Name Discrimination and Email Clustering using Unsupervised Clustering and Labeling of Similar Contexts

Anagha Kulkarni and Ted Pedersen

Department of Computer Science
University of Minnesota
Duluth, MN 55812, USA
{kulka020,tpederse}@d.umn.edu
http://senseclusters.sourceforge.net

Abstract. In this paper, we apply an unsupervised word sense discrimination technique based on clustering similar contexts (Purandare and Pedersen, 2004) to the problems of name discrimination and email clustering. Names of people, places, and organizations are not always unique. This can create a problem when we refer to or seek out information about such entities. When this occurs in written text, we show that we can cluster ambiguous names into unique groups by identifying which contexts are similar to each other. It has been previously shown by (Pedersen, Purandare, and Kulkarni, 2005) that this approach can be successfully used for discrimination of names with two-way ambiguity. Here we show that it can be extended to multiway distinctions as well. We adapt the cluster labeling technique introduced by (Kulkarni, 2005) for the multiway distinctions of name discrimination. On the similar lines of contextual similarity, we also observe that email messages can be treated as contexts, and that in clustering them together we are able to group them based on their underlying content rather than the occurrence of specific strings.

1 Introduction

Humans and systems alike have long encountered the problem of name ambiguity caused by multiple people or places or organizations sharing the same name. With the perpetual growth of World Wide Web this problem is becoming more and more pervasive. For example a Google search for the name *George Miller* returns web-pages related to the famous Psychologist from Princeton University and also returns web-pages about an Australian movie director. Another example can be the city name *Duluth* which when looked up on Google comes up with various links to *Duluth, Minnesota* as well as *Duluth, Georgia*. Such unresolved ambiguity can lead to degradation of systems like information retrieval, search engines to name a few.

Bhanu Prasad (Editor): IICAI-05, pp. 703-722, 2005.

We extend and adapt the methods proposed by Purandare and Pedersen [8] for unsupervised word sense discrimination to this problem of name discrimination. They base their approach on the methods proposed by Schüzte[14] and Pedersen and Bruce [15]. The philosophy underlying all these methods is a hypothesis proposed by Miller and Charles [17] where they state that *two words are semantically similar to the extent that their contextual representations are similar.* This philosophy holds true for name discrimination problem too because given various occurrences of an ambiguous name in a corpus the occurrences that share the similar neighborhood (surrounding context) can be expected to refer to the same underlying entity. For instance all the occurrences of *George Miller* which occur along with *Princeton University* or *WordNet* can be expected to refer to the Psychologist whereas the ones co-occurring with *Australia* or *Mad Max* are highly likely to refer to the Australian Movie Director.

Typically determining the actual meaning for an ambiguous word in a given context requires manually annotated training data or resources like machine readable dictionaries, thesauri or ontologies. Such manually crafted data can bring good results but creating and maintaining such data can quickly become tedious, costly and thus a bottleneck for the system. Secondly the annotations are domain specific, for example the word *Virus* would mean or would be annotated as *Malicious Program* in computer related domain whereas it would be labeled as *Infectious Particle* in biological domain. Lastly the methods relying on such manually created resources inherently become language dependent. Thus in the proposed approach we avoid this dependence on external knowledge sources by using unsupervised techniques for extracting and learning. Another important reason for adopting unsupervised method especially for the name discrimination problem is creating and maintaining annotated data or semantic networks for proper names is a highly infeasible task.

We also present the preliminary exploration of email clustering domain by adapting the unsupervised word sense discrimination methods. The main objective is to be able to cluster a given set of emails based upon the overall topic of the email. This can be very useful for managing emails when we do not want to group or separate emails based only upon presence or absence of a particular string but want to cluster them based on the similarity of the underlying topic of the email.

The main difference between name discrimination and email clustering is that for email clustering no specific ambiguous word is targeted but the contextual similarity between the entire email contents is estimated. The ambiguous word is referred to as the target-word or also as the head-word and thus name discrimination experiments are also referred as *headed clustering* while email clustering is referred to as *headless clustering.* Thus for *headless clustering* like email clustering we can expect that given a set of emails discussing various topics like *computer graphics*, *politics* etc. the method would separate the emails based on the overall topic of the emails instead of any particular word(s).

The contexts to be clustered can be as large as a complete document as in case of the email clustering problem or can be restricted to few words around the target-word as in case of name discrimination problem.

We have adapted and extended an Open Source suite of Perl programs developed by Pedersen and Purandare for word sense discrimination namely the SenseClusters' package.

2 Related Work

Bagga and Baldwin [13] have proposed a method using the Vector Space Model to disambiguate references to a person, place or event across documents. The proposed approach uses their previously developed system CAMP (from the University of Pennsylvania) to find *within document* coreference. For example, it might determine that *he* and *the President* refers to *Bill Clinton*. CAMP creates co-reference chains for each entity in a single document, which are then extracted and represented in the vector space model. This model is used to find the similarity among referents, and thereby identify the same referent that occurs in multiple documents.

Mann and Yarowsky[10] take an approach to name discrimination that incorporates information from the World Wide Web. They propose to use various contextual characteristics that are typically found near and within an ambiguous proper-noun for the purpose of disambiguation. They utilize categorical features (e.g.,age, date of birth), familial relationships (e.g., wife, son, daughter) and associations that the entity frequently shows (e.g. country, company, organization). Such biographical information about the entities to be disambiguated is mined from the Web using a bootstrapping method. The Web pages containing the ambiguous name are assigned a vector depending upon the extracted features and then these vectors are grouped using agglomerative clustering.

Pantel and Ravichandran[7] have proposed an algorithm for labeling semantic classes, which can be viewed as a form of cluster. For example, a semantic class may be formed by the words: *grapes, mango, pineapple, orange* and *peach*. Ideally one would like this cluster to be labeled as the semantic class of *fruit*. Each word of the semantic class is represented by a feature vector. Each feature consists of syntactic patterns (like verb-object) in which the word occurs. The similarity between a few features from each cluster is found using point-wise mutual information (PMI) and their average is used to group and rank the clusters to form a grammatical template or signature for the class. Then syntactic relationships such as *Noun like Noun* or *Noun such as Noun* are searched for in the templates to give the cluster an appropriate name label. The output is in the form of a ranked list of concept names for each semantic class.

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 and thereby introduce 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.

There has been some research on automatically organizing email based on topic or category. However, many of these techniques use supervised learning, which requires an existing pool of labeled examples to serve as training data, and the learned model is limited to assigning incoming email to an existing category.

For example, Bekkerman et al.,[4] propose a supervised approach for categorizing emails into predefined folders. They apply Maximum Entropy, naive Bayes, Support Vector Machine (SVM) and Wide-Margin Winnow classifiers to the Enron and SRI[1] email corpora. They have extended the conventional Winnow classifier for multi-class problems. A Winnow classifier is similar to a simple perceptron which learns weights to be applied to the data instances to determine their output class. The learning involves updating the weights using training data instances. Bekkerman et al. propose using a wider margin or separation between target classes by adjusting the weights not only when a training instance is classified incorrectly but also when it is classified correctly with very small margin over other classes. They use the traditional bag-of-words approach to represent features of the emails. They also introduce an evaluation methodology which they refer to as *step-incremental time-based split* that provides better evaluation of the proposed techniques for the given task of email foldering. There reason behind not using the traditional random test/train split is to account for the time-dependent nature of the email categorization task. Therefore they sort the emails based on their time-stamps and then train on the first N emails and test on the next N emails, subsequently train on the first 2N emails and test on the next N emails and so on. In their results, although the SVMs achieved the best accuracy most of the times, their Wide-Margin Winnow classifier compared fairly well given its simplicity and speed.

Kushmerick and Lau[1] automate email management based on the structured activities that occur via email, eg: Ordering a book from Amazon.com will lead to a thread of emails regarding - order confirmation, order status, order delivery. The authors use finite-state automata to formalize this problem where the states of the automata are the status of the process (eg: order conformation) and the transitions are the email messages. They divide the problem into four tasks of Activity Identification, Transition Identification, Automaton Induction and Message Classification. The first task handles the identification of various emails that are related to an activity. Next task identifies all the emails that cause transition from one state to another. In the third task the process model is

[1] http://www.ai.sri.com/project/CALO

generated using the data identified in the previous two tasks. Finally the last task takes care of assigning the appropriate transition to a new message entering the system. Kushmerick and Lau report the results in terms of the accuracy (86% to 97%) with which the methods were able to predict - the next state, the end of activity and the overlap between the predicted and correct transition message.

The 20 Newsgroup email corpora that we experiment with was developed by Ken Lang. This data was then used for the NewsWeeder system[16] which learns user preferences while reading the NetNews. NewsWeeder prompts the user to rate the document that he reads over the range of rating 1 to 5 and uses these ratings to build the user specific learning model. It uses the bag-of-words approach for feature representation and uses tf-idf (term frequency - inverse document frequency) or Minimum Description Length (MDL) to decide the predicted rating for any new document.

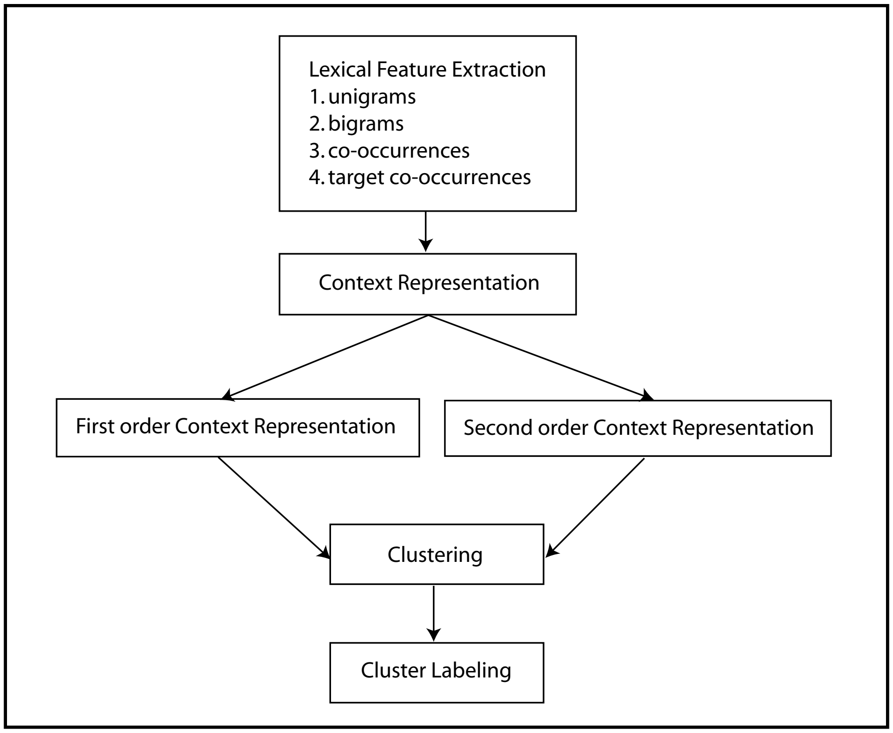

Fig. 1. Flowchart

3 Methodology

In this section we briefly describe the overall methodology and then elaborate upon individual phases in subsequent sections.

As shown in Figure 1 we start with the lexical feature extraction phase. Lexical features are short textual entities like single words or word-pairs. These lexical features are usually looked at as the representative of the underlying corpus or also as the entities which can translate a corpus to an abstract level. With SenseClusters these features can be of type unigrams which are significant individual words or bigrams which are ordered pairs of words or co-occurrences which are unordered pairs of words or target-co-occurrences which are unordered pairs of words in which one of the words is the target-word. These features can be extracted from a separate feature selection data (which is not clustered) or from the data that is to be clustered. These features can be selected by simple ranking of their frequency of occurrence or by using statistical tests of association.

The next step is to represent each contexts in terms of the extracted features in the earlier step. By doing this each context is translated into a vector representation. As noted in the Figure 1 the representation can be either first order context representation or second order context representation. The exact details about this representations are described later in the *Context Representation* section. A typical context for name discrimination experiments consists of a marked target-word and few words/sentences pivoted around this target-word. While for email clustering the complete email is represented as a context without any target-word.

Once each context either *headed* or *headless* is represented as a vector all these vectors are clustered into separate clusters such that intra-cluster similarity is maximal and inter-cluster similarity is minimal. SenseClusters utilizes CLUTO[2] which is a suite of various clustering algorithms.

Further each cluster thus formed is assigned a list of significant words referred to as the label. These words are picked from the contents of the cluster and thus these cluster labels summarize the contents of the cluster and in a way provide an easy automated way to identify the entity that the cluster represents.

Next we describe each step of our methodology in details.

3.1 Feature Identification

Feature Identification/extraction is the first step of the methodology. The identified lexical features determine how accurately the contexts are represented and thus the extent to which the similarity between the contexts can be captured.

SenseClusters supports four different types of lexical features and each one of them captures slightly different information from the others. The supported features are unigrams, bigrams, co-occurrences and target co-occurrences. Unigrams are the simplest of the features. These are individual words that occur

[2] http://www-users.cs.umn.edu/∼karypis/cluto

in the corpus more often than some cut-off frequency. Thus unigrams represent the class of single words from the feature selection corpus that occur frequently enough to clear the basic test of noise elimination. Bigrams are ordered pairs of words, thus "World History" and "History World" would be two separate bigrams. The word-pairs can optionally have intervening words between them in the actual corpus which are ignored while forming the bigram. The number of the intervening words that are allowed between the word-pair is decided by the specified window size. For example in this phrase "World and its ancient History" the intervening words "and its ancient" are ignored while selecting the bigram and the window size must have been at least 5. Co-occurrences are unordered word-pairs, thus "World History" and "History World" will be two occurrences of just one co-occurrence feature. Co-occurrences can also have intervening words in the actual corpus. By retaining the positional information for bigrams we expect to capture phrases or collocations whereas with co-occurrences we identify the word-pairs that tend to occur together. Finally the target co-occurrences are the unordered pairs of words where one of the words is the target-word and thus can be used only with name discrimination or in general applications where some specific word is being disambiguated (*headed* data) but not with email clustering (*headless data*). This feature is based on the reasoning that the words near to the ambiguous word are more likely to be related to it than the words that are farther away. Another restriction that this feature type imposes is on the corpus of feature extraction - a plain text corpus cannot be used to extract target co-occurrences features but a corpus where the occurrences of the target-word are appropriately marked, that is, *headed* data can only be used.

These features can be specified to be selected from the test data or from a separate set of raw data which is not clustered but is only used for feature identification and thus referred to as the feature selection corpus. In either case, there is no manually annotated information about the underlying entity of ambiguous names or the correct clustering of email messages available.

Though using all the identified features is certainly possible for representing the contexts, we always select and use only the significant features. The simplest method of choosing significant features is to rank them according to the frequency of their occurrence in the corpus and then select the top n ranked features. This is the only technique that can be used with unigram feature types. For bigrams and co-occurrences more complex techniques - statistical tests of association like log-likelihood ratio, mutual information, point-wise mutual information etc. can be used. These techniques help identify the significant word-pairs which are the word-pairs that occur more often than expected by chance.

3.2 Context Representation

Once we have the set of identified lexical features we proceed to represent each context in terms of these feature either directly or indirectly. With SenseClusters contexts can be represented using first order or second order context representation. The first order representation is based upon the technique that Pedersen and Bruce [15] adopt. In this representation we create a matrix with each context representing a row and each selected feature representing a column. The value at cell ([x,y]) in this matrix represents the frequency of occurrence of the yth column (feature) in the xth row (context). This matrix is a large sparse matrix with mostly binary values. The sparsity of the matrix is because the number of features present in any given context are very few compared to the total number of identified features. And the binary nature of the matrix value is because any open class word or pair of words will rarely occur more than once in a sentence. But this theory of binary values is true only if each context is a sentence or a small piece of text because words can occur more than once in a paragraph/document.

We use Singular Value Decomposition (SVD) to reduce the dimensionality of this matrix and thus reduce the sparsity. SVD also has a effect of smoothing the values. The post-SVD column dimensions of the matrix are minimum of 10% of the actual column dimensions or 300. Thus if the original column dimensions were more than 3000 then the matrix is reduced to 300 columns but if the original column dimensions were less than 3000 then the reduced column dimensions are 10% of the actual size. Each row of this reduced matrix can be now looked at as a vector representing the context at the row. Thus the matrix translates into context vectors at each row of the matrix which are later clustered.

Our second order context representation is adapted from Schüzte [14]. This representation cannot be used with unigram feature type. We start by representing the identified bigram or co-occurrence features in a word by word matrix format where first word of the feature is represented across the row, the second word across the column and the cell values are either their co-occurrence frequencies or the statistical scores of test of associativity. Note that this matrix does not incorporate any information from test data in it. SVD is employed to this matrix for dimensionality reduction and smoothing of values. Each row of this reduced matrix can be interpreted as the word vector for the word it represents. This word vector carries the information about the co-occurrence pattern of the word, that is, this word vector gives the information about which other words does this word occur with. These word-vectors are used to create the context vector representation. A context vector is created by averaging the word-vectors for the words that occur in the context. For name discrimination problems we can restrict the words whose word-vectors contribute towards the formation of the context vector to the ones nearer to the target-word. We refer to this as restricting the scope and show the improvement gained by its use in [2] and [3]. This is based on the theory that a word positionally nearer to the target-word is more likely to be related to the target-word than a word farther from it. Thus if we restrict the words that are averaged we reduce the noise (the information not related to the target-word), picked by the words at greater distance from the target-word. Once the contexts are represented in vector format clustering follows.

3.3 Clustering

Clustering algorithms are broadly classified as Hierarchical, Partitional and Hybrid.

Hierarchical clustering is further classified into the bottom-up agglomerative and the top-down divisive techniques. Agglomerative clustering starts with singleton clusters where each context is placed in a separate cluster. At every stage two clusters which are most similar to each other are merged together. This can be repeated until all the contexts are combined into a single cluster. The divisive clustering works in exactly the opposite direction - from one cluster to multiple cluster, splitting clusters at every stage based on their dissimilarity. Thus clustering of data using Hierarchical techniques proceeds in stages.

Unlike Hierarchical clustering, Partitional techniques cluster the data in just one stage. K-means is a one of the widely used Partitional clustering technique. The clustering mechanism of K-means is based on the centroids/means of the cluster. K-means starts by choosing k random data points as the centroids and then assigns the remaining data points to the nearest centroid. The centroids of clusters thus formed are re-calculated. This process of assigning data points to the nearest centroid is repeated until the centroids do not change and thus k clusters are obtained from the dataset.

As one would expect the Hierarchical techniques have quadratic time complexity whereas the Partitional techniques have linear time complexity. Although Partitional techniques are computationally more efficient than the Hierarchical clustering it has been traditionally believed that Hierarchical techniques produce better solutions than Partitional. Zhao and Karypis [11] have found that contrary to the common belief about inefficiency of the Partitional algorithms, they out-perform other clustering algorithms for large corpora. They attribute this poor performance of Hierarchical clustering to the merging errors that occur at the early stages of the clustering and get multiplied along the way. The merging errors usually get introduced in the early stages because of the nature of hierarchical clustering which lacks the global view of the data.

For our domain we have found that the partitional methods that produce Hierarchical Clustering solutions using repeated bisections usually give the best results. The intention is to take advantage of the global view of the partitional algorithms but also to reduce the instability induced because of the initial k random centroids. Specifically, instead of partitioning the data directly into k partitions, the data is partitioned into 2 (bisected) clusters at each stage. Thus the initial one cluster containing all the contexts is partitioned into two clusters. Then the larger of the two is further partitioned into two clusters and this can continue until each of the contexts are partitioned into its own separate cluster or k-1 times if k clusters are expected from the data.

CLUTO, a suite of clustering algorithms that provides various algorithms under all the above mentioned categories is integrated with SenseClusters seamlessly for clustering of contexts.

Currently SenseClusters requires to be provided with the number of clusters we want to group the contexts into. We are diligently working towards automating this process by looking at the trend of the criterion function and various cluster stopping rules.

3.4 Cluster Labeling

We try to address the commonly faced problem of identifying the underlying entity that a cluster represents without having to manually examine the cluster contents.

Once the context vectors are separated into various clusters we assign each of these clusters a list of bigrams which we refer to as a label for the cluster. The purpose of assigning these labels is to summarize the contents of the clusters in terms of the most significant words that occur in the cluster and thus help one identify the actual underlying entity. The labels are classified into two types specifically *Descriptive* and *Discriminating*. The *Descriptive* labels are the top N bigrams and *Discriminating* labels are the top N bigrams which are unique to the cluster. The *Descriptive* labels capture the main concept or entity of the cluster whereas the *Discriminating* clusters try to convey the contents that separates one cluster from another cluster.

Each cluster is composed of various contexts grouped under that cluster by the clustering algorithm, these contexts together form a corpus for our label generation process. Similar to the feature selection process we identifying the bigrams by ranking them either on their frequency of occurrence or their statistical scores. The top N bigrams are picked as the *Descriptive* bigrams while top N unique bigrams are picked as *Discriminating* labels.

3.5 Test Data

For the name discrimination problem there are very few compiled sets of test data. There are few well-known datasets like the John Smith corpus compiled by Bagga and Baldwin and the name data by Mann and Yarowsky which can be used. Thus to overcome this deficit of test data, we have developed a utility called NameConflate[3] which creates artificial ambiguity.

NameConflate can create test data from the English GigaWords and plain text Corpus. It looks for occurrences of the specified words and conflates them, for example all the occurrences of *Tony Blair* and *Vladimir Putin* will be replaced with *TonyBlairVladimirPutin* and thus will create artificial ambiguity. The original word is retained in a separate tag for evaluation purposes. Each such conflated occurrence is divided into contexts with n words on either sides of the conflated word. Finally all these contexts are embedded in the SENSEVAL2[4] format compliant tags.

[3] http://www.d.umn.edu/∼kulka020/kanaghaName.html

[4] http://www.senseval.org/

Artificially created ambiguity for word sense discrimination of general English words is usually questionable because one cannot be certain about the various actual underlying senses of the conflated word. But use of conflation for testing proper noun - name discrimination, can be justified based on the fact that generally for any given unambiguous name one can be certain about the underlying entities.

Relatively more data is available for email clustering experiments. Resources like 20 NewsGroup Data[5] and Enron Data[6] are widely used for research purposes.

The Senseval2 formatted test data can be easily created from an email corpus by representing contents of each email as an individual context and by adding all the other necessary format tags.

4 Experimental Data

We make use of data for which the correct entity in case of the name discrimination problem and the group in case of email clustering problem is already known so that we can automatically evaluate our methods and do not have to rely on manual evaluation. This information is strictly ignored till the evaluation stage.

4.1 Name Discrimination

We have used Agence France Press English Service (AFE) portion of the English GigaWords Corpus, as distributed by the Linguistic Data Consortium for the name discrimination experiments. In particular, we use the corpus with 234,162,179 words which had appeared as AFE newswire data from January 2002 to June 2002.

We have categorized the experiments into two types namely 2-way and 3-way experiments. Using the NameConflate utility we conflate two names to ambiguate all the occurrences of these two names and thus create 2-way ambiguity whereas for 3-way we ambiguate three names. For few 3-way experiments we introduce a new dimension to the existing 2-way experiment like for example we add *India* to the 2-way experiment of *Mexico* and *Uganda*. And for others we have grouped together two different experiments of 2-way category, for example the experiment where we group *Microsoft* and *Compaq* example of 2-way category with *Serena Williams* from another category of 2-way experiment.

4.2 Email Clustering

We use the 20 NewsGroup Corpus of USENET articles for the email clustering experiments. The 20 NewsGroup Corpus consists of approximately 20,000

[5] http://people.csail.mit.edu/jrennie/20Newsgroups

[6] http://www.cs.cmu.edu/ enron

articles already classified into 20 categories such as *computer graphics*, *recreational motorcycles*, *science electronics* and so on. We ignore this categorization information and use it only in the evaluation stage.

Similar to name discrimination we have categorized these experiments into 2-way and 3-way categories.

5 Experimental Setup

The feature type that we use for all the experiments is the bigram feature which captures more information than the unigrams and also is less restrictive than the target co-occurrences which mandate the word-pairs to contain the target-word. We use the log-likelihood ratio with cutoff of 3.841 for ranking the bigrams according to the associativity between the two words of the bigram. The cutoff of 3.841 signifies 95% certainty that the two words in the bigrams are not occurring together strictly by chance. The bigrams which occur less than five times in the corpus are ignored. We also employ Or stop-listing which means that if either of the word in a bigram is a stop word then the bigram is filtered out. After enforcing all these elimination rules what we are left with is a rich set of features.

We experiment with both the context representations - first order and second order context representation. Though provided as an optional feature by the SenseClusters package we perform Singular Value Decomposition for all the experiments.

We have restricted the scope for the second order name discrimination experiments to five words on either sides of the target-word. By this we restrict the number of word-vectors that will be averaged to generate the context vector to ten word-vectors.

Each experiment is performed once with number of clusters set to actual number of entities or news groups present in the data and once with number of clusters set to six. The theory behind setting the cluster number to artificially high value is to test the method in the situation where the user does not know how many entities are present in the test data. The expectation is such cases where the test data has n actual entities and the number of clusters specified is c (where c > n) is that finally even if c number of clusters are generated only n of them would be populated and the c-n clusters would be empty and thus can be ignored.

The experimental settings used for the cluster labeling is similar to feature identification settings. We use log-likelihood ratio with the cutoff of 3.841 to rank the identified bigrams. We also use Or stop-listing and a frequency cutoff of five occurrences. Then out of the remaining bigrams we select the top ten bigrams as the *Descriptive* labels and the top ten unique bigrams as the *Discriminating* labels.

6 Evaluation

We evaluate the methods in terms of the agreement between the clustering performed by the system and that given by the gold standard.

We create a ”cluster” by ”senses” contingency matrix. The ”senses” are the actual names for name discrimination and the email groups for the email clustering experiments. Table 1 and Table 2 show such contingency matrix for the *Tony Blair* and *Vladimir Putin* 2-way experiment and *Tony Blair*, *Vladimir Putin* and *Saddam Hussein* 3-way experiment. Distribution of contexts in each cluster is noted across the rows. Column total gives the total number of contexts present in the test data for the sense represented by the column. For example we can verify with the Table 4 that number of contexts for *Tony Blair* are indeed 1436 as indicated in Table 1 by the column marginal of the column representing the sense *Tony Blair*. In ideal case we would expect every cluster to be composed of only one type of contexts which is represented via the hypothetical contingency matrix as shown in Table 3.

Once such contingency matrix is created we re-order the columns to maximize the total across the major diagonal which leads to maximization of the agreement. The reason for doing this re-ordering is that each cluster can represent only one sense even if it contains contexts of other senses. Once a sense is assigned to a cluster the contexts of all the other senses present in the cluster are incorrectly grouped. Thus by re-ordering we try to maximize the contexts that get assigned to their correct sense (name/group). This assignment of senses to clusters in a way that would maximize the correct assignment can be directly mapped to the classical assignment problem. This problem can be certainly solved by using Brute-force method that is by trying each possible combination of assignment. But this method quickly becomes computationally inefficient as the number of clusters and senses increase. Thus we adopt a highly efficient algorithm called Kuhn-Munkres' Algorithm [21] [20] or also referred to as the Hungarian Algorithm to solve the assignment problem.[7]

Finally note that before this evaluation process starts the system has already discriminated the contexts and this stage does not change the performance of the system but automates the evaluation or the agreement measuring process and thus avoids having to manually scan through the clustered contexts.

Table 1. Re-ordered Name to Cluster Assignment Contingency Matrix (2-way)

	Tony Blair	Vladimir Putin	
C0	1341	27	1368
C1	95	1761	1856
	1436	1788	3224

[7] http://search.cpan.org/dist/Algorithm-Munkres/

Table 2. Re-ordered Name to Cluster Assignment Contingency Matrix (3-way)

	Tony Blair	Vladimir Putin	Saddam Hussein	
C0	1341	27	5	1373
C1	5	1463	614	2082
C2	90	298	429	817
	1436	1788	1048	4272

Table 3. Ideal Case of Name to Cluster Assignment Contingency Matrix (3-way)

	Tony Blair	Vladimir Putin	Saddam Hussein	
C0	1436	0	0	1436
C1	0	1788	0	1788
C2	0	0	1048	1048
	1436	1788	1048	4272

7 Experimental Results

The results, which are the agreement between the clustering proposed by the methods and that proposed by the answer key of the dataset are specified in terms of F measure which is a harmonic mean of Precision (P) and Recall (R) values for the experiment. The Precision value is the percentage of contexts clustered correctly out of those that were attempted while Recall is percentage of contexts clustered correctly out of the total number of contexts.

Specifically,

$$F = \frac{2 * P * R}{(P + R)} \tag{1}$$

The baseline used in these experiments is the Majority Classifier which is calculated by clustering all the contexts in a single cluster and then calculating the Precision, Recall and F measure for this single cluster. In other words this baseline specifies the result that one can expect without clustering the contexts at all which is the lowest threshold for effectiveness of any method.

Table 4 summarize the experimental results for the name discrimination task. The first column of the table specifies the original names that were conflated. The next column (M) gives the count of the contexts per word in the test data, that is, the test data for *Tony Blair* and *Vladimir Putin* contains 1436 contexts about *Tony Blair* and 1788 contexts about *Vladimir Putin*. Thus this column shows the distribution of the names in the test data. The third column specifies two values - the first one (MAJ.)is the majority classifier for the experiment which is the baseline and the next (N) is the sum of earlier column which is the total number of contexts in the test data. The fourth column shows the number of clusters we sought to discover (K) for the experiment. The next to last column

(Order 1) and the last column (Order 2) indicates the results for experiment with first order and second order context representation respectively.

Table 5 summarizes the results for the email clustering experiments and the column heading interpretation is same as that for Table 4.

Labels assigned to the clusters of *Tony Blair* and *Vladimir Putin* in case of 2-way experiment and the labels assigned to *Tony Blair*, *Vladimir Putin* and *Saddam Hussein* for the 3-way experiment are shown in Table 6. The bigrams in bold face indicate those word-pairs that were selected as *Descriptive* as well as *Discriminating* labels for that cluster and bigrams in normal font face are the *Descriptive* labels.

As we can see majority of the labels are *Descriptive* as well as *Discriminating*. This is expected and in fact indirectly indicates the effectiveness of the clustering algorithm because if the clustering algorithm is able to separate the contexts correctly then the contents of the clusters which are unique to it should be the one which are also significant and commonly occurring in it. In short overlapped *Descriptive* and *Discriminating* labels indicate that the identified clusters are clearly distinct.

Table 4. Experimental Results for name discrimination in terms of F measure

Target Word	M	MAJ. (N)	K	Order 1	Order 2
Tony Blair	1436	55.45	2	**94.88**	**96.22**
Vladimir Putin	1788	(3224)	6	61.44	76.17
Mexico	1256	50.00	2	**60.11**	**59.16**
Uganda	1256	(2512)	6	51.37	51.89
Microsoft	380	50.00	2	**68.42**	**70.26**
Compaq	380	(760)	6	54.37	57.57
Serena Williams	308	51.41	2	53.09	**68.95**
Tiger Woods	291	(599)	6	51.23	63.39
Sonia Gandhi	112	50.45	2	**89.15**	**91.03**
Leonid Kuchma	110	(222)	6	60.12	54.37
Tony Blair	1436	41.85	3	72.66	**75.68**
Vladimir Putin	1788	(4272)	6	62.23	67.31
Saddam Hussein	1048				
Mexico	1256	33.34	3	**44.75**	**46.44**
Uganda	1256	(3768)	6	37.66	**45.25**
India	1256				
Microsoft	380	33.34	3	51.95	52.60
Compaq	380	(1140)	6	**56.62**	52.08
Serena Williams	380				

Table 5. Experimental Results for Email Clustering in terms of F measure

	M	MAJ.	K	Order 1	Order 2
NewsGroup		(N)			
COMP.GRAPHICS	(584)	50.04	2	50.90	**62.19**
MISC.FORSALE	(585)	(1169)	6	34.49	41.37
COMP.GRAPHICS	(584)	50.87	2	**68.82**	57.06
TALK.POLITICS.MIDEAST	(564)	(1148)	6	41.23	47.81
REC.MOTORCYCLES	(598)	50.12	2	**61.53**	**62.70**
SCI.CRYPT	(595)	(1193)	6	44.27	42.54
REC.SPORT.HOCKEY	(600)	50.04	2	55.55	**63.14**
SOC.RELIGION.CHRISTIAN	(599)	(1199)	6	40.71	41.29
SCI.ELECTRONICS	(591)	50.33	2	50.25	**54.87**
SOC.RELIGION.CHRISTIAN	(599)	(1190)	6	38.18	46.85
COMP.GRAPHICS	(584)	33.70	2	36.69	**39.84**
REC.AUTOS	(594)	(1777)	6	37.07	34.76
SOC.RELIGION.CHRISTIAN	(599)				
MISC.FORSALE	(585)	33.63	2	**40.15**	**40.20**
SCI.MED	(594)	(1775)	6	33.43	**40.08**
REC.SPORT.BASEBALL	(597)				
TALK.POLITICS.GUNS	(546)	35.80	3	**40.06**	38.35
TALK.POLITICS.MIDEAST	(564)	(1575)	6	**35.26**	30.58
TALK.POLITICS.MISC	(465)				

Table 6. Cluster Labels for name discrimination Experiments

True Name	Created Labels
Cluster 0: Tony Blair	**British Prime, Minister, Downing Street, Middle East, President George, words moved**, George W, Prime Minister, United States, W Bush
Cluster 1: Vladimir Putin	**Cold War, President, Russian President, Saint-Petersburg, TV 6, news agency,** George W, Prime Minister, United States, W Bush
Cluster 0: Tony Blair	**British Prime, Minister, Downing Street, Middle East, words moved,** President George, George W, Prime Minister, United States, W Bush
Cluster 1: Vladimir Putin	**Iraqi President, President, Russian President, TV 6, news agency,** George W, Prime Minister, United States, W Bush
Cluster 2: Saddam Hussein	**Iraqi leader, Russian counterpart, US President, counterpart, leader** George W, President George, Prime-Minister, United States, W Bush

8 Discussion

As we can see from the Table 4 almost all the results are significantly above the baseline, especially for the experiments where the number of clusters specified is equal to the actual senses in the test data and is not an artificially high value.

Another trend that can be clearly seen from the Table 4 results is that not just people names but particularly names of personalities related to politics are disambiguated more effectively. By that we are referring to the 2-way experiments about *Tony Blair*, *Vladimir Putin* and *Sonia Gandhi*, *Leonid Kuchma* which show remarkable increase in the results over the baseline. But at the same time the *Serena Williams* and *Tiger Woods* experiment though about names of personalities which are often discussed in news paper does not perform as good as we would expect. This trend is certainly driven by the nature of the data used for feature identification. Since the data used for feature identification is a newswire data which usually contains political and related article in much more details than other topics. Thus contexts containing politics related words generate richer context vectors by virtue of richer feature set. If we had used a corpus compiled from some sports magazines we would expect the sport related features to be much stronger than other features and thus the experiment about Sports personalities to outperform experiments from other categories.

In almost all the experiments where results are significantly above the majority classifier the second order vector representation out-performs first-order representation which can be attributed to its ability of capturing direct as well as the indirect relationships. For example if the word *tea* occurs with *cup* and also with *coffee* but *cup* does not occur with *coffee* still *cup* and *coffee* will be indirectly related by the the virtue of word *tea*.

Though results of some experiments with email clustering are above the majority classifier they are not as good as the name discrimination results, there are several reasons to that. First of all the words contributing towards the creation of context vectors cannot be restricted to the words nearer to the target-word. Thus especially for second order context representation the large amount of noise gets inducted into the context vector. The next important factor is the effect of the writing styles for emails versus news paper article. News paper article have more organized and defined writing style usually that reflects in the vocabulary being used consistently, the tense used and the active or passive phrasing of sentences. Consistency in all these factors in case of news paper articles helps build rich features. Whereas emails tend to be more in-formal and loose with possible occurrences of slangs and regional and very domain specific vocabulary which is not widely known and used. This results in very large set of weak features. Currently we do not filter out the email headers and as a result email-stoplist words like *Subject*, *Reply* etc. are not removed. These words occur in all the emails and thus have heavy and highly skewed vectors and using such vectors for final context vector creation adversely affects the performance. We plan to create a separate email specific stoplist which would filter out all such words.

Table 6 shows that even the simple scheme of labeling the clusters with significant bigrams gives clear hints about the entity that the cluster represents. For example the labels assigned to the clusters in case of the 2-way experiment can certainly help identify that the cluster-0 represents *Tony Blair* and cluster-1 represents *Vladimir Putin*. From the 3-way example we can see that though the *Cluster 0* can be confidently assigned to *Tony Blair* the next two clusters cannot be easily identified based on the labels assigned. This happens because each cluster is labeled with the bigrams selected from the contexts present in the cluster and thus the labels assigned to a cluster depend directly on the clustering performance of the system. Thus in our current example the *Cluster 1* and *Cluster 2* evidently have contexts related to both *Vladimir Putin* as well as *Saddam Hussein*.

9 Future Work

We are currently working on the methods to automatically determine the optimal number of clusters that the contexts should be separated into. This is a non-trivial problem because specifying number of clusters less than the optimal can force merging of unrelated contexts into same cluster while too many clusters can result into artificially fine-grained distinction among related contexts. We are exploring the *GAP Statistics* proposed by Tibshirani, Walther and Hastie[12] for this problem and other cluster stopping rules like Calinski and Harabasz[19] and also a method proposed by Hartigan[18]. All these methods more or less try to maximize the within cluster similarity while minimizing the between cluster similarity or minimize within cluster dispersion and maximize between cluster dispersion. Graphically this translates to locating an knee/elbow (depending upon what is represented across the y-axis) on a graph with the goodness/error plotted across y-axis and the different k values (number of clusters) plotted across x-axis. The problem though is that as the overlap of data increases this knee/elbow losses its sharpness or starts becoming smooth and then deciding which exact value of k is to be used becomes difficult. The *L method* proposed by Salvador and Chan[9] estimates the optimal cluster number by using the property of the knee shaped graph. They note that the leftmost region of this graph (with the sharp slope) and the almost flat region on the right are generally linear. They have found that the x-axis value at which the best-fit lines representing these two linear regions intersect is the optimal number of clusters for that dataset.

We plan to supplement our cluster labeling mechanism by tapping the huge amount of information present over the World Wide Web. We are cautious though that we would have to develop some technique to filter out the noisy data that WWW brings along with the good data.

Second possible approach of extending the cluster-labeling will be using WordNet [8] via WordNet::Similarity [9] as proposed by McCarthy et.al. in [6]

[8] http://wordnet.princeton.edu

[9] http://wn-similarity.sourceforge.net/

The averaging of word-vectors for all the words in the context for generating the context vector in case of second order context representation induces significant amount of noise in the context vectors. Our current solution to this problem is to limit the words whose word-vectors are averaged to the ones which are nearer to target-word cause words in vicinity are more likely to be related to it than the words which are farther from it. This solutions improves the results significantly but this method cannot be used with applications where the contexts do not have any target-word like for example email clustering. Thus we wish to improve on this word selection method by identifying content rich word instead of relying on the position of the word.

Our current methods address the one-to-many kind of problem where the ambiguity caused by same name being used by multiple people is resolved. We plan to test and extend our methods to many-to-one kind of discrimination problems where the same entity is referred by different names for example *Tony Blair* can be addressed as *Prime Minister Blair* or *Mr. Blair* etc.

Finally we also plan to test our system on real life data. Specifically, we plan to experiment with the John Smith Corpus compiled by Bagga and Baldwin and also the name data generated by Mann and Yarowsky.

10 Conclusion

We have shown in this paper that the Word Sense Discrimination techniques proposed by Purandare and Pedersen can be effectively applied to the problems of name discrimination and email clustering. The results obtained with second order context representation out-perform the results obtained using first order context representation. The simple yet elegant cluster labeling technique helps identify the underlying entity that a cluster represents via the *Descriptive* and *Discriminating* labels.

11 Acknowledgment

This research has been supported by a National Science Foundation Faculty Early CAREER Development Award (#0092784). All experiments were done with SenseClusters-v0.69 package.

References

1. Kushmerick N. and Lau T.: Automated Email Activity Management: An Unsupervised Learning Approach *International Conference on Intelligent User Interfaces*, (2005) San Diego, CA
2. Kulkarni A.: Unsupervised Discrimination and Labeling of Ambiguous Names. *The Proceedings of the Student Research Workshop of the 43rd Annual Meeting of the Association of Computational Linguistics*,(2005). Ann Arbor, USA.

3. Pedersen T., Purandare A. and Kulkarni A.: name discrimination by Clustering Similar Contexts. *The Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*, (2005) 226–237. Mexico City, Mexico.
4. Bekkerman R., McCallum A. and Huang G.: Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora. *Center for Intelligent Information Retrieval, Technical Report IR-418*, (2004)
5. Gooi C. and Allan J.: Cross-document coreference on a large scale corpus. *The Proceedings of HLT-NAACL*, (2004) 9-16. Boston, Massachusetts, USA
6. McCarthy D., Koeling R., Weeds J. and Carroll J.: Finding Predominant Word Senses in Untagged Text. *The Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, (2004) 279–286. Barcelona, Spain.
7. Pantel P. and Ravichandran D.: Automatically Labeling Semantic Classes. *The Proceedings of HLT-NAACL*, (2004) 321–328. Boston, MA.
8. Purandare A. and Pedersen T.: Word sense discrimination by clustering contexts in vector and similarity spaces. *The Proceedings of the Conference on Computational Natural Language Learning*, (2004) 41–48. Boston, MA.
9. Salvador, S. and Chan, P.: Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms. *The Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, (2004) 576–584.
10. Mann G. and Yarowsky D.: Unsupervised personal name disambiguation. *The Proceedings of the Conference on Computational Natural Language Learning*, (2003) 33–40. Edmonton, Canada.
11. Zhao Y. and Karypis G.: Evaluation of hierarchical clustering algorithms for document datasets. *In Proceedings of the 11th Conference of Information and Knowledge Management (CIKM)*, (2002) 515-524.
12. Tibshirani R., Walther G. and Hastie T.: Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistics Society (Series B)*, (2000).
13. Bagga A. and Baldwin B.: Entity–based cross–document co–referencing using the vector space model. *The Proceedings of the 17th international conference on Computational linguistics*, (1998) 79–85. Montreal, Quebec, Canada.
14. Schütze H.: Automatic Word Sense Discrimination *Computational Linguistics*, 24(1): (1998) 97–124.
15. Pedersen T. and Bruce R.: Distinguishing word senses in untagged text. *The Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, (1997) 197–207. Providence, RI
16. Lang K.: Newsweeder: Learning to filter netnews *The Proceedings of the Twelfth International Conference on Machine Learning*, (1995) 331-339.
17. Miller G. and Charles W.: Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1), (1991) 1–28
18. Hartigan J.: Clustering Algorithms. John Wiley and Sons, (1975) New York, NY
19. Calinski T. and Harabasz J.: A dendrite method for cluster analysis. *Communications in statistics*, 3, (1974) 1–27.
20. Munkres J.: Algorithm for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5: (1957) 32–38.
21. Kuhn H.: The Hungarian Method for the assignment problem *Naval Research Logistics Quarterly*, 2: (1955) 83–97.