The Effect of Different Context Representations on Word Sense Discrimination in Biomedical Texts

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

ABSTRACT

Unsupervised word sense discrimination relies on the idea that words that occur in similar contexts will have similar meanings. These techniques cluster multiple contexts in which an ambiguous word occurs, and the number of clusters discovered indicates the number of senses in which the ambiguous word is used. One important distinction among these methods is the underlying means of representing the contexts to be clustered. This paper compares the efficacy of first–order methods that directly represent the features that occur in a context with several second–order methods that use a more indirect representation. The experiments in this paper show that second order methods that use word by word co–occurrence matrices result in the highest accuracy and most robust word sense discrimination. These experiments were conducted on MedLine abstracts that contained pseudo–words created by conflating together pairs of MeSH preferred terms to create new ambiguous words. The experiments were carried out with SenseClusters, a freely available open source software package.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Natural Language Processing; I.2.7 [**Natural Language Processing**]: Text Analysis; H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Experimentation

Keywords

natural language processing, semantic ambiguity, word sense discrimination

1. INTRODUCTION

Semantic ambiguity is a persistent problem in Natural Language Processing of general English and in the biomedical domain. Many abbreviations, words, terms, and phrases have multiple possible meanings, and NLP systems are frequently called upon to resolve these ambiguities in some way.

There are two general ways to frame this problem. The first and arguably more common is as *word sense disambiguation*. This is the process of assigning a sense to a word, where that sense is found in a dictionary or other pre–determined sense inventory. For example, given *The patient complained of cold hands*, a word sense disambiguation system might be asked to select between two different senses of *cold*: *the sensation produced by low temperatures* or *a mild viral infection involving the nose and respiratory passages (but not the lungs)*. Word sense disambiguation is frequently approached as a problem in supervised learning, where manually annotated examples of the word in different senses are used to train a classifier. This can also be approached using rule based systems, which again have knowledge of the underlying sense inventory.

The second way of framing the problem of ambiguity is as *word sense discrimination*. In this case there is no pre–existing sense inventory, and the task is to determine if the multiple occurrences of a word in different contexts are being used in different senses. If so, the task is to group together the occurrences of the word that are being used in the same sense. Typically this is approached as a unsupervised learning problem, where multiple occurrences of the same word in different contexts are clustered, and each resulting cluster is viewed as representing a distinct meaning of that word.

For example, a user might have retrieved 300 sentences that include the word *cold*. Some of those sentences may be about the temperature, while others might be about the illness. Rather than trying to identify a precise set of possible senses and then choose from among those, word sense discrimination would separate these sentences into some number of clusters that correspond with the various underlying meanings of *cold* without necessarily labeling those clusters with a definition from a dictionary or sense inventory. In many practical settings such as Web Search and Information Retrieval, the goal really is to organize information for presentation to a user, who can then examine each cluster to see which seems most appropriate for the task at hand.

Word Sense Disambiguation is limited by the coverage of the underlying dictionary or other lexical resource that defines which senses a system can recognize. In general the coverage of dictionaries in terms of vocabulary and senses is far from complete, and in many domains there is new terminology and new usages of words, acronyms and abbreviations that emerge that will not be recorded in dictionaries

IHI'10, November 11–12, 2010, Arlington, Virginia, USA.

for some time. As a result Word Sense Disambiguation systems run the risk of not being terribly flexible or portable, and they may make distinctions in meanings that are either not relevant or are not appropriate for certain tasks.

Given the limitations of word sense disambiguation, this paper instead focuses on word sense discrimination. It compares different representations of the contexts that contain an ambiguous term that are particularly suitable for unsupervised clustering approaches. This paper presents the results of a number of word sense discrimination experiments using contexts from MedLine abstracts. Since the focus of the paper is on the underlying representation, many other settings are held constant across the methods, so as to make the effect of the different representation methods clear.

In order to perform a rigorous and extensive experimental evaluation, thirty ambiguous terms were created artificially by finding all occurrences of two distinct terms, and then replacing those occurrences with a new ambiguous *pseudo–word*. For example, one of the pairs of terms that was conflated was *colon* and *leg*. All occurrences of these two words (and their morphological variants) were converted into an ambiguous term *xyz-ID*, where ID is an integer that uniquely identifies the conflated pair. The contexts containing this new term were discriminated using various methods as implemented in the freely available open source software package SenseClusters.

This paper continues with a discussion of how contexts can be represented in an unsupervised clustering problem, and focuses on first and second–order lexical features. It then goes into a more detailed discussion of these representations and the word sense discrimination methods used in this paper. The method by which new ambiguous words were created for experiments is described, and then the experimental results are presented. This paper features results from six different unsupervised clustering methods applied to 30 different words using three different methods for determining the total number of clusters; a total of 540 different experiments. The paper concludes with an analysis of these results, which includes suggestions for future directions.

2. BACKGROUND

An unsupervised word sense discrimination system takes as input N contexts that each contain a single occurrence of a particular ambiguous word[1]. The word to be discriminated is known as the *target word*. These contexts are grouped into k clusters, where each cluster includes contexts that are similar to each other and the number of clusters k is automatically determined. Each resulting cluster is presumed to correspond to a sense of the ambiguous word, based on the assumption that instances of a word that occur in similar contexts are being used with similar meanings. These methods are said to be unsupervised because they do not use any manually annotated training data, and they are not guided by any underlying knowledge source or human intervention. They simply cluster contexts based on the similarity of the text based on information found within that text.

Given the limited information available to unsupervised methods, the underlying representation of the contexts to be clustered is paramount. As such a number of different techniques have been developed (cf., [2]). The most well known and perhaps most obvious method is to take a *bag of words* approach and let each word in the contexts to be clustered represent a feature, and then contexts that share the most words between them will be judged the most similar, usually via the application of some kind of clustering algorithm (cf., [4]). This is referred to as a *first–order* representation (o1) since contexts are represented directly by the words that occur in them. It is also possible to part of speech tag, parse, or otherwise process the text and use the resulting syntactic or linguistic information as a kind of feature, but the underlying representation remains unchanged.

Put very simply, first–order methods establish the similarity between contexts by finding those contexts that share the largest number of words between them. This can work reasonably well given fairly large amounts of text that includes a certain amount of specialized terminology (as would be found in MedLine abstracts, for example), but can run into difficulties with smaller amounts of text and/or noisier text.

Second–order methods are an alternative that represent the words in a context to be clustered based on some indirect (second–order) information. Many of these methods find their origins in Latent Semantic Indexing ([3]), which was originally applied to problems in Information Retrieval. In fact it also has a long history of use in the biomedical domain (cf., [1]).

The general idea of LSI is to represent the words in a collection of contexts in a word by context matrix[2], where words that occur in approximately the same set of documents are judged to be similar to each other. Next, this matrix is reduced (usually) via Singular Value Decomposition (SVD) which is thought to reduce the noise in the data and make relationships between underlying concepts more clear. The overall goal of LSI is to make it possible to recognize that two very different words might be similar (because they are used in many of the same documents), and thereby improve the ability of a system to provide a user with documents relevant to their query. A user searching for *kidney disease* may well find contexts (e.g., web page snippets or extracts from journal abstracts) that mention *renal failure* of considerable interest.

Schütze [24] developed an extension to LSI that made it possible to carry out word sense discrimination. Rather than creating a term by document matrix, he introduced the idea of using a word by word co–occurrence matrix to represent words. Words that occur with many of the same other words will have similar word vectors in this co–occurrence matrix, and will be judged to be similar. In Schütze's approach, the data from which the co–occurrence matrix is derived can come from some external source, or it could come from the contexts that are being clustered. This paper takes the latter approach, although SenseClusters conveniently supports the use of external data. Regardless of where the co–occurrence data comes from, each word in a context to be clustered is replaced by a vector that represents the words that co-occur with it. The vectors for all the words in a context are averaged together to create a single vector that becomes the second–order word by word co–occurrence representation of that context. After all of the contexts have been represented

[1]While this paper generally uses *word* to refer to the object of discrimination, it should be understood that this is meant to also include terms and phrases interchangeably. In fact the experiments in this paper include many terms and phrases as well as individual words

[2]Note that in the LSI literature this is often referred to as a term by document matrix.

in this way, the clustering algorithm discovers how many different clusters exist in this data, which will reveal the number of senses and which contexts are used in which sense.

Another second–order method that is closely related to LSI is Latent Semantic Analysis (LSA) [12]. Like LSI, LSA uses a word by context representation that is reduced by SVD in order to recognize similar words and concepts. In many respects the main difference between LSI and LSA is in the domain of intended use. LSI is often regarded as an Information Retrieval method, while LSA is often used in applications and experiments relating to psychology, cognitive science, and education (cf., [13]). The underlying methodologies are very similar. While it was not originally applied to word sense discrimination, LSA was recently evaluated on this problem [14]. It has also been employed in the biomedical domain in order to discover missing word senses in a sense inventory [6].

LSI, LSA, and Schütze's approach all build a second–order representation of words or terms. The key step in these methods (that makes them second–order) is that the words in the contexts to be clustered are replaced by something else. In the case of LSI and LSA, each word is replaced by a vector that shows which other contexts in which it has occurred. In Schütze's method each word is replaced by a vector which represents the other words with which it has occurred. Once the words are replaced by the appropriate kind of vector, all of the vectors representing a single context are averaged together to create a single vector representation of that context, and then all the contexts are input to the clustering algorithm. They key motivation behind second–order approaches is to recognize similarity between different words. LSA and LSI rely on the idea that words that occur in approximately the contexts are similar, while Schütze's method is based on the idea that words that occur with with approximately the same set of other words should be considered similar. No doubt there is truth in both points of view.

For example, *aspirin* and *ibuprofen* are different words and would be treated as being just as different as *aspirin* and *blood* by a first–order method. However, they have similar meanings and could reasonably be used in contexts with some of the same other words. Suppose that *pain, inflammation* and *body aches* all occur in contexts with both *aspirin* and *ibuprofen*. As such these would be considered as second–order co–occurrences of *aspirin* and *ibuprofen* and would be used as replacements (intuitively speaking) for those words in Schütze's second–order method. A similar argument can be made for a word by context approach like LSA or LSI, which would recognize that *aspirin* and *ibuprofen* are being used in many of the same contexts, and therefore should be judged to be somewhat similar.

All of first and second–order approaches described above have been implemented in the SenseClusters package, and numerous variations on the original approaches have been developed. These are described in greater detail in a number of publications (e.g., [21], [22]). All of the experiments in this paper are conducted using SenseClusters.

The key distinctions in the different methods that are evaluated in this paper are the underlying representation of the contexts to be clustered. There are three basic schemes that will be described in more detail in the following section: first–order methods that recognizes features that are shared between contexts (o1), the LSI/LSA second–order methods that uses word by context matrices to represent words in context (o2LSA), and Schütze's second–order methods that use word by word co–occurrence matrices to represent words in context (o2SC).

3. METHODOLOGY

This section describes the details of six word sense discrimination methods used in the experiments described later in this paper. The overall framework of each approach is the same - they each accept as input N contexts, where each context includes a particular ambiguous target word. The goal of word sense discrimination is to divide those N contexts into k clusters, where k is automatically determined and is presumed to represent the number of senses in which the target word is being used in these contexts.

The focus of this study is on the underlying representation of the contexts to be clustered. As such various other settings for these methods have been held constant across the experiment. This is not because they are not important or are not potentially interesting, it is simply the goal of this paper to shed light on the consequences of choosing one representation scheme over another. Factors that are held constant across all methods include the clustering algorithm employed, the types of features used and the methods for identifying them, and the methods used for identifying the number of clusters in the data.

3.1 Lexical Features

All six methods use lexical features that are found in the contexts to be clustered. Note that in general it would be possible to obtain information about lexical features from other corpora, and if the number of contexts to be clustered is relatively small this is a very necessary and desirable step to take. However, in these experiments the number of contexts was sufficiently large to simply rely on the contexts themselves to provide this data. Further note that there was no syntactic or grammatical processing of these contexts, nor was stemming or any other kind of normalization performed. As such these methods follow a very knowledge–lean minimum–resource approach which makes them easy to port to new domains and even new languages. However, it is also clear that syntactic information can be used to good advantage with unsupervised word sense discrimination methods (cf., [5]).

3.2 Clustering

All of these methods cluster the contexts with the method of Repeated Bisections using the I2 criterion function and the cosine similarity measure (cf., [9]). Repeated Bisections starts clustering with all the contexts in one cluster, and repeatedly partitions them (in two) in order to optimize the I2 criterion function. This bisection is performed by using the standard k-means clustering with k=2. The I2 criterion function finds the average pairwise similarity between each context in the cluster and the centroid using the cosine measure, and sums these values across all the clusters to find the overall criterion function. This is done for all possible number of clusters from N to 1, where N is the number of contexts.

3.3 Cluster Stopping

Thereafter the optimal number of clusters is automatically identified by finding the point at which the I2 criterion

function reaches a plateau and stops improving. This decision is made by three different cluster stopping methods: PK2, PK3, and the Adapted Gap Statistic. Note that all of these methods require that the data be clustered N times, where the number of clusters k ranges from the total number of contexts N to 1. Then these measures examine various characteristics of these different solutions to determine which value of k is the most appropriate choice.

PK2 is based on Hartigan's approach [8], and takes the ratio of whatever criterion function is being used (in this case I2) over successive pairs of values of k. As this ratio approaches 1 the quality of the clustering solution is no longer improving, so clustering will stop when this ratio is within one standard deviation of 1. PK3 is based on three k values, and computes the ratio of twice the criterion function at k with the sum of the criterion functions at $k-1$ and $k+1$. As this ratio approaches 1 the quality of the clustering solution is not improving, so again we stop if that ratio is within one standard deviation of 1. The Adapted Gap Statistic is based on the Gap Statistic [25], and creates a reference sample of the observed data as if it were only composed of noise (and had no clusters within it). Then, the criterion function for different values of k in the actual observed data is compared with the criterion function for these values of k on the noisy reference sample, in order to identify the value of k where there is the greatest divergence between the observed data and the noisy data. This value of k represents the clustering of the observed data that is least like noise, and is therefore the value selected for k.

Additional details on these cluster stopping methods are available in various sources (e.g., [10], [19], [20]).

3.4 First-Order Context Vectors

First–order context vectors directly represent the features that occur in the contexts to be clustered. In these experiments those features are individual words (unigrams) and bigrams. Any unigram that occurs five or more times in the contexts to be clustered is included as a feature, as long as that word does not appear in a stoplist of approximately 200 common English words that consists of function words such as conjunctions, determiners, articles, prepositions, etc. This same list of stop words is used in all other experiments as well. This method is referred to as o1-uni in later discussion and tables. This is very similar to the traditional *bag of words* representation that has been used in text classification for many years.

Another first–order approach is taken where the features are bigrams, which are ordered pairs of words that occur together more often than would be expected by chance. The order of the words does matter, so *elbow brace* and *brace elbow* are treated as distinct features. The words that make up a bigram may be separated by up to 8 intervening words in order to allow for certain long–distance dependencies to be discovered. For example, *smoker* and *cough* may appear in relatively close proximity to each other, as in *as a smoker he tends to cough* and *the smoker had a bad cough*. All occurrences of *smoker* and *cough* that appear in that order and with at most 8 intervening words between them will be counted as occurrences of the bigram *smoker cough*. However, if either of the words in the bigram are a stopword then it will not be considered as a feature.

The statistical significance of the bigrams that occur five or more times is measured by Fisher's Exact Text (left–sided), and any pair of words with a p–value greater than or equal to 0.99 is included as a feature [18]. This definition of bigram and the technique for identifying those that should be treated as features is used in all other experiments except o1-uni (which is the only method that selects strictly based on frequency). This method is referred to as o1-big in later discussion and tables.

After the set of features are determined (be they bigrams or unigrams) then each context to be clustered is represented as a binary vector that indicates whether or not a particular feature has occurred in that context. This is what is meant by saying that first–order methods directly represent the features that occur in the contexts to be clustered. These vectors are the input to the clustering algorithm, so contexts that share multiple words or bigrams as features are likely to be clustered together. The underlying premise of this approach is that contexts that include a particular target word and that share a number of other words are likely to be using that target word in the same sense. This is certainly a reasonable assumption, although it is also true that there is usually more than one way to express the same idea, and first order methods rely on a fairly high degree of repetition among key discriminating words.

3.5 Second-Order Word by Word

The use of second–order word by word co–occurrences is characteristic of the word sense discrimination approach defined by Schütze [24], and has been implemented in a modified form in SenseClusters.

The second–order method that relies on word by word co-occurrence matrices identifies bigrams in the contexts to be clustered in exactly the same way as o1-big. However, after the bigram features are identified, they are used to define a co–occurrence matrix where the rows represent the first words in the bigram, and the columns represent the second. Since bigrams are order–dependent this matrix is asymmetric. This version of this method is referred to as o2SC in subsequent discussion and tables.

Since the co–occurrence matrix is large and sparse, one variation of o2SC is to reduce the number of columns down to 300 via Singular Value Decomposition (SVD). This process can be viewed (very intuitively and somewhat imprecisely) as a kind of clustering performed on the columns, which in effect combines together some of the second words in the bigrams when they occur with the same first words. The efficacy of SVD remains an open question, and a number of researchers have reported limited or slightly negative effects (e.g., [21], [27]). In order to study this question further, this paper also includes a version of o2SC where SVD has been applied to the word by word co–occurrence matrix prior to building the second order representation. This variation is referred to as o2SC-SVD in subsequent discussion and tables.

Whether SVD is performed or not, the contexts to be clustered are represented by simply checking each word in the context to see if there is a corresponding row entry in the co–occurrence matrix. If there is, then that row is used to replace (in effect) the word in the context. After all the words in the context have been replaced by their corresponding row from the co–occurrence matrix (if one exists) then all of these word vectors are averaged together to create a second order representation of the context to be clustered. There is one averaged vector created per context to be clustered,

and these averaged vectors become the input to the clustering algorithm. At this point the clustering and prediction of the number of clusters proceeds identically to all of the other methods.

The premise of this approach is that there are many words that can be used to express the same idea, and that a first–order method which relies on finding the exact same words in different contexts to assess similarity will potentially miss a great deal that lurks just below the surface of the text. A pair of words that co–occur with the same set of other words (their second–order co-occurrences) but not each other will still be considered similar. Given this it is more likely that similar contexts can be identified, and that the different senses in which an ambiguous target word is being used will be more subtly observed.

3.6 Second-Order Word by Context

The creation of a second–order representation based on word by context co–occurrences (or term by document more generally) is characteristic of both Latent Semantic Indexing (LSI) [3] and Latent Semantic Analysis (LSA) [12].

In this framework the features are simply words (unigrams) that occur 5 or more times in the contexts to be clustered, and that aren't included on the stoplist. This is identical to o1-uni. Then, the contexts are represented in exactly the same way as they are in o1-uni, where each context is converted to a vector that shows which words occur in it. The resulting matrix (which is given as input to the clustering algorithm in o1-uni) is not used for clustering in this method, but is instead transposed to become a word by context co–occurrence matrix. This matrix shows in which contexts each word occurs.

At this point each word in a context to be clustered is replaced by the corresponding row in the word by context co–occurrence matrix (if one exists), and then these word vectors are all averaged together to create a second–order representation of the contexts. This is the same representation technique as used in o2SC, but instead of using a word by word co–occurrence matrix as an underlying representation this method uses a word by context co-occurrence matrix. This is referred to as o2LSA in subsequent discussion and tables.

As is the case with o2SC, the word by context co–occurrence matrix is very large and spare, and so Singular Value Decomposition can be performed prior to substituting the rows from the matrix for the words in the contexts to be clustered. This variation of the method is referred to as o2LSA-SVD in later discussion and tables.

4. EXPERIMENTAL DATA

In order to evaluate the effectiveness of unsupervised word sense discrimination, there must be some gold standard available with which to compare. Such a gold standard would include a large number of contexts manually grouped into appropriate clusters that could be compared with the automatically created clusters.

While it would be possible to manually create a gold standard for such tasks, in general this is a rather time–consuming and error–prone process. While there is a gold standard dataset of 50 ambiguous words available for the biomedical domain ([26]), it has a fairly limited number of contexts per word (100) which make it particularly well suited for supervised learning evaluation. However, it should be noted that a preliminary study of unsupervised word sense discrimination methods was made on this data set by [23]. This work used variants of the first–order unigram and second–order word by word co–occurrence representations described here, but without the benefit of automatic cluster stopping.

Given the lack of any other gold standard data, a collection of ambiguous pseudo–words was automatically created for use in this paper via the following steps:

1. Randomly select 60 terms from the set of MESH preferred terms, and pair them randomly.
2. For each pair of terms, select all the contexts in the MedLine abstracts that contain one or both of those terms (and their simple morphological variants). Let each occurrence of one of these terms be surrounded by up to 50 words before and after, so that each context consists of 100 words where the target word is located (approximately) in the middle.
3. For each pair of preferred terms create a new ambiguous pseudo–word by replacing all occurrences of the two preferred terms with a single ambiguous term xyz-ID, where ID is a unique integer associated with the new term.
4. For each new ambiguous term, randomly select a sample of N contexts such that each underlying "sense" occurs exactly half the time.
5. This process results in 30 samples of contexts, where each sample is made up of 100 word–long contexts with an ambiguous pseudo–word in the middle. This ambiguous word has 2 possible "meanings", where each meaning occurs in 50% of the contexts.

The data created for these experiments following the process outlined above is shown in Table 4. Note that in this table simple regular expressions have been used to show the morphological variants allowed for the terms.

The use of pseudo–words in word sense disambiguation and discrimination is generally accepted as a reasonable (although possibly limiting) alternative to manually annotated data. For example, Schütze used pseudo–words as a part of his original word sense discrimination study [24], and they have also been employed in various unsupervised name discrimination and email categorization studies (e.g., [11], [21]).

Various authors have shown that pseudo–word data can be very effectively used, assuming that the pseudo–words are selected from more restricted sets of categories of relatively monosemous words (e.g., [7], [17]). The danger of an unrestricted selection of pseudo–words is that two relatively ambiguous words (e.g., *line* and *cold*) could be conflated, in which case the underlying meanings of the pseudo–words themselves can confound the discrimination process.

In general these experiments assume that the MeSH preferred terms are relatively unambiguous. While this is not always the case, it appears to be true often enough to result in meaningful results which we review in the sections that follow.

5. EXPERIMENTAL RESULTS

Tables 2, 3, and 4 show the results of applying the six methods of word sense discrimination previously discussed

Table 1: Terms Conflated to Create Ambiguity : each occurs in 50% of contexts

Term1	Term2	ID	contexts
colon(s|ic)?	legs?	1	10,000
patient care	osteoporosis	2	10,000
blood transfusions?	ventricular functions?	3	10,000
randomized controlled trials?	haplotypes?	4	10,000
vasodilations?	bronchoalveolar lavages?	5	10,000
toluenes?	thinking	6	10,000
duodenal ulcers?	clonidines?	7	10,000
myomas?	appetites?	8	5,000
glycolipids?	prenatal care	9	10,000
thoracic surger(y|ies)	cytogenetic analys(is|es)	10	5,000
measles virus(es)?	tissue extracts?	11	5,000
lanthanums?	curiums?	12	5,000
adrenal insufficienc(y|ies)	(recurrent)?laryngeal nerves?	13	5,000
glucokinases?	xeroderma pigmentosums?	14	5,000
polyvinyl alcohols?	polyribosomes?	15	2,000
urethral strictures?	resistance training	16	5,000
cholesterol esters?	premature births?	17	2,000
odontoblasts?	anurias?	18	2,000
brain infarctions?	health resources?	19	2,000
turbinates?	aphids?	20	5,000
cochlear nerves?	(protein)?kinases? inhibitors?	21	2,000
hematemesis	gemfibrozils?	22	2,000
nectars?	work of breathing	23	2,000
fusidic acids?	dicarboxylic acids?	24	2,000
brucellas?	potassium iodides?	25	1,000
walkers?	primidones?	26	2,000
hepatitis(b)?	flavoproteins?	27	5,000
prognathisms?	plant roots?	28	1,000
plant proteins?	(persistent)?vegetative states?	29	2,000
prophages?	porphyrias?	30	5,000

on the 30 conflated pseudo–words. Each table shows the results of a particular cluster stopping method, which predicts the optimal number of clusters (senses) per word. Note that to save space the conflated word pairs are referred to by their ID number in the results tables.

There are two figures of merit shown in these tables - the first is the SenseClusters F-score, which is a percentage that ranges from 0 to 100. A score of 100 means that the clusters created correspond exactly to the gold standard solution. The F-score will assign each discovered cluster to a gold standard cluster on a 1 to 1 basis, and then determine how many contexts found in the discovered cluster are also found in the gold standard cluster to which it has been assigned. The F-score is computed by dividing the number of contexts that are found to be in corresponding discovered and gold standard clusters by the total number of contexts for that ambiguous term. The assignment of discovered to gold standard clusters is made such that the overall agreement is maximized. In fact this method of evaluation reduces to an instance of the Hungarian Algorithm, which SenseClusters solves by using the Munkres algorithm [16].

Note that the F-score penalizes methods that predict the wrong number of clusters rather harshly, since the 1:1 alignment of discovered clusters to gold standard clusters will result in a number of discovered or gold clusters being ignored (if the the number of discovered clusters diverges significantly from the gold standard). The second figure of merit is the value of k, which is the number of clusters predicted by the method. The number of senses in the gold standard data is always 2 for these experiments. In addition to showing these values word by word, the overall average (AVG) and standard deviation (STD) of the F-score and predicted k is presented as well.

Note that if all of the contexts are assigned to just one cluster, then the effect of this evaluation technique is to assign an F-score equal to the percentage of contexts that belong to the most frequent sense in the gold standard. In the experimental data in this paper each ambiguous word has two possible senses, where each sense occurs an equal number of times, so the F-score that results when assigning all contexts to one cluster is 50.

6. DISCUSSION

Tables 2, 3, and 4 show that there is considerable variation in the results from these experiments. Since the main difference among the methods is the underlying representation of the contexts, this reveals a few general points that can be made about these techniques.

6.1 SVD reduces accuracy of results

Perhaps the most noticeable result is a negative one. Singular Value Decomposition (SVD) appeared to do considerable harm whenever it was applied. Regardless of the the cluster stopping method, adding SVD to the o2SC and o2LSA method resulted in a significant decrease in the overall F-score average. For example, for the PK2 stopping

Table 2: F-Score and predicted k by *method* using PK2 cluster stopping

ID	o1-big	k	o1-uni	k	o2SC	k	o2SC-SVD	k	o2LSA	k	o2LSA-SVD	k
1	61.43	6	46.84	4	55.14	3	53.54	2	55.87	3	65.66	2
2	70.65	5	83.27	3	93.44	2	66.08	3	93.69	2	54.62	2
3	70.05	4	81.75	3	96.72	2	62.53	2	79.17	3	42.17	9
4	65.45	5	88.53	3	98.77	2	63.68	3	96.24	2	43.67	11
5	69.88	4	71.99	4	96.10	2	29.17	13	95.33	2	73.80	3
6	67.61	6	75.13	5	94.84	2	61.26	2	91.57	2	43.29	10
7	69.63	4	65.85	4	91.65	2	68.46	3	90.16	2	78.97	2
8	66.43	4	88.16	3	89.38	2	36.36	12	75.61	2	81.92	2
9	67.60	5	87.81	3	99.54	2	92.37	2	98.99	2	79.29	3
10	89.07	3	83.46	3	95.44	2	66.68	2	91.26	2	44.56	8
11	61.71	5	78.82	3	92.36	2	72.58	2	91.56	2	35.46	10
12	46.04	5	46.97	9	61.45	4	44.57	3	49.40	6	64.00	2
13	75.31	4	80.15	3	85.40	2	31.89	11	81.36	2	45.07	6
14	87.36	3	97.76	2	95.98	2	89.44	2	96.04	2	78.52	2
15	57.63	7	61.04	6	86.57	3	44.26	8	53.01	8	46.84	6
16	87.17	3	98.60	2	99.10	2	43.55	9	98.08	2	47.04	8
17	56.95	7	62.35	5	97.70	2	52.01	6	98.30	2	53.75	5
18	65.39	6	89.06	3	99.10	2	50.70	6	96.40	2	58.24	7
19	46.45	8	51.00	5	84.40	2	48.36	7	81.45	2	50.00	1
20	74.31	4	92.70	2	94.30	2	86.92	2	88.90	2	88.60	2
21	62.08	6	69.02	4	95.20	2	49.05	7	93.80	2	60.65	5
22	43.46	8	67.84	4	97.10	2	51.76	7	61.73	5	51.19	4
23	46.97	8	97.25	2	98.15	2	71.61	7	96.20	2	57.42	5
24	53.55	7	58.13	5	77.73	3	46.07	6	56.43	7	48.76	6
25	42.11	10	41.60	11	61.01	4	52.51	3	59.53	4	27.52	14
26	60.41	7	62.44	5	93.40	2	52.23	6	64.13	7	69.59	5
27	85.91	3	96.98	2	96.70	2	86.02	2	95.40	2	44.50	8
28	58.22	8	62.45	6	98.80	2	71.85	3	99.50	2	67.43	4
29	45.07	8	71.04	4	97.55	2	41.40	8	97.65	2	54.99	6
30	85.03	3	99.16	2	99.06	2	38.84	10	98.16	2	79.32	2
AVG	64.63	5.53	75.24	4.00	90.74	2.23	57.52	5.30	84.16	2.90	57.89	5.33
STD	13.52	3.52	16.65	2.00	11.67	0.56	16.75	3.32	16.16	1.76	15.24	3.23

measure the addition of SVD to o2SC dropped the overall F-score from 90.74 to 57.52, and increased the average predicted number of senses from 2.23 to 5.30.

It seems that rather than merging and smoothing redundant information (as SVD is intended to do) it apparently lost important levels of detail while reducing the dimensionality of the co-occurrence matrices. This is perhaps most dramatically illustrated by the results with the Adapted Gap Statistic (Table 4), where for o2-SVD and o2-LSA nearly every word was predicted to have just one sense. This shows rather clearly that there was a fairly extreme and negative loss of information after performing SVD. It is significant that both word by word (o2SC) and word by context (o2LSA) methods were affected in very similar ways, suggesting that the issue is more with SVD than with a particular representation scheme.

6.2 First-order unigrams effective but brittle

The use of bigrams as a first–order feature was not particularly effective (o1-big), and in all cases the use of first–order unigrams (o1-uni) resulted in higher F-scores and predicted values of k closer to the actual value of 2. The motivation for using bigrams rather than unigrams is that they are potentially less ambiguous and more precise than unigrams. However, they are also more sparse, and it would appear that collectively they did not capture as much information as did the unigrams.

The o1-uni method was somewhat brittle however, in that the F-score showed relatively high variance across the different cluster stopping methods. For PK2 the F-score was 75.24, for PK3 it was 84.24, and for Gap it was 87.50. On the other hand, o2SC had F-scores of 90.74, 90.68 and 88.57 for those three cluster stopping methods, suggesting that it is somewhat more robust. o2LSA had F-scores of 84.16, 87.43 and 83.93, which is again relatively consistent and robust.

The variance of the first–order F-scores is not surprising since the methods o1-uni and o1-big require that at least some of the same words must be observed in the contexts to be clustered in order to be regarded as similar. Second–order methods are somewhat less susceptible to variations in vocabulary in the contexts since words are represented by the words with which they co–occur (o2SC) or by the contexts in which they occur (o2LSA).

However, the first order method with unigrams when combined with the Adapted Gap Statistic was extremely effective in identifying the correct number of clusters (see Table 4). Over the 30 words it predicted that 26 of them had 2 clusters and that 4 of them had 1. The overall average of predicted k was 1.87, which was the closest of all the methods to actual k of 2, and the standard deviation for predicted k was the lowest (0.34) of all the methods. This combination

Table 3: F-Score and predicted k by *method* using PK3 cluster stopping

ID	o1-big	k	o1-uni	k	o2SC	k	o2SC-SVD	k	o2LSA	k	o2LSA-SVD	k
1	72.15	3	57.64	3	52.05	4	53.54	2	55.87	3	65.66	2
2	81.92	4	94.05	2	93.44	2	66.08	3	93.69	2	54.62	2
3	84.76	3	81.75	3	96.72	2	62.53	2	79.17	3	67.70	4
4	70.30	4	97.41	2	98.77	2	63.68	3	96.24	2	58.65	2
5	69.88	4	86.79	3	96.10	2	56.43	3	86.97	3	73.80	3
6	67.88	3	95.09	2	94.84	2	61.26	2	91.57	2	71.65	2
7	69.63	4	90.99	2	91.65	2	68.46	3	90.16	2	78.97	2
8	66.43	4	92.40	2	89.38	2	50.00	1	75.61	2	81.92	2
9	87.97	3	99.13	2	99.54	2	92.37	2	98.99	2	79.29	3
10	76.95	4	92.74	2	95.44	2	66.68	2	91.26	2	73.00	2
11	83.73	3	56.74	7	92.36	2	72.58	2	91.56	2	58.39	3
12	36.92	7	54.55	7	54.43	8	44.57	3	56.22	3	64.00	2
13	89.86	3	80.15	3	85.40	2	58.36	2	81.36	2	51.24	3
14	99.19	2	97.76	2	95.98	2	89.44	2	96.04	2	78.52	2
15	57.63	7	73.82	4	86.57	3	70.74	3	68.59	5	65.71	3
16	98.97	2	98.60	2	99.10	2	67.93	5	98.08	2	85.74	2
17	82.87	4	62.35	5	97.70	2	74.85	2	98.30	2	53.26	3
18	90.38	3	98.30	2	99.10	2	56.28	3	96.40	2	58.31	4
19	56.33	6	76.77	3	84.40	2	70.24	3	81.45	2	50.00	1
20	80.29	3	92.70	2	94.30	2	86.92	2	88.90	2	88.60	2
21	63.33	5	95.60	2	95.20	2	88.80	2	93.80	2	50.00	1
22	84.58	3	67.84	4	97.10	2	73.70	2	96.40	2	50.00	1
23	68.77	4	97.25	2	98.15	2	89.90	2	96.20	2	65.37	3
24	66.88	4	58.13	5	77.73	3	59.71	4	84.95	2	78.25	2
25	69.57	4	74.21	4	69.49	3	57.30	2	59.53	4	57.50	2
26	71.32	5	87.85	3	93.40	2	53.64	4	85.00	2	75.99	3
27	85.91	3	96.98	2	96.70	2	86.02	2	95.40	2	82.82	2
28	75.76	3	99.50	2	98.80	2	83.40	2	99.50	2	67.43	4
29	67.28	6	71.04	4	97.55	2	84.75	2	97.65	2	69.80	2
30	74.87	2	99.16	2	99.06	2	72.92	2	98.16	2	79.32	2
AVG	75.08	3.83	84.24	3.00	90.68	2.37	69.44	2.47	87.43	2.30	67.85	2.37
STD	12.86	1.29	14.86	1.41	12.04	1.14	13.14	0.81	12.54	0.69	11.47	0.79

accurate prediction of k combined with small standard deviation is very appealing, and certainly the reasons for this performance should be studied further.

6.3 Conflated words not perfect but useful

The experimental data in this study was created by conflating together two terms to create a new ambiguous term. The terms to be conflated were randomly selected from the MeSH preferred terms, but in general it would be fair to say that these new ambiguous *words* represent fairly coarse distinctions in meaning and that all of the conflated pairs probably result in ambiguous terms of comparable levels of difficulty. There are no extremely fine grained distinctions in the experimental data, although there are some potentially confusable pairs, for example *plant protein* and *persistent vegetative state* (ID-29).

As a refinement to this method of creating experimental data, it would be very useful to generate pairs of terms to conflate that are known to be semantically similar (and therefore representing more subtle distinctions in meaning). This could be done using information gleaned from a thesaurus or via automatic methods of identifying similar and related concepts (cf., [15]). One possible mechanism for doing this would be to randomly select pairs of terms and then measure their semantic similarity or relatedness automatically. Thereafter, chose a certain number of pairs from different ranges of similarity to conflate in order to create a data set of ambiguous words with varying degrees of difficulty.

Also note that the size of these experiments was in a somewhat restricted range from 1,000 to 10,000 contexts per ambiguous word. This seems to be a reasonable number of contexts and is representative of many practical problems. However, there will be different challenges posed with either smaller or larger amounts of data, and certainly those should be studied in future. Finally, all of the conflated–ambiguous terms in this study had only two possible meanings. Future experiments should certainly increase the amount of ambiguity in order to extend these results.

Despite these limitations, the experimental data in this study was in the end randomly created, and there was no tuning of the system to particular words. In general the methods that performed well did so across a range of words and varying numbers of contexts, which gives some confidence that the results will generalize to other settings.

6.4 Second-order methods robust, accurate

The second order methods o2SC and o2LSA both performed quite accurately, and generally had consistent results regardless of the cluster stopping method employed. o2SC was overall more accurate than o2LSA, although the differences were slight. The main advantage of creating

Table 4: F-Score and predicted k by *method* using Adapted Gap Statistic cluster stopping

ID	o1-big	k	o1-uni	k	o2SC	k	o2SC-SVD	k	o2LSA	k	o2LSA-SVD	k
1	49.23	11	52.13	2	52.05	4	50.00	1	55.87	3	50.00	1
2	81.92	4	94.05	2	93.44	2	50.00	1	75.50	4	50.00	1
3	97.31	2	90.80	2	96.72	2	50.00	1	79.17	3	50.00	1
4	41.10	12	97.41	2	98.77	2	50.00	1	87.61	3	50.00	1
5	51.82	9	96.31	2	86.35	3	50.00	1	86.97	3	50.00	1
6	49.62	13	50.00	1	94.84	2	50.00	1	91.57	2	50.00	1
7	46.03	9	90.99	2	79.26	3	50.00	1	81.53	3	50.00	1
8	62.93	6	92.40	2	89.38	2	50.00	1	75.61	2	50.00	1
9	57.37	10	99.13	2	99.54	2	50.00	1	92.36	3	50.00	1
10	89.07	3	92.74	2	95.44	2	50.00	1	80.35	3	50.00	1
11	61.71	5	92.18	2	92.36	2	50.00	1	91.56	2	50.00	1
12	46.24	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1
13	50.96	9	85.76	2	83.19	3	50.00	1	75.35	3	50.00	1
14	74.33	4	97.76	2	88.97	3	50.00	1	85.41	3	50.00	1
15	51.39	9	95.95	2	86.57	3	50.00	1	83.79	3	50.00	1
16	98.97	2	98.60	2	99.10	2	50.00	1	98.08	2	50.00	1
17	56.95	7	98.60	2	97.70	2	50.00	1	98.30	2	50.00	1
18	98.52	2	98.30	2	99.10	2	50.00	1	96.40	2	50.00	1
19	92.41	2	84.95	2	76.41	3	50.00	1	81.45	2	50.00	1
20	43.78	10	92.70	2	94.30	2	50.00	1	88.90	2	50.00	1
21	62.08	6	95.60	2	95.20	2	50.00	1	93.80	2	50.00	1
22	67.04	4	97.00	2	97.10	2	50.00	1	96.40	2	50.00	1
23	98.52	2	97.25	2	98.15	2	50.00	1	96.20	2	50.00	1
24	66.88	4	90.90	2	77.73	3	50.00	1	84.95	2	50.00	1
25	56.17	5	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1
26	60.41	7	50.00	1	93.40	2	50.00	1	50.00	1	50.00	1
27	46.20	12	96.98	2	96.70	2	50.00	1	95.40	2	50.00	1
28	88.98	2	99.50	2	98.80	2	50.00	1	99.50	2	50.00	1
29	67.28	6	97.80	2	97.55	2	50.00	1	97.65	2	50.00	1
30	50.15	9	99.16	2	99.06	2	50.00	1	98.16	2	36.93	11
AVG	65.51	6.23	87.50	1.87	88.57	2.23	50.00	1.00	83.93	2.30	49.56	1.33
STD	18.55	3.52	16.97	0.34	14.20	0.62	0.00	0.00	14.69	0.69	2.35	1.80

second–order representations of context from word by word co–occurrence matrices may be that this is more fine grained information that is a word by context representation. In order for words to be considered co–occurring they must appear within 8 positions of each other, whereas in a word by context representation the information captured is about the contexts that a word occurs in, not the words with which it occurs.

In these experiments the contexts were created from MedLine abstracts and are therefore relatively short and very focused. In less clearly defined texts (as might be found by searching the web for a given keyword) it may be that a word by word representation would be able to pick out distinctions that a word by context might miss due to the larger amount of noise that would likely be found in other sources of data.

7. CONCLUSIONS

This paper presents an experimental comparison of first and second–order methods of representing contexts that include an ambiguous word that is to be discriminated. This comparison includes first–order context vectors, and second–order representations that are created using word by word or word by context co-occurrence matrices. Of these approaches, it is shown that second–order methods have clear advantages over first–order methods, and that second order methods based on word by word co-occurrences result in slightly better accuracy than those based on word by context co-occurrences. These results are generally consistent regardless of what method is used to identify the number of clusters. It is also shown that Singular Value Decomposition (SVD) has a negative effect when used on the word by context and word by word matrices.

8. ACKNOWLEDGMENTS

The experimental data in this paper is derived from MedLine and is freely available contingent upon having an appropriate MedLine license from the National Library of Medicine. Please contact the author for details.

All of the experiments in this paper were conducted using version 1.01 of SenseClusters, which is freely available from http://senseclusters.sourceforge.net. SenseClusters was designed and implemented by Amruta Purandare and Anagha Kulkarni, with support for Latent Semantic Analysis added by Mahesh Joshi. This paper and quite a few others would not have been possible without their very fine work.

The creation of SenseClusters was funded by a National Science Foundation Faculty Early CAREER Development Award (#0092784). The preparation of this paper was supported in part by a grant from the National Library of Medicine, National Institutes of Health (1R01LM009623-01A2).

9. REFERENCES

[1] C. Chute, Y. Yang, and E. Evans. Latent Semantic Indexing of medical diagnoses using UMLS semantic structures. In *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, pages 185–189, 1991.

[2] T. Cohen and D. Widdows. Empirical distributional semantics: methods and biomedical applications. *Journal of Biomedical Informatics*, 42(2):390–405, 2009.

[3] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[4] W. Duan, M. Song, and A. Yates. Fast max-margin clustering for unsupervised word sense disambiguation in biomedical texts. *BMC Bioinformatics*, 10 (Suppl 3), 2009.

[5] J. Fan and C. Friedman. Semantic classification of biomedical concepts using distributional similarity. *Journal of the American Medical Informatics Association*, 14(4):467–477, 2007.

[6] R. Figueroa, Q. Zeng-Treitler, S. Goryachev, and E. Wiechmann. Tailoring vocabularies for NLP in sub-domains: a method to detect unused word sense. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 188–192, San Francisco, 2009.

[7] T. Gaustad. Statistical corpus-based word sense disambiguation: Pseudowords vs. real ambiguous words. In *Companion Volume to the Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL/EACL 2001) – Proceedings of the Student Research Workshop*, pages 61–66, Toulouse, France, 2001.

[8] J. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.

[9] G. Karypis. CLUTO - a clustering toolkit. Technical Report 02-017, University of Minnesota, Department of Computer Science, August 2002.

[10] A. Kulkarni. Unsupervised context discrimination and automatic cluster stopping. Master's thesis, University of Minnesota, July 2006.

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

[12] T. Landauer and S. Dumais. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[13] T. Landauer, D. McNamar, S. Dennis, and W. Kintsch, editors. *Handbook of Latent Semantic Analysis*. Psychology Press, Philadelphia, PA, 2007.

[14] E. Levin, M. Sharifi, and J. Ball. Evaluation of utility of LSA for word sense discrimination. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 77–80, New York City, June 2006.

[15] B. McInnes, T. Pedersen, and S. Pakhomov. UMLS-Interface and UMLS-Similarity : Open source software for measuring paths and semantic similarity. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 431–435, San Francisco, 2009.

[16] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, March 1957.

[17] P. Nakov and M. Hearst. Category-based pseudowords. In *Companion Volume to the Proceedings of HLT-NAACL 2003 - Short Papers*, pages 67–69, Edmonton, Alberta, Canada, May 27 - June 1 2003.

[18] T. Pedersen, M. Kayaalp, and R. Bruce. Significant lexical relationships. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 455–460, Portland, OR, August 1996.

[19] T. Pedersen and A. Kulkarni. Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of the Demonstration Session of the Human Language Technology Conference and the Sixth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 276–279, New York City, June 2006.

[20] T. Pedersen and A. Kulkarni. 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*, pages 111–114, Trento, Italy, April 2006.

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

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

[23] G. Savova, T. Pedersen, A. Purandare, and A. Kulkarni. Resolving ambiguities in biomedical text with unsupervised clustering approaches. Technical report, University of Minnesota Supercomputing Institute Research Report UMSI 2005/80 and CB Number 2005/21, May 2005.

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

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

[26] M. Weeber, J. Mork, and A. Aronson. Developing a test collection for biomedical word sense disambiguation. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 746–750, Washington, DC, 2001.

[27] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. How latent is Latent Semantic Analysis? In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 932–937, Stockholm, 1999.