Evaluating Feature Extraction Methods for Knowledge-based Biomedical Word Sense Disambiguation

Sam Henry **Clint Cuffy** **Bridget T. McInnes**

Department of Computer Science
Virginia Commonwealth University
Richmond, VA 23284 USA

henryst@vcu.edu cuffyca@vcu.edu btmcinnes@vcu.edu

Abstract

In this paper, we present an analysis of feature extraction methods via dimensionality reduction for the task of biomedical Word Sense Disambiguation (WSD). We modify the vector representations in the 2-MRD WSD algorithm, and evaluate four dimensionality reduction methods: Word Embeddings using Continuous Bag of Words and Skip Gram, Singular Value Decomposition (SVD), and Principal Component Analysis (PCA). We also evaluate the effects of vector size on the performance of each of these methods. Results are evaluated on five standard evaluation datasets (Abbrev.100, Abbrev.200, Abbrev.300, NLM-WSD, and MSH-WSD). We find that vector sizes of 100 are sufficient for all techniques except SVD, for which a vector size of 1500 is preferred. We also show that SVD performs on par with Word Embeddings for all but one dataset.

1 Introduction

Word Sense Disambiguation (WSD) is the task of automatically identifying the intended sense (or concept) of an ambiguous word based on the context in which the word is used. Automatically identifying the intended sense of ambiguous words improves the performance of clinical and biomedical applications such as medical coding and indexing for quality assessment, cohort discovery (Plaza et al., 2011; Preiss and Stevenson, 2015), and other secondary uses of data such as information retrieval and extraction (Stokoe et al., 2003), and question answering systems (Ferrández et al., 2006). These capabilities are becoming essential tasks due to the growing amount of information available to researchers, the transition of health care documentation towards electronic health records, and the push for quality and efficiency in health care.

Previous methods using distributional context vectors have been shown to perform well for the task of WSD. One problem with distributional vectors is the sparseness of the vectors and noise (defined here as information that does not aid in the discrimination between word senses). Word embeddings have become an increasingly popular method to reduce the dimensionality of vector representations, and have been shown to be a valuable resource for NLP tasks including WSD (Sabbir et al., 2016).

Prior to word embeddings, (Deerwester et al., 1990) proposed Latent Semantic Indexing (LSI) which reduces dimensionality using the factor analysis technique, singular value decomposition (SVD). When performing SVD, some information is lost. Intuitively the lost information is noise, and its removal causes the similarity and non-similarity between words to be more discernible (Pedersen, 2006).

Similar to SVD is principal component analysis (PCA). PCA transforms the vectors into a new basis of *principal components*, which are created by orthogonal linear combinations of the original features. Each principal component captures as much variance in the data as possible while maintaining orthogonality. Dimensionality reduction is performed by removing principal components that capture little variance.

In this paper, we evaluate the performance of word embeddings, SVD, and PCA for dimensionality reduction for the task of knowledge-based WSD. Explicit vectors are trained on Medline abstracts and performance is evaluated on five reference standards. Specifically, the contributions of this paper are an analysis of:

- Vector Representation: SVD, PCA, and word embeddings using continuous bag of words (CBOW) and skip-gram are evaluated as dimensionality reduction techniques applied to the task of knowledge-based WSD. Evaluation is performed on several standard evaluation datasets, and compared against explicit co-occurrence vectors as a baseline.

- Dimensionality: the dimensionality of the reduced vectors is a parameter, and the value can effect performance. We evaluate each vector representation's performance at dimensionalities of 100, 200, 500, 1000, and 1500.

2 Related Work

Existing biomedical WSD methods can be classified into three groups: unsupervised (Brody and Lapata, 2009; Pedersen, 2010), supervised (Zhong and Ng, 2010; Stevenson et al., 2008), and knowledge-based methods (Navigli et al., 2011). Unsupervised methods use the distributional characteristics of an outside corpus and do not rely on sense information or a knowledge source (Pedersen, 2006).

Supervised methods use machine learning algorithms to assign senses to instances containing the ambiguous word. Although supervised methods have the best performance, they require training data for each target word to be disambiguated. Whether this is done manually or automatically, it is infeasible to create such data on a large scale. Recently, (Sugawara et al., 2015) created a supervised system that uses word2vec word embeddings as input to a support vector machine classifier. They compare the word vectors generated by word2vec with the word vectors generated by SVD, and show that word2vec slightly outperforms SVD with vector dimensionality of 300.

Knowledge-based methods do not use manually or automatically generated training data, but instead use information from an external knowledge source (e.g. taxonomy). These knowledge-based methods can be classified into two categories, graph-based and vector-based approaches. Here, we focus on vector-based approaches as it relates to this research.

(Humphrey et al., 2006) introduce a vector-based method that assigns a sense to a target word by first identifying its semantic type with the assumption that each possible sense has a distinct semantic type. In this method, semantic type (st-) vectors are created for each possible semantic type. The st-vectors consist of binary values for each one word term in the United Medical Language System (UMLS); a one if that word has a sense of the semantic type, else a zero. A target word (tw-) vector is created using the words surrounding the target word. The cosine of the angle between the tw-vector and each of the st-vectors is calculated and the sense whose st-vector is closest to the tw-vector is assigned to the target word. The limitation of this method is that two possible senses may have the same semantic type. For example, the term *cortices* can refer to either the cerebral cortex (C0007776) or the kidney cortex (C0022655), both of which have the same semantic type, "Body Part, Organ, or Organ Component". Analysis of the 2009 Medline data [1] shows that there are 1,072,902 terms in Medline that exist in the UMLS of which 35,013 are ambiguous and 2,979 have two or more senses with the same semantic type. This indicates that approximately 12% of the ambiguous words cannot be disambiguated using the knowledge-based methods discussed above, and another method is required.

(Jimeno-Yepes et al., 2011) attempt to address this limitation by introducing two methods, MRD and 2-MRD. In these methods a sense vector (s-vector) is created for each possible sense of a target word using the definition information from the UMLS. A target word (tw-) vector is created using the words surrounding the target word. The cosine of the angle between the tw-vector and each of the s-vectors is calculated and the sense whose s-vector is closest to the tw-vector is assigned to the target word. The MRD method uses the words within the definition weighted based on their occurrence statistics across definitions in the UMLS. The 2-MRD method (discussed more fully in Section 3) uses second-order context vectors to represent the concept's definition.

(Pakhomov et al., 2016) and (Tulkens et al., 2016) explore using the 2-MRD method in conjunction with word embeddings, and evaluate their performance with varying training corpora. Their results are promising, however evaluation is limited to a single dataset (MSH-WSD), vector size is not varied, and they do not compare performance with different word2vec models.

[1] http://mbr.nlm.nih.gov/index.shtml

3 Method

We modify the vector representations of the 2-MRD WSD algorithm using four different vector representations: SVD, PCA, and word embeddings using continuous bag of words (CBOW) and skip-gram. Explicit vectors are word-by-word co-occurrence vectors, and are used as a baseline. The disadvantage of explicit vectors is that the word-by-word co-occurrence matrix is sparse and subject to noise introduced by features that do not distinguish between the different senses of a word. The goal of the dimensionality reduction techniques is to generate vector representations that reduce this type of noise. Each method is described in detail here.

3.1 2-MRD Algorithm

In this section we describe the 2-MRD WSD algorithm at a high level: a vector is created for each possible sense of an ambiguous word, and the ambiguous word itself. The appropriate sense is then determined by computing the cosine similarity between the vector representing the ambiguous word and each of the vectors representing the possible senses. The sense whose vector has the smallest angle between it and the vector of the ambiguous word is chosen as the most likely sense.

To create a vector for a possible sense, we first obtain a textual description of sense from the UMLS, which we refer to as the *extended definition*. Each sense, from our evaluation set, was mapped to a concept in the UMLS, therefore, we use the sense's definition plus the definition of its parent/children and narrow/broader relations and associated synonymous terms. After the extended definition is obtained, we create the second-order vector by first creating a word by word co-occurrence matrix in which the rows represent the content words in the extended definition, and the columns represent words that co-occur in Medline abstracts with the words in the definition. Each word in the extended definition is replaced by its corresponding vector, as given in the co-occurrence matrix. The centroid of these vectors constitutes the second order co-occurrence vector that is used to represent the sense.

The second-order co-occurrence vector for the ambiguous word is created in a similar fashion, only rather than using words in the extended definition, we use the words surrounding the word in the instance. Second-order co-occurrence vectors were first described by (Schütze, 1998) and extended by (Purandare and Pedersen, 2004) and (Patwardhan and Pedersen, 2006) for the task of word sense discrimination. Later, (McInnes et al., 2011; Jimeno-Yepes et al., 2011) adapted these vectors for the task of disambiguation rather than discrimination.

3.2 Singular Value Decomposition

Singular Value Decomposition (SVD), used in Latent Semantic Indexing, is a factor analysis technique to decompose a matrix, M into a product of three simpler matrices, such that $M = U \cdot \Sigma \cdot V^T$. The matrices U and V are orthonormal and Σ is a diagonal matrix of eigenvalues in decreasing order. Limiting the eigenvalues to d, we can reduce the dimensionality of our matrix to $M_d = U_d \cdot \Sigma_d \cdot V_d^T$. The columns of U_d correspond to the eigenvectors of M_d. Typically this decomposition is achieved without any loss of information. Here though, SVD reduces a word-by-word co-occurrence matrix from thousands of dimensions to hundreds, and therefore the original matrix cannot be perfectly reconstructed from the three decomposed matrices. The intuition is that any information lost is noise, the removal of which causes the similarity and non-similarity between words to be more discernible (Pedersen, 2006).

3.3 Principal Component Analysis

Principal Component Analysis (PCA) is similar to SVD, and is commonly used for dimensionality reduction. The goal of PCA is to map data to a new basis of orthogonal principal components. These principal components are linear combinations of the original features, and are ordered by their variance. Therefore, the first principal components capture the most variance in the data. Under the assumption that the dimensions with the most variance are the most discriminative, dimensions with low variance (the last principal components) can safely be removed with little information loss.

PCA may be performed in a variety of ways, however the implementation we chose makes the parallels between PCA and SVD clear. First the co-occurrence matrix, M is centered to produce the matrix C. Centering consists of subtracting the mean of each column from values in that column. PCA is sensitive to scale, and this prevents the variance of features with higher absolute counts from dominating. Mathematically, this allows us to compute the principal components using SVD on C. This is because C^TC is proportional to the covariance matrix of M, and is used in the calculation of SVD. Applying SVD to C, such that $C = U \cdot \Sigma \cdot V^T$, the principal components are obtained by the product of U and Σ (e.g. $M_{PCA} = U \cdot \Sigma$). For dimensionality reduction all but the first d columns of M_{PCA} are removed. This captures as much variation in the data with the fewest possible dimensions.

3.4 Word embeddings

The word embeddings method, proposed by (Mikolov et al., 2013), is a neural network based approach that learns a representation of a word-word co-occurrence matrix. The basic idea is that a neural network is used to learn a series of weights (hidden layer with in the neural network) that either maximizes the probability of a word given the surrounding context, referred to as the continuous bag of words (CBOW) approach, or to maximize the probability of the context given a word, referred to as the Skip-gram approach;

For either approach, the resulting hidden layer consists of a matrix where each row represents a word in the vocabulary and columns a word embedding. The basic intuition behind this method is that words closer in meaning will have vectors closer to each other in this reduced space.

4 Data

4.1 Training Data

We develop our vectors using co-occurrence information from Medline [2]. Medline is a bibliographic database containing around 23 million citations to journal articles in the biomedical domain and is maintained by National Library of Medicine. The 2015 Medline Baseline encompasses approximately 5,600 journals starting from 1948, and contains 22,775,609 citations, of which 13,835,206 contain abstracts. In this work, we use Medline titles and abstracts from 1975 to present day to generate word embeddings, and to generate the co-occurrence matrix of explicit vectors that is the input into SVD and PCA. Prior to 1975, only 2% of the citations contained an abstract.

4.2 Evaluation Data

We evaluate using several standard WSD evaluation datasets which include the following.

Abbrev. The Abbrev dataset [3] developed by Stevenson, et al. (Stevenson et al., 2009) contains examples of 300 ambiguous abbreviations found in MEDLINE that were initially presented by (Liu et al., 2001). The data set was automatically recreated by identifying the abbreviations and long-forms (unabbreviated terms) in MEDLINE abstracts, and replacing the long-form in the abstract with its abbreviation. The abbreviations' long-forms were manually mapped to concepts in the UMLS by Stevenson, et al. Each abstract contains approximately 216 words. The datasets consist of a set of 21 different ambiguous abbreviations for which the number of labeled instances of those abbreviations varies. Abbrev.100 contains 100 instances, Abbrev.200 contains 200, and Abbrev.300 contains 300 labeled instances. Two abbreviations contain less than 200 instances, and three abbreviations contain less than 300 instances, and are omitted from Abbrev.200 and Abbrev.300 respectively. The average number of long-forms per abbreviation is 2.6 and the average majority sense across all subsets is 70%.

NLM-WSD. The National Library of Medicine's Word Sense Disambiguation (NLM-WSD) dataset [4] developed by (Weeber et al., 2001) contains 50 frequently occurring ambiguous words from the 1998 MEDLINE baseline. Each ambiguous word in the NLM-WSD dataset contains 100 ambiguous instances randomly selected from the abstracts totaling to 5,000 instances. The instances were manually disambiguated by 11 evaluators who assigned the ambiguous word to a concept (CUI) in the UMLS, or assigned the concept as "None" if none of the possible concepts described the term. The average number of senses per term is 2.3, and the average majority sense is 78%.

MSH-WSD. The National Library of Medicine's MSH Word Sense Disambiguation (MSH-WSD) dataset [5] developed by (Jimeno-Yepes et al., 2011) contains 203 ambiguous terms and abbreviations from the 2010 MEDLINE baseline. Each target word contains approximately 187 instances, has 2.08 possible senses, and has a 54.5% majority sense. Out of 203 target words, 106 are terms, 88 are abbreviations, and 9 have possible senses that are both abbreviations and

[2] http://mbr.nlm.nih.gov/Download/index.shtml

[3] http://nlp.shef.ac.uk/BioWSD/downloads/corpora

[4] http://wsd.nlm.nih.gov

[5] http://wsd.nlm.nih.gov

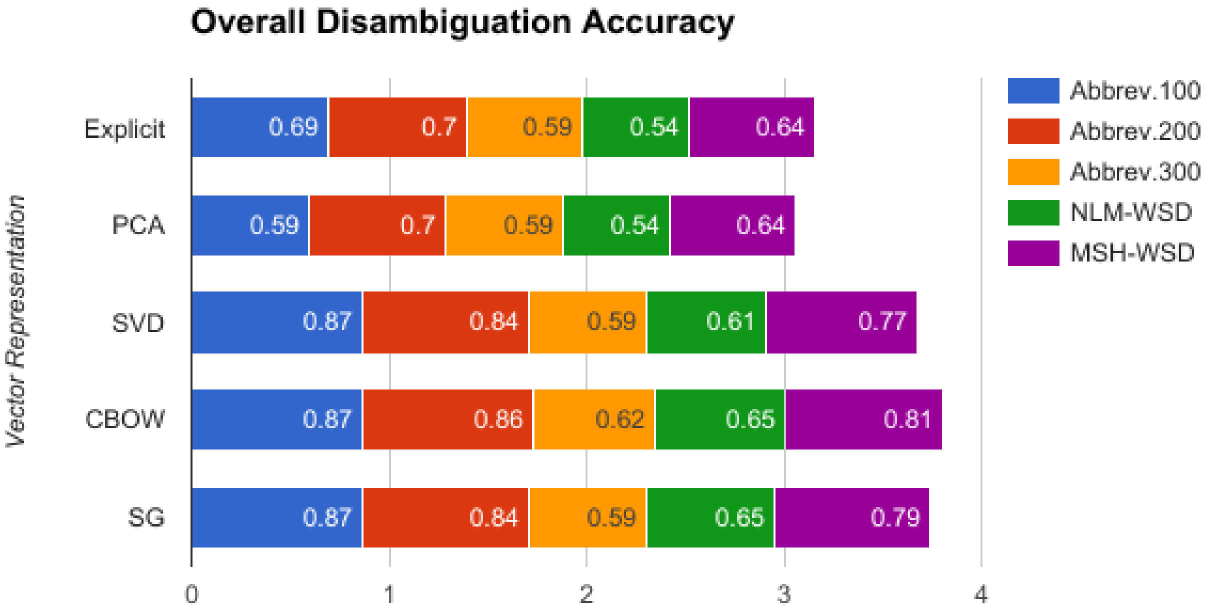

Figure 1: Comparison between accuracy of vector representations on WSD datasets

terms. For example, the target word *cold* has the abbreviation *Chronic Obstructive Lung Disease* as a possible sense, as well as the term *Cold Temperature*. The total number of instances is 37,888.

5 Experimental Framework

We used the following packages to obtain our vector representations:

[1] Explicit Representation: We used the Text::NSP packaged developed by (Pedersen et al., 2011). We used a windows size of 8, a frequency cutoff of 5, and removed stopwords.

[2] Singular Value Decomposition: We ran the MATLAB R2016b implementation of sparse matrix SVD (svds) on the explicit representation matrix, and used each row of the resulting U matrix as a reduced vector.

[3] Principal Component Analysis: We centered the explicit representation matrix, and used the MATLAB R2016b implementation of sparse matrix SVD (svds) on the centered matrix to obtain the U and Σ matrices. The reduced vectors are obtained from the product of U and Σ.

[4] Word Embeddings: We used the $word2vec$ package developed by (Mikolov et al., 2013) for the continuous-bag-of-words (CBOW) and skip-gram word embedding models with a window size of 8, a frequency cutoff of 5, and default settings for all other parameters.

We use the Word2vec::Interface package [6] version 0.03 to obtain the disambiguation accuracy for each of the WSD datasets. The differences between the means of disambiguation accuracy were tested for statistical significance using pair-wise Students t-test.

6 Results and Analysis

6.1 Results

Figure 1 compares the performance of each vector representation technique, and shows the best results (best among all dimensionalities tested) of each of the vector representations on the WSD datasets. Explicit refers to the co-occurrence vector without dimensionality reduction, PCA refers to the principal component analysis representation, SVD refers to singular value decomposition representation, CBOW refers to the word embeddings continuous bag of words representation and SG refers to the word embeddings skip gram representation. The colored bars show results for individual datasets, and the total length shows the sum of accuracies for all datasets.

[6] http://search.cpan.org/dist/Word2vec-Interface/

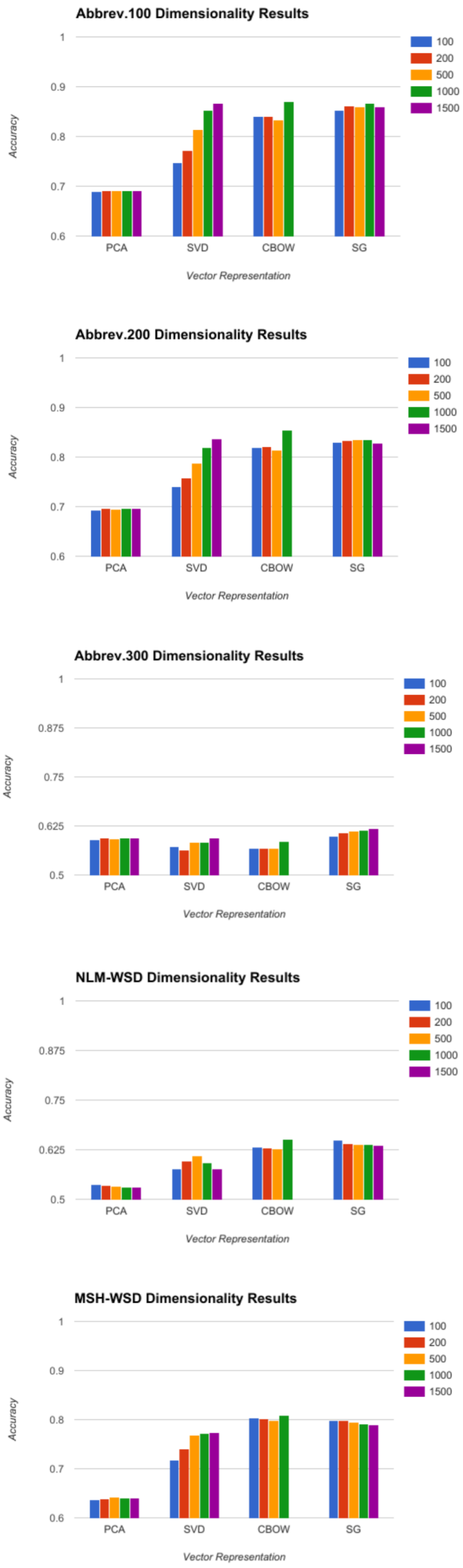

Figure 2: Effect of dimensionality on accuracy

Abbrev.100				
	PCA	SVD	CBOW	SG
explicit	0.65	0.0008	0.0015	0.0006
PCA		0.0007	0.0013	0.0005
SVD			0.94	0.97
CBOW				0.93
Abbrev.200				
	PCA	SVD	CBOW	SG
explicit	0.29	0.006	0.0047	0.0045
PCA		0.005	0.0042	0.0037
SVD			0.56	0.93
CBOW				0.60
Abbrev.300				
	PCA	SVD	CBOW	SG
explicit	1.0	1.0	0.41	0.63
PCA		1.0	0.41	0.63
SVD			0.29	0.21
CBOW				0.08
NLM-WSD				
	PCA	SVD	CBOW	SG)
explicit	0.35	0.10	0.0062	0.0127
PCA		0.087	0.0042	0.009
SVD			0.2489	0.2993
CBOW				0.66
MSH-WSD				
	PCA	SVD	CBOW	SG
explicit	0.37	0.0001	0.0001	0.0001
PCA		0.0356	0.0005	0.0001
SVD			0.0005	0.0346
CBOW				0.056

Table 1: The p-values using Student's pairwise *t-test*. Each table corresponds to a different dataset, each row and column a different dimensionality reduction technique.

The Abbrev.100, Abbrev.200, and Abbrev.300 results show that SVD (0.87/0.84/0.62), CBOW (0.87/0.86/0.62), and SG (0.87/0.84/0.59) obtained a statistically higher overall disambiguation accuracy ($p \leq 0.05$) than explicit (0.69/0.70/0.59) and PCA (0.59/0.70/0.59), while the difference between their respective results was not statistically significant. The NLM-WSD results also show that SVD (0.61), CBOW (0.65), and SG (0.65) obtained a statistically higher disambiguation accuracy than explicit (0.54) and PCA (0.54), while the difference between their respective results was not statistically significant. The MSH-WSD results show a statistically significant difference ($p \leq 0.05$) between explicit (0.64), PCA (0.64), SVD (0.77), CBOW (0.81), and SG (0.79) except for Explicit and PCA. Table 1 shows the p-$values$ between the vector representations for each of the datasets.

Figure 2 shows the effects of dimensionality on

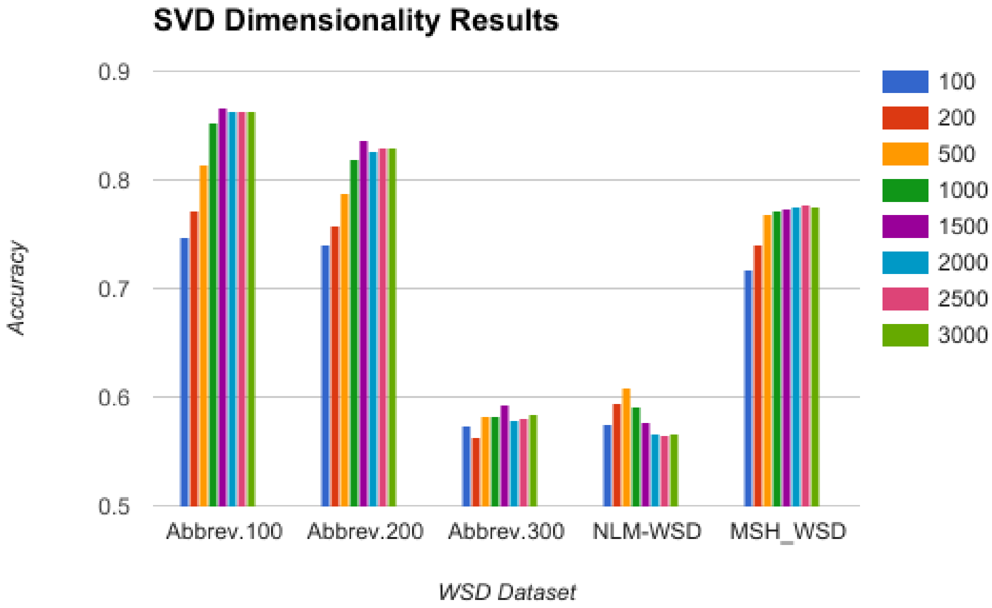

Figure 3: Effect of dimensionality on the accuracy of SVD

disambiguation accuracy of PCA, SVD, CBOW [7], and SG for each of the datasets for dimensionality reduction of d = 100, 200, 500, 1000 and 1500. The PCA, CBOW, and SG all show little change in accuracy as the dimensions vary. This indicates lower dimensional vector representations are sufficient for these techniques. SVD on the other hand shows for all of the datasets except NLM-WSD, an increase in accuracy as dimensionality increases. To discover an upper bound on dimensionality and performance, we continued to increase the dimensions of SVD up to 3000. Results are shown in Figure 3, and indicate that after $d = 1500$ there are not significant gains in accuracy, indicating that a dimensionality of 1500 is sufficient for SVD.

6.2 Analysis

This study indicates that SVD performs on par with word embeddings for most datasets. This is exciting because the co-occurrence matrix that is the input for SVD can be easily modified to hopefully increase performance. The word embeddings algorithms use a neural network approach which can approximate any function, but does not provide any insights about the features being approximated; instead accuracy gains are often achieved by increasing the amount of training data.

One disadvantage of SVD is that it, unlike word embeddings, may not be scalable to massive corpora. Since we are using the majority of MEDLINE, we feel that SVD is sufficient, and previous studies (Pakhomov et al., 2016; Pedersen et al., 2007) have shown that beyond 100 million tokens little performance gains can be achieved.

Surprisingly the results showed that PCA did not obtain a higher accuracy than the explicit co-occurrence vector. We believe this is a result of centering the matrix, and believe that in language absolute counts are important. When the matrix is centered, only relative counts are considered. This could create a situation where infrequently used words have distributions similar to commonly used words, adversely effecting results.

With respect to dimensionality, we found that low vector dimensionality ($d = 100$) is sufficient for CBOW and SG, but that a higher dimensionality ($d = 1500$) obtained better results with SVD. In addition, we found that although PCA is commonly used for dimensionality reduction in many fields, it does not improve results for WSD.

We found that CBOW and SG achieve approximately the same accuracy which is important because SG takes much longer to compute (our rough estimates indicate that SG takes between 5 and 9 times as long to train).

6.3 Comparison with previous work

Recently, word embeddings have been used for word sense disambiguation in the biomedical do-

[7] CBOW crashed due to memory constraints for $d = 1500$

Table 2: Comparison with Previous Work on MSH-WSD

Method	Medline	MIMIC-III	BioASQ	Fairview	PMC
(Pakhomov et al., 2016) (CBOW)				0.72	0.78
(Tulkens et al., 2016) (SG)	0.80	0.69	0.84		
SG	0.81				
CBOW	0.79				
SVD	0.77				
PCA	0.64				
Explicit	0.64				

main. (Tulkens et al., 2016) evaluated the skip gram model on the MSH-WSD dataset with three different sets of training data: a subset of Medline abstracts, the MIMIC-III corpus of clinical notes, and BioASQ Medline abstracts. (Pakhomov et al., 2016) evaluated CBOW on the MSH-WSD dataset using two different types of training data: clinical (clinical notes from the Fairview Health System) and biomedical (PMC corpus).

Table 2 shows the comparison between the previous works' reported results and our current results. The table shows that our skip gram and CBOW results are similar to those reported by both (Tulkens et al., 2016) (0.80 versus 0.81) and (Pakhomov et al., 2016) (0.78 versus 0.79) respectively. We believe that the small variations in accuracy are due to the difference in training data. The table also shows that SVD performs on par with previous word embeddings results.

6.4 Limitations

This study focused on comparing vector representations and the effects of dimensionality for WSD. We did not experiment with other parameters, such as window size, cut-off level, and sampling parameters. We also limited our technique to the 2-MRD WSD algorithm. This is a well known algorithm that has been shown to perform well in the past, and allows comparison between similar papers. These vector representations can be used for other WSD algorithms as well, including supervised or "distantly supervised" approaches (Sabbir et al., 2016) which may achieve higher accuracies, but are limited to pre-labeled or preprocessed datasets.

7 Conclusions and Future Work

In this study we analyzed the performance of vector representations using the dimensionality reduction techniques of word embeddings (continuous bag of words and skip-gram), singular value decomposition (SVD), and principal component analysis (PCA) on five evaluation standards (Abbrev.100, Abbrev.200, Abbrev.300, NLM-WSD, MSH-WSD). We used explicit co-occurrence vectors as the baseline. The results show that word embeddings and SVD outperform PCA and explicit representations for all datasets. PCA does not increase performance over explicit, and word embeddings are significantly different from SVD on just a single dataset (MSH-WSD). The method (CBOW versus SG) in which word embeddings are generated makes no statistically significant difference in WSD results. We also varied the dimensionality of the vectors to 100, 300, 500, 1000, and 1500. We found that the smallest dimensionality of 100 is sufficient for all vector representations except SVD. For SVD we found that increasing dimensionality does increase performance, and continued to increase the dimensionality to 2000, 2500, and 3000. Accuracy stopped increasing at 1500, indicating that a dimensionality of 1500 is sufficient for SVD.

An interesting result of this research is that SVD performs essentially on par with word embeddings. In the future we hope to increase the accuracy of SVD by modifying the co-occurrence matrix that is input into SVD to include incorporating knowledge sources (such as the UMLS) for term expansion by capturing co-occurrences with synonymous terms, and creating a UMLS concept (CUI) co-occurrence matrix. Additionally, this concept co-occurrence matrix can then be augmented to exploit the hierarchical structure of the UMLS. Using a matrix of similarities or association scores may also be interesting. Independent from how vectors are generated, we could use similarity metrics other than cosine, similar to those from (Sabbir et al., 2016) that incorporate both magnitude and orientation.

References

S. Brody and M. Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 103–111.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41(6):391.

S. Ferrández, S. Roger, A. Ferrández, A. Aguilar, and P. López-Moreno. 2006. A new proposal of word sense disambiguation for nouns on a question answering system. *Advances in Natural Language Processing. Research in Computing Science* 18:83–92.

S.M. Humphrey, W.J. Rogers, H. Kilicoglu, D. Demner-Fushman, and T.C. Rindflesch. 2006. Word sense disambiguation by selecting the best semantic type based on journal descriptor indexing: Preliminary experiment. *Journal of the American Society for Information Science and Technolology* 57(1):96–113.

A. Jimeno-Yepes, B.T. McInnes, and A. Aronson. 2011. Exploiting mesh indexing in medline to generate a data set for word sense disambiguation. *BMC Bioinformatics* .

H. Liu, Y.A. Lussier, and C. Friedman. 2001. Disambiguating ambiguous biomedical terms in biomedical narrative text: an unsupervised method. *Journal of Biomedical Informatics* 34(4):249–261.

B.T. McInnes, T. Pedersen, Y. Liu, S.V. Pakhomov, and G.B Melton. 2011. Using second-order vectors in a knowledge-based method for acronym disambiguation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 145–153.

T. Mikolov, I. Sutskever, Kai Chen, G.S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

R. Navigli, S. Faralli, A. Soroa, O. de Lacalle, and E. Agirre. 2011. Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, pages 2317–2320.

S.V. Pakhomov, G. Finley, R. McEwan, Y. Wang, and G.B. Melton. 2016. Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics* 32(23):3635–3644.

S. Patwardhan and T. Pedersen. 2006. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*. Trento, Italy, pages 1–8.

T. Pedersen. 2006. Unsupervised corpus-based methods for wsd. *Word sense disambiguation: algorithms and applications* pages 133–166.

T. Pedersen. 2010. The Effect of Different Context Representations on Word Sense Discrimination in Biomedical Texts. In *Proceedings of the 1st ACM International Health Informatics Symposium*. pages 56–65.

T. Pedersen, S. Banerjee, B.T. McInnes, S. Kohli, M. Joshi, and Y. Liu. 2011. The Ngram statistics package (Text::NSP): A flexible tool for identifying ngrams, collocations, and word associations. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*. Association for Computational Linguistics, pages 131–133.

T. Pedersen, S.V. Pakhomov, S. Patwardhan, and C.G. Chute. 2007. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics* 40(3):288–299.

L. Plaza, A.and Díaz A. Jimeno-Yepes, and A.R. Aronson. 2011. Studying the correlation between different word sense disambiguation methods and summarization effectiveness in biomedical texts. *BMC bioinformatics* 12(1):355.

J. Preiss and M. Stevenson. 2015. The effect of word sense disambiguation accuracy on literature based discovery. In *Proceedings of the ACM Ninth International Workshop on Data and Text Mining in Biomedical Informatics*. ACM, pages 1–1.

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

AKM Sabbir, A.J. Yepes, and R. Kavuluru. 2016. Knowledge-based biomedical word sense disambiguation with neural concept embeddings and distant supervision. *arXiv preprint arXiv:1610.08557* .

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

M. Stevenson, Y. Guo, A. Al Amri, and R. Gaizauskas. 2009. Disambiguation of biomedical abbreviations. In *Proceedings of the ACL BioNLP Workshop*. pages 71–79.

M. Stevenson, Y. Guo, R. Gaizauskas, and D. Martinez. 2008. Disambiguation of biomedical text using diverse sources of information. *BMC Bioinformatics* 9(Suppl 11):11.

C. Stokoe, M.P. Oakes, and J. Tait. 2003. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, pages 159–166.

H. Sugawara, H. Takamura, R. Sasano, and M. Okumura. 2015. Context representation with word embeddings for wsd. In *International Conference of the Pacific Association for Computational Linguistics*. Springer, pages 108–119.

S. Tulkens, S. Šuster, and W. Daelemans. 2016. Using distributed representations to disambiguate biomedical and clinical concepts. *arXiv preprint arXiv:1608.05605* .

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

Z. Zhong and H.T. Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*. Association for Computational Linguistics, pages 78–83.