{n_{ij}}{m_{ij}} \tag{2}$$

n_{ij} is the observed count of bigrams, where i and j can have values 1 and 2. n_{11} is the count of the candidate bigram, and n_{12} is the count of bigrams where the first word is the same as the candidate but the second is different. Similarity, n_{21} is the count of bigrams where the second word is the same as the candidate, but the first is different. n_{22} represents the count of bigrams where both the first and second word are different than the candidate bigram.

Note that PMI is in fact one term in this equation (when i and j are both equal to 1). However, rather than focusing on just the count and expected value of the candidate bigram, G^2 considers the counts of the other bigrams in the sample as well. This allows for a formal test of statistical significance, which answers the question of how likely it would be for the candidate bigram to be drawn from the given sample, if the words in the candidate bigram are truly independent.

Thus, the larger the G^2 value the greater the deviation from the expected values, and the more likely that the words in the bigram are not independent. For selecting features we use 3.84 as our threshold, which is the value associated with a 95% probability that the words in the bigram are not independent. This values comes from the Chi–squared distribution, which approximates the distribution of the Log–Likelihood Ratio and can therefore be used as a source of critical values.

2.3 Stop–Lists

A stop–list is made up of words that are not likely to be of use in discriminating among the Web contexts, and which are likely to add further noise to the data if they are used as features. Stop–lists are most often made up of lists of function words (prepositions, conjunctions, etc.) that do not contain a great deal of topical information.

In normal text processing applications, a manually constructed stop list is often utilized, for English this can consist of between 100 and 400 words. We have used one such list in our preliminary experiments which consists of approximinately 200 words and is based on the SMART stop list with some minor modifications.

However, our hypothesis is that Web data may have rather different characteristics than standard written English text, so we have automatically created stop–lists from the Web contexts using Inverse Document Frequency (IDF). This is a widely used measure in Information Retrieval that assigns a score to each word based on the number of documents in a collection, divided by the number of those documents in which the word occurs. Here we consider each Web context to be a document. So, if there are 500 contexts and a word occurs in 100 of them, the IDF score of that word would be 5.

We created a stoplist by grouping together all the contexts for all the names and using that as a single corpus for which we computed IDF scores. We also created stoplists specific to each word by only using the the contexts for that name.

The intuition behind combining the contexts from multiple names is that for very skewed names (where most of the contexts are associated with a single identity) the IDF derived from just the contexts for that name might include words that are very characteristic of the underlying identity. So, by taking the IDF from a more general collection of contexts (and yet still specific to Web data) we hope to identify true stop words and not eliminate genuine content words. We created stoplists by selecting any word with an IDF value equal to or greater than 2, 5, or 10, meaning that the word must occur in at least 50%, 20%, or 10% of the contexts to be considered a stop word.

The IDF-5 stoplist was created from all the contexts for all five names, and by selecting all words with an IDF value of 5 or greater. This resulted in a stoplist made up of the following 22 words (listed in order of IDF score):

> the, of, and, in, a, to, for, on, is, by, with, that, was,
> he, as, this, from, his, at, it, new, an

While we could have simply used a standard stoplist and gotten much the same effect, this list shows that automatic creation is a viable option, which could be employed when manually created lists are not available.

3 Clustering Contexts

We take an approach to clustering Web contexts that is based on the use of shallow lexical features, so as to allow the approach to be language independent and unsupervised. We start with some number of contexts to cluster, each of which contains a particular ambiguous name. We represent these contexts using first or second order context vectors. A first–order vector represents a context based on the features that occur in that context. A second–order vector represents a context by taking an average of word co–occurrence vectors that represent each of the words in a context.

Whether we are using first or second order contexts, our first step is to identify features from the Web contexts to be clustered. These features can be unigrams or bigrams, as was previously described.

In the case of first order features, the process is relatively simple. The features are selected, then each context to be clustered is checked to see if it contains an occurrence of that feature. If it does this is indicated in a feature vector that represents the context, either via a binary value that simply shows the feature occurs, or via a count that indicates the number of times the feature occurs in the context. For first order features we use unigram features selected based on their frequency. We exclude any unigram that appears in our stoplist or is a single character or a digit. The resulting context vectors may optionally be reduced by Singular Value Decomposition (SVD), which is set to reduce the columns in the matrix to 10% of their original number.

For second order features we must use bigrams features. After the bigrams are selected, we construct a word co–occurrence matrix, where the rows correspond to the first word in the bigram, and the columns to the second. The cells in the matrix contain the Pointwise Mutual Information score for that pair of words. In effect, each word is represented by a vector that shows all the words with which it occurs as a bigram. The word matrix may optionally be reduced by Singular Value Decomposition (SVD), again set to reduce the columns down to 10% of their original number.

Then, each word in a Web context is replaced by its vector from the co-occurrence matrix described above. These vectors are then averaged together to create a representation of the context. The averaging operation carried out on these vectors to represent a context is similar to what is done in Latent Semantic Analysis [Landauer *et al.*, 1998], although there the vectors that are averaged together indicate the contexts in which a word has occurred.

Once each context is represented by either a first or second order vector, then clustering can proceed. We employ a hybrid clustering algorithm known as Repeated Bisections, which divides the contexts into partitions, and then clusters each partition using agglomerative techniques.

4 Cluster Stopping

We use an adaption of the Gap Statistic [Tibshirani *et al.*, 2001] and the PK2 measure, both of which were developed by [Pedersen and Kulkarni, 2006] and are a part of the SenseClusters package. These will predict the optimal number of clusters for the Web contexts associated with a given name (a process we refer to as *Cluster Stopping*). As such we do not need to specify ahead of time the number of clusters that we expect to find, this is determined automatically.

In these experiments we have used an internal criterion function (I2) for both techniques, which is a measure of within–cluster (intra) similarity. This is measured by finding the distance of all the contexts in that cluster to the centroid. The goal is to have a cluster that is as tight as possible, that is where the within cluster similarity is maximized.

To perform cluster stopping we carry out clustering where we find solutions with 1 cluster, then 2 clusters, and so forth, up to some number of clusters where there is no further improvement in the quality of the solution. Then, we examine the tend of criterion function scores (I2) for these successive solutions, and seek the point at which adding to the number of clusters does not significantly improve upon the quality of the solution.

The PK2 measure simply compares the value of the criterion function for successive pairs of clusters k and $k-1$. When this ratio approaches 1, then the creation of additional clusters is not improving the quality of the solution, and should be stopped. We select the value of k that is closest to but still greater than one standard deviation in the value of the PK2 scores. Note that this measure is unable to predict the case where there is simply one cluster in that data, which can prove to be a significant limitation since noise or very skewed data may make it impossible to identify any more than one cluster.

The Gap Statistic relies upon comparing the observed value of the criterion function with the value of the criterion function that is estimated from a data set that is essentially random. To do this, a matrix is randomly generated such that the marginal totals are the same as the observed data, but where the internal cell counts are randomly (subject to the constraints imposed by the given marginal totals). Thus, this matrix represents the same population as that of the observed data, except that the data is made up of noise. As such when this data is clustered the criterion function should exhibit a relatively consistent score which will quantify the amount of noise present in the data. Then selecting the number of clusters reducing to finding the point where the difference between the observed and randomly generated data criterion function score is greatest. This is the point at which the observed data is least like noise, and the point where the optimal number of clusters exists.

5 Evaluation

In these experiments we have manually created a gold standard, where ambiguous names have been categorized into their true underlying identity. After some number of clusters have been discovered, they are aligned with the human gold standard such that the agreement between the two is maximized. Each discovered cluster is aligned to a single gold standard cluster, and it is possible that the number of discovered clusters will be more or less than the gold standard amount.

We measure the quality of the resulting clustering using the F–measure, which is the harmonic mean of precision and recall. We define precision to be the number of contexts that are assigned to their correct class, divided by the number of contexts that are assigned a class. Recall is defined as the number of contexts assigned to their correct class, divided by the total number of contexts. Precision and recall differ because the clustering algorithm may decide not to cluster a context, and if the clustering algorithm creates more clusters than there are in the human gold standard, the extra clusters that remain after alignment with the human gold standard are discarded.

Thus, the F-measure provides an indication of how well the clustering is being carried out both in terms of discovering the number of clusters, and then in terms of the quality of the resulting clusters.

6 Experimental Data

We have manually disambiguated Web contexts obtained from the Google Search Engine API for five different ambiguous names:

> Richard Alston, Sarah Connor, George Miller, Ted Pedersen, Michael Collins

Web contexts for each of these names was collected in May 2006 using the Google API, as supported by the CPAN module WebService-GoogleHack-0.15. The top 50 html (or htm) pages found when searching for each of these names were retrieved, and any links from those pages to pages in the same domain were followed and those pages retrieved. However, the links on the second level pages were not traversed.

All the pages retrieved were formatted and cleaned as follows. First, all HTML tags were stripped away using the CPAN module HTML-Format-2.04. This data was divided into contexts using the freely available NameConflate program (version 0.16)[2]. Each context contains a single ambiguous name. Note that contexts may contain variants of the names listed above, such as *M. Collins* or *Ted A. Pedersen*.

Each Web context consists of approximately 100 total words, where the ambiguous name is located in the center of the context. To illustrate the nature of the data, here we show a randomly selected instance from the Richard Alston data, where the target word is in bold face:

> dancers crew musicians permanent members of staff artistic director richard alston administrative director chris may marketing manager sarah lowry production manager helen cain current dancers luke baio annelie binder amie brown jonathan goddard martin lawrance maria nikoloulea sonja peedo francesca romo silvestre sanchez strattner dam van huynh contact details **richard alston** dance company the place duke's road london wc h ab tel fax email radc theplace org uk www theplace org uk exclusive overseas representation david lieberman artists representative s inc po box newport beach california usa info dlartists com www dlartists com photograph hugo glendinning dancers martin lawrance

Table 1 shows the number of contexts associated with each name, and the distribution of identities associated with the contexts:

Note that in clustering if all of the Web contexts for a given name are assigned to the same cluster, the F–Measure will be equal to the percentage of the majority identity in the data. Thus, this serves as a baseline measure to which we can compare.

7 Experimental Results

The goal of our experiments was to compare the effect of various parameter settings that were made in the unsupervised clustering process. We conducted an extensive series of preliminary experiments that are not reported on in detail here, but that led to the following findings.

- We observed that Pointwise Mutual Information nearly always resulted in better performance than the Log–Likelihood Ratio when selecting bigram features. We believe this is because PMI is biased towards pairs of words that only occur with each other, while Log–Likelihood has a bias towards higher frequency words.

[2] http://www.umn.edu/home/tpederse/tools.html

Table 1: Name Data

Name:	Identity	Count	%
Richard Alston:		247	
	Choreographer	176	71.3
	Senator (Aus.)	71	28.7
Sarah Connor:		150	
	German Singer	109	72.7
	Terminator Character	41	27.3
George Miller:		286	
	Congressman (USA)	217	75.9
	Film Director (Aus.)	57	19.9
	Princeton Professor	12	4.2
Michael Collins:		359	
	Irish Leader	269	74.9
	MIT Professor	41	11.4
	Wisc. Professor	32	8.9
	NASA Astronaut	17	4.7
Ted Pedersen:		333	
	Minn. Professor	255	76.6
	Children's Author	43	12.9
	Son of Sea Captain	25	7.5
	TV Writer	10	3.00

Thus, it is easier for Log–Likelihood to be overwhelmed by features from a dominant majority identity, whereas PMI is able to pick out a few relatively low frequency features for the minority class.

- Frequency cutoffs of 2 for unigrams and bigrams significantly degrade performance for cluster stopping and F-Measures. We believe that at this level too much noise is introduced into the feature set.
- Stoplists created using IDF values of 2 and 10 resulted in significantly lower F-measures than those created with 5. The IDF of 2 resulted in an extremely small stop list, with only a few very frequent words like *the* and *an* being present. A value of 10 resulted in the elimination of many content words.
- Stoplists created per name were significantly less effective than those created from all the named contexts. Our data is rather skewed in terms of the underlying identities, and the creation of name specific stoplists led to the removal of a many content words related to that dominant identity.

Thus, we settled on the use of PMI for bigram feature selection, and frequency cutoffs of 5 and 10 for unigrams and bigrams. We elected to use a single stoplist created from all the contexts using an IDF value of 5, which we refer to as IDF-5. In our final round of experiments, we focused on the following questions:

- How effective is the Gap Statistic versus PK2 for predicting the number of clusters given very noisy data?
- Does a stoplist help when dealing with noisy data, or is no stoplist better?
- Does Singular Value Decomposition help to reduce the impact of noise in data, as reflected in both cluster stopping and F-Measure scores?
- Are first or second order representations of context more effective given noisy data?
- Is there any impact on performance in using a frequency cutoff of 5 versus a frequency cutoff of 10?

For each of the five names in our data set, we ran 32 experiments, where each of the five parameters mentioned above was varied between two values. The results of these experiments are shown in Tables 2 through 6. Each of these tables is formatted such that F–Measures are shown in descending order. The column labels are numbers associated with the parameters mentioned above (in order). To summarize, the possible parameter settings represented by each column are: (1) Gap Statistic or PK2 for cluster stopping, (2) IDF-5 or no stoplist during feature selection, (3) SVD prior to clustering (or not), (4) first order unigram or second order bigram features, (5) frequency cutoff of 5 or 10 for feature selection.

A horizontal line is drawn through the table to indicate all those experiments were the F-Measure is equal to or greater than the majority identity in the gold standard data.

For the Richard Alston results shown in Table 2, the highest F–Measure obtained was 99.6%, which means that all of the contexts except one were clustered correctly. This level of performance was attained by one setting, where the Gap statistic was used to determine there were 2 clusters, and with unigram features selected with a frequency cutoff of 10, using an IDF-5 stoplist, and no SVD. We can see that using bigram features results in nearly the same level of performance.

In general most of the different combinations stop–lists, feature type, and context representation resulted in F-Measures significantly higher than the majority identity distribution of 71.26%, which suggests that the Alston data was relatively easy to discriminate. The dominant identity is that of a British choreographer, while the minority identify is an Australian Senator. Both the difference in geography and professions resulted in distinct features for each, making it possible to discriminate these identities with nearly perfect accuracy.

We can make a few other observations about the Alston results. First, SVD seems often to degrade performance, so that does not appear to be helping with this data. Second, the Gap Statistic is more often successful at finding the 2 clusters, while PK2 only seems to do that when contexts are represented using second order bigram features. The top 12 results all have 2 predicted clusters so in general this data seems to have been clearly separable in that number of clusters.

For the Michael Collins results shown in Table 3, the highest F–Measure attained was 93.04%. This was reached using a very simple combination of methods, PK2 for cluster stopping, no stoplist, no SVD, and unigram features with a frequency cutoff of 5. For the Collins data there is a strong majority identity associated with the Irish political leader, and both the geography and profession of this identity was quite distinct from the other identities. As such simply using the unigram features

This provides a stark example of where performing SVD hurts considerably, when only this is changed the cluster stop-

Table 2: Richard Alston Results (2 identities)

(1)	(2)	(3)	(4)	(5)	#	F
gap	idf5	no	uni	10	2	99.60
gap	idf5	no	bi	10	2	97.73
gap	idf5	yes	bi	10	2	97.73
gap	idf5	no	uni	5	2	90.69
gap	none	no	bi	5	2	89.47
pk2	none	no	bi	5	2	89.47
gap	none	no	bi	10	2	88.66
pk2	none	no	bi	10	2	88.66
pk2	none	yes	bi	5	2	88.66
pk2	none	yes	bi	10	2	88.66
gap	none	no	uni	5	2	88.26
gap	none	no	uni	10	2	88.26
pk2	none	no	uni	10	2	88.26
pk2	idf5	no	bi	10	3	83.76
pk2	idf5	yes	bi	10	3	83.76
pk2	idf5	yes	bi	5	4	78.45
pk2	idf5	yes	uni	5	4	77.67
gap	idf5	no	bi	5	3	76.51
pk2	idf5	no	bi	5	3	76.51
pk2	idf5	no	uni	10	4	76.00
pk2	idf5	no	uni	5	4	73.79
pk2	none	yes	uni	5	3	73.39
pk2	none	yes	uni	10	3	73.39
gap	none	yes	bi	5	1	71.26
gap	none	yes	bi	10	1	71.26
gap	none	yes	uni	5	1	71.26
gap	idf5	yes	bi	5	1	71.26
gap	idf5	yes	uni	5	1	71.26
pk2	idf5	yes	uni	10	4	69.74
pk2	none	no	uni	5	3	68.42
gap	none	yes	uni	10	28	27.18
gap	idf5	yes	uni	10	36	21.01

Table 3: Michael Collins Results (3 identities)

(1)	(2)	(3)	(4)	(5)	#	F
pk2	none	no	uni	5	3	93.04
pk2	none	yes	bi	5	2	76.60
pk2	none	yes	bi	10	2	75.21
gap	idf5	yes	uni	5	1	74.93
gap	idf5	yes	uni	10	1	74.93
gap	idf5	yes	bi	10	1	74.93
gap	none	no	bi	5	1	74.93
gap	none	no	bi	10	1	74.93
gap	none	no	uni	5	1	74.93
gap	none	no	uni	10	1	74.93
gap	none	yes	bi	5	1	74.93
gap	none	yes	bi	10	1	74.93
gap	none	yes	uni	5	1	74.93
gap	none	yes	uni	10	1	74.93
gap	idf5	no	bi	10	1	74.93
gap	idf5	yes	bi	5	1	74.93
gap	idf5	no	bi	5	5	72.45
pk2	idf5	no	bi	5	5	72.45
pk2	none	no	bi	5	2	71.31
pk2	none	no	bi	10	2	70.19
gap	idf5	no	uni	5	6	64.59
pk2	idf5	no	uni	5	6	64.59
gap	idf5	no	uni	10	4	60.72
pk2	idf5	no	uni	10	7	59.43
pk2	none	no	uni	10	4	58.50
pk2	idf5	yes	bi	5	5	57.81
pk2	idf5	yes	uni	5	7	57.73
pk2	idf5	yes	bi	10	3	53.48
pk2	idf5	no	bi	10	4	47.46
pk2	none	yes	uni	10	41	23.96
pk2	none	yes	uni	5	41	23.15
pk2	idf5	yes	uni	10	49	19.95

ping method finds 41 clusters and the F-measure is 23.15%, so the results go from best to next to last. It should be noted that in most cases a number of clusters was predicted that differed from what was in our gold standard. PK2 was able to identify 3 clusters in three different experiments, whereas Gap never did. This combined with the fact that the Gap statistic often found 1 cluster suggests that there may not have been many clear patterns in the data beyond those provided by unigrams with a frequency cutoff of 5. There is a curious note on that point, which is that the best results were found when not using a stoplist. When the stoplist IDF-5 was used for the top ranked combination, the F-Measure score fell almost 30 points and the number of clusters was predicted as 6.

The Sara Connor results are shown in Table 4. The dominant identity (German Pop singer) is quite easy to identify in the features, but the minority identity is not (a character in the film *The Terminator*). As such the unigram features performed quite well here, in that they were able to clearly identify the dominant class using a few features like *German* and *music*. As a result 4 of the 5 most accurate methods used unigram features and attained an F-Measure score of 90.0%.

One striking characteristic of these results is that of the top 12 methods, none of them used a stop list. This is perhaps reasonable for bigram features, where having a bigram with one function word and one content word could still be useful. However, the top 4 methods all use unigram features, and the difference in their performance when using a stoplist and not is quite start. For example, the Gap statistic with no stoplist, no SVD, and unigram features that occur more than 5 times reaches 90.00% F-measure, and when the IDF-5 stoplist is used that score drops by 23 points to 67.20%. Clearly some of the function words must be strongly associated with one of the identities, so this might suggest that not using a stoplist would be preferred.

One of the surprises of these experiments was the comparative difficulty of the George Miller and Ted Pedersen, as shown in results Tables 5 and 6. In neither case was any method able to improve upon the majority identity value. This initially surprised us since both of these names have fairly distinct senses.

However, upon examining the features we found that the contexts for the majority identities were extremely rich in text, while the minority sense were somewhat impoverished.

Table 4: Sara Connor Results (2 identities)

(1)	(2)	(3)	(4)	(5)	#	F
gap	none	no	uni	5	2	90.00
gap	none	no	uni	10	2	90.00
pk2	none	no	uni	5	2	90.00
pk2	none	no	uni	10	2	90.00
pk2	none	yes	bi	5	2	90.00
pk2	none	yes	uni	10	2	86.00
pk2	none	yes	uni	5	2	84.00
gap	none	no	bi	5	2	83.33
gap	none	no	bi	10	2	83.33
pk2	none	no	bi	5	2	83.33
pk2	none	no	bi	10	2	83.33
pk2	none	yes	bi	10	2	82.00
pk2	idf5	yes	uni	10	4	76.81
gap	none	yes	uni	5	1	72.67
gap	none	yes	uni	10	1	72.67
gap	idf5	yes	bi	5	1	72.67
gap	idf5	yes	uni	5	1	72.67
gap	idf5	yes	uni	10	1	72.67
gap	idf5	no	bi	10	1	72.67
gap	idf5	yes	bi	10	1	72.67
pk2	idf5	no	uni	10	4	69.83
gap	none	yes	bi	10	5	67.24
pk2	idf5	no	uni	5	4	67.20
gap	none	yes	bi	5	5	63.72
gap	idf5	no	uni	5	8	62.10
pk2	idf5	no	bi	5	4	59.74
gap	idf5	no	uni	10	8	59.53
pk2	idf5	yes	uni	5	4	59.29
gap	idf5	no	bi	5	7	52.68
pk2	idf5	yes	bi	5	4	50.19
pk2	idf5	no	bi	10	4	46.36
pk2	idf5	yes	bi	10	4	46.36

Table 5: George Miller Results (3 identities)

(1)	(2)	(3)	(4)	(5)	#	F
gap	none	no	bi	10	1	75.87
gap	none	no	uni	5	1	75.87
gap	none	no	uni	10	1	75.87
gap	none	yes	bi	10	1	75.87
gap	idf5	no	bi	10	1	75.87
gap	idf5	yes	bi	5	1	75.87
gap	idf5	yes	bi	10	1	75.87
gap	idf5	yes	uni	5	1	75.87
gap	idf5	yes	uni	10	1	75.87
gap	none	yes	bi	5	1	75.87
gap	none	yes	uni	10	1	75.87
gap	none	yes	uni	5	1	75.87
gap	idf5	no	bi	5	3	56.64
pk2	idf5	no	bi	5	3	56.64
pk2	idf5	no	bi	10	3	55.94
gap	idf5	no	uni	5	3	53.50
gap	idf5	no	uni	10	3	53.15
pk2	none	yes	bi	5	2	52.80
pk2	none	yes	bi	10	2	52.45
pk2	idf5	no	uni	5	6	51.44
gap	none	no	bi	5	2	51.40
pk2	none	no	bi	10	2	51.40
pk2	none	no	bi	5	2	51.40
pk2	idf5	yes	bi	10	2	50.70
pk2	idf5	no	uni	10	6	50.21
pk2	idf5	yes	bi	5	3	48.60
pk2	none	no	uni	5	5	45.15
pk2	none	no	uni	10	5	44.49
pk2	idf5	yes	uni	5	7	37.19
pk2	none	yes	uni	10	32	24.56
pk2	none	yes	uni	5	37	20.25
pk2	idf5	yes	uni	10	41	18.96

Thus, no matter what kind of feature identification techniques were employed, it was simply not possible to identify features for any of the minority classes.

The Gap Statistic actually proved to be quite valuable in the case of the Pedersen and Miller data, in that it was able to determine that only 1 cluster should be identified. Other measures that we used in preliminary experiments were unable to recognize the situation where no meaningful distinctions could be made, so they would tend to split the data up into clusters fairly arbitrarily, which led to rather bad F–Measure scores (in the 40% to 50% range). However, because it specifically accounts for noise in the data, the Gap Statistic was able to recognize that it could not distinguish the observed data from noise, and it therefore did not attempt to further divide the data into additional clusters, which at least led to an F–Measure accuracy at the level of the majority identity. This suggests to us that the Gap Statistic is a particularly appropriate choice when working with very noisy data.

8 Conclusions

These experiments illustrate some of the challenges of working with very noisy data, such as we find on the Web. We described an unsupervised method of context discrimination and applied that to the problem of discriminating among named entities. We explored the settings of the parameters for this data, and drew a number of conclusions.

Pointwise Mutual Information is particularly well suited for identifying bigram features in Web data since it focuses on those pairs that only occur with each other, and are somewhat noise resistant as a result.

We found that both the Gap Statistic and the PK2 measure performed well at cluster stopping, although the Gap Statistic offers the significant advantage of identifying when there is just one cluster present, which can be somewhat common given very noisy data.

Somewhat to our surprise, many of our results are often better when not using a stoplist. We are uncertain as to why this would be the case, and it is certainly not universal. However, in general a majority of the experiments that achieved F-Measures above the majority identity did not use a stoplist.

The results for Singular Value Decomposition are somewhat mixed. It clearly helped with the Collins and Pedersen results, but hurt the Alston, Connor, and Miller results.

First order unigrams performed better with the Alston,

Table 6: Ted Pedersen Results (4 identities)

(1)	(2)	(3)	(4)	(5)	#	F
gap	none	yes	bi	5	1	76.58
gap	none	yes	bi	10	1	76.58
gap	none	yes	uni	5	1	76.58
gap	none	yes	uni	10	1	76.58
gap	idf5	yes	bi	5	1	76.58
gap	idf5	yes	bi	10	1	76.58
gap	idf5	yes	uni	10	1	76.58
pk2	idf5	yes	uni	10	5	69.85
pk2	idf5	yes	uni	5	5	60.72
pk2	none	yes	uni	5	5	59.87
pk2	idf5	no	uni	5	6	58.27
gap	idf5	no	uni	10	3	57.96
pk2	none	yes	uni	10	5	55.68
pk2	idf5	no	uni	10	6	54.25
pk2	none	no	uni	5	3	53.15
pk2	none	no	uni	10	3	53.15
pk2	idf5	yes	bi	10	2	53.15
gap	none	no	uni	5	2	52.55
gap	none	no	uni	10	2	52.55
pk2	none	yes	bi	5	2	51.05
pk2	none	yes	bi	10	2	51.05
gap	idf5	no	bi	5	3	50.75
gap	none	no	bi	5	2	49.85
pk2	none	no	bi	5	2	49.85
pk2	none	no	bi	10	2	49.55
gap	none	no	bi	10	2	49.55
pk2	idf5	no	bi	5	6	46.29
gap	idf5	no	bi	10	5	46.10
pk2	idf5	no	bi	10	5	46.10
gap	idf5	no	uni	5	8	45.79
pk2	idf5	yes	bi	5	5	45.37
gap	idf5	yes	uni	5	48	22.40

Connor, and Collins data, where the results were well above the majority identity. Second order bigrams performed better with the Miller and Pedersen and data, which did not exceed the majority identity percentage. This may suggest that second order bigrams should be used when the data is harder to seperate, and that unigrams fare perfectly well in somewhat less challenging circumstances.

9 Acknowledgments

This research has been supported by a National Science Foundation Faculty Early CAREER Development Award (#0092784). All of the experiments in this paper were done with version 0.95 of the SenseClusters system, which is freely available from http://senseclusters.sourceforge.net.

References

[Dunning, 1993] Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1):61–74.

[Harris, 1968] Harris, Z. 1968. *Mathematical Structures of Language*. Wiley, New York.

[Kulkarni and Pedersen, 2005] Kulkarni, A. and Pedersen, T. 2005. Name discrimination and email clustering using unsupervised clustering and labeling of similar contexts. In *Proceedings of the Second Indian International Conference on Artificial Intelligence*, Pune, India. 703–722.

[Landauer *et al.*, 1998] Landauer, T.K.; Foltz, P.W.; and Laham, D. 1998. An introduction to Latent Semantic Analysis. *Discourse Processes* 25:259–284.

[Levin *et al.*, 2006] Levin, E.; Sharifi, M.; and Ball, J. 2006. Evaluation of utility of LSA for word sense discrimination. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, New York City. 77–80.

[Mann and Yarowsky, 2003] Mann, G. and Yarowsky, D. 2003. Unsupervised personal name disambiguation. In Daelemans, W. and Osborne, M., editors 2003, *Proceedings of CoNLL-2003*. Edmonton, Canada. 33–40.

[Miller and Charles, 1991] Miller, G.A. and Charles, W.G. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes* 6(1):1–28.

[Pedersen and Bruce, 1997] Pedersen, T. and Bruce, R. 1997. Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, RI. 197–207.

[Pedersen and Kulkarni, 2006] Pedersen, T. and Kulkarni, A. 2006. Selecting the right number of senses based on clustering criterion functions. In *Proceedings of the Posters and Demo Program of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. 111–114.

[Pedersen *et al.*, 2005] Pedersen, T.; Purandare, A.; and Kulkarni, A. 2005. Name discrimination by clustering similar contexts. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City. 220–231.

[Pedersen *et al.*, 2006] Pedersen, T.; Kulkarni, A.; Angheluta, R.; Kozareva, Z.; and Solorio, T. 2006. An unsupervised language independent method of name discrimination using second order co-occurrence features. In *Proceedings of the Seventh International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City. 208–222.

[Purandare and Pedersen, 2004] Purandare, A. and Pedersen, T. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, Boston, MA. 41–48.

[Schütze, 1998] Schütze, H. 1998. Automatic word sense discrimination. *Computational Linguistics* 24(1):97–123.

[Tibshirani *et al.*, 2001] Tibshirani, R.; Walther, G.; and Hastie, T. 2001. Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistics Society (Series B)* 411–423.