Domain Adapted Word Embeddings for Improved Sentiment Classification

Prathusha K Sarma, Yingyu Liang and William A Sethares

University of Wisconsin-Madison
{kameswarasar,sethares}@wisc.edu,
yliang@cs.wisc.edu

Abstract

Generic word embeddings are trained on large-scale generic corpora; *Domain Specific* (DS) word embeddings are trained only on data from a domain of interest. This paper proposes a method to combine the breadth of generic embeddings with the specificity of domain specific embeddings. The resulting embeddings, called *Domain Adapted* (DA) word embeddings, are formed by first aligning corresponding word vectors using Canonical Correlation Analysis (CCA) or the related nonlinear Kernel CCA (KCCA) and then combining them via convex optimization. Results from evaluation on sentiment classification tasks show that the DA embeddings substantially outperform both generic, DS embeddings when used as input features to standard or state-of-the-art sentence encoding algorithms for classification.

1 Introduction

Generic word embeddings such as Glove and word2vec (Pennington et al., 2014; Mikolov et al., 2013) which are pre-trained on large bodies of raw text, have demonstrated remarkable success when used as features for supervised learning problems. There are, however, many applications with domain specific vocabularies and relatively small amounts of data. The performance of generic word embeddings in such applications is limited, since word embeddings pre-trained on generic corpora do not capture domain specific semantics/knowledge, while embeddings learned on small data sets are of low quality.

A concrete example of a small-sized domain specific corpus is the Substances User Disorders (SUDs) data set (Quanbeck et al., 2014; Litvin et al., 2013), which contains messages on discussion forums for people with substance addictions. These forums are part of mobile health intervention treatments that encourages participants to engage in sobriety-related discussions. The goal of such treatments is to analyze content of participants' digital media content and provide human intervention via machine learning algorithms. This data is both domain specific and limited in size. Other examples include customer support tickets reporting issues with taxi-cab services, product reviews, reviews of restaurants and movies, discussions by special interest groups and political surveys. In general they are common in domains where words have different sentiment from what they would have elsewhere.

These data sets present significant challenges for word embedding learning algorithms. First, words in data on specific topics have a different distribution than words from generic corpora. Hence using generic word embeddings obtained from algorithms trained on a corpus such as Wikipedia, would introduce considerable errors in performance metrics on specific downstream tasks such as sentiment classification. For example, in SUDs, discussions are focused on topics related to recovery and addiction; the sentiment behind the word 'party' may be very different in a dating context than in a substance abuse context. Thus domain specific vocabularies and word semantics may be a problem for pre-trained sentiment classification models (Blitzer et al., 2007). Second, there is insufficient data to completely retrain a new set of word embeddings. The SUD data set consists of a few hundred people and only a fraction of these are active (Firth et al., 2017), (Naslund et al., 2015). This results in a small data set of text messages available for analysis. Furthermore, these messages are unstructured and the language used is informal. Fine-tuning the generic

Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP, pages 51–59
Melbourne, Australia July 19, 2018.

word embedding also leads to noisy outputs due to the highly non-convex training objective and the small amount of the data. Since such data sets are common, a simple and effective method to adapt word embedding approaches is highly valuable. While existing work (e.g (Yin and Schütze, 2016)) combines word embeddings from different algorithms to improve upon intrinsic tasks such as similarities, analogies etc, there does not exist a concrete method to combine multiple embeddings for extrinsic tasks. This paper proposes a method for obtaining high quality domain adapted word embeddings that capture domain specific semantics and are suitable for tasks on the specific domain. Our contributions are as follows.

1. We propose an algorithm to obtain Domain Adapted (DA) embeddings. DA embeddings are obtained by performing three steps. (i) First, generic embeddings are obtained from algorithms such as Glove or word2vec that are trained large corpora (such as wikipedia, common crawl). (ii) Next we learn domain specific (DS) embeddings by applying algorithms such as Latent Semantic Analysis (LSA) on the domain specific corpus. (iii) We then perform Canonical Correlation Analysis (CCA) or kernelized CCA (KCCA) to obtain projected DS and projected generic embeddings. The projected DS and generic embeddings are linearly combined via an optimization formulation to obtain a single DA embedding for each word.

2. We propose two optimization based approaches to combining generic and DS embeddings. In the first method, we minimize the sum of squared distance of the DA embeddings from the projected embeddings. The second approach combines projected embeddings in such a way that the document clusters are tightly packed. This helps in our downstream sentiment analysis task by separating out the clusters.

3. We demonstrate the efficacy of our embeddings by measuring the accuracy of our classifiers built using various embeddings on a sentiment analysis task. In the first set of experiments (Table (1)) we train logistic regression classifiers using a bag-of-words (BOW) framework, for the problem of sentiment analysis. Our experimental results show that the classifier built using DA word embeddings outperform the classifiers built using Glove, word2vec or LSA. Our classifier also outperforms the classifier built using the embeddings output by the concSVD algorithm (Yin and Schütze, 2016) which, obtains a word embedding by performing SVD on a matrix of word embeddings.

4. In the second set of experiments we demonstrate the efficiency of DA embeddings when used to initialize InferSent; a bi-LSTM, encoder/decoder architecture that learns sentence embeddings from input word embeddings. The resulting document embeddings are classified using logistic regression classifier. Performance metrics (see Table (2)) show that DA embeddings outperform generic embeddings such as Glove common crawl, when used to initialize InferSent. Furthermore, we also outperform RNTN, which is a recursive neural network based sentiment analysis algorithm (Socher et al., 2013).

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 briefly introduces the CCA/KCCA and details the procedure used to obtain the DA embeddings. Section 4 describes the experimental set up and discusses the results from sentiment classification tasks on benchmark data sets using standard classification as well as using a sentence encoding algorithm. Section 5 concludes this work.

2 Related Work

This work is related to three areas of research which are outlined below.

CCA based word embeddings and applications in multilingual correlation: In (Dhillon et al., 2012), the authors proposed an algorithm called Two Step CCA to learn word embeddings from a one hot encoding representation of words in a given vocabulary. CCA has been used to learn multilingual word embeddings (Faruqui and Dyer, 2014) from words aligned in text across different languages. Building on this work, (Lu et al., 2015) developed deep CCA to learn multilingual word embeddings using neural networks. In both these algorithms, word embeddings are learned for words and their translations across multiple languages such as English-German or English-

French, separately via a LSA based approach. Embeddings in the two different languages are then projected onto the best k correlated dimensions via CCA.

Recently, (Gouws et al., 2015) proposed a neural network based model that learns across multiple languages without the need for word alignment. This algorithm jointly optimizes learning of monolingual embeddings via an objective similar to (Mikolov et al., 2013), along with a cross lingual alignment task. Recently, CCA has been applied to perform cross-lingual entity linking tasks (Tsai and Roth, 2016).

Most applications of CCA in NLP, as stated above, have focused on multilingual settings. In contrast, in this paper we use CCA/KCCA to improve performance of monolingual word embeddings across data sets in different application domains/contexts for the purpose of a given downstream task such as sentiment classification.

Domain Adaptation with CCA: The idea of using word embeddings across different domains has been explored by (Luo et al., 2014) where word embeddings are learned independently from two large corpora and then combined via a neural network. This is different from our approach where the CCA-based approach is used to exploit cooccurrences and context information in the domain specific data set along with linear properties of the generic word embedding. More recently, (Yin and Schütze, 2016) propose an ensemble approach of combining word embeddings learned via different embedding algorithms across different data sets. One of their proposed approaches is to concatenate word vectors from multiple embeddings and to then performing SVD on the resulting matrix. The resulting embeddings are then evaluated on several intrinsic tasks such as word similarities and analogies. In contrast, our work focuses on adapting the word vectors to incorporate domain specific knowledge which is important for the extrinsic objective of sentiment classification. In Section (4) we compare our algorithms against the algorithms of (Yin and Schütze, 2016) and show that we outperform (Yin and Schütze, 2016) for the task of sentiment analysis. Some other work (Blitzer et al., 2011), (Anoop et al., 2015) and (Mehrkanoon and Suykens, 2017) explores CCA based dimensionality reduction techniques for domain adaptation in problems with multi-modal data in general, but not necessarily natural language data.

Transfer Learning using Sentence Embeddings: The idea of training on a large corpus and testing on a different yet related data set has been successfully explored via transfer learning in computer vision applications (Taigman et al., 2014; Sharif Razavian et al., 2014; Antol et al., 2015). A similar idea has been explored to solve problems in NLP applications via sentence level embeddings with and without composition of word embeddings. An unsupervised algorithm such as skip-thought (Kiros et al., 2015) that adapts the word level skip-gram model by (Mikolov et al., 2013) to sentence level embeddings has demonstrated success in transfer learning tasks. Similarly, (Hill et al., 2016) compare task-specific sentence embeddings to supervised methods for applications on machine translation data.

However, these supervised models fail to perform as well as an unsupervised Skip-Net. The current state-of-the art in sentence embedding algorithms is InferSent (Conneau et al., 2017), which learns a sentence embedding via an encoder trained on the Stanford Natural Language Inference data set. It has demonstrated success in many transfer tasks. While domain adaptation is not the focus of these algorithms, the underlying idea of training a model on one data set/task and testing on a different data set/task is relevant to the theme of this paper. In fact, our experiments demonstrate that our domain adapted embeddings combined with the InferSent architecture can significantly improve over generic embeddings combined with InferSent in the sentiment classification task.

3 Domain Adapted Word Embeddings

Training word embedding algorithms on small data sets leads to noisy outputs, due to lack of data, while embeddings from generic corpora fail to capture specific local meanings within the domain. For example, the word “alcohol” has a somewhat neutral to a mildly positive tone in the common crawl corpus, whereas this same word has a strong negative sentiment in a substance use disorder (SUD) dataset. In order to learn useful word embeddings that incorporate the sentimentality of a small target corpus, we propose learning domain specific embeddings obtained by applying word embedding algorithms on the given target corpus and generic embeddings, obtained using applying word embedding algorithms on large generic corpora, using CCA or kernel CCA (KCCA).

Let $\mathbf{W}_{DS} \in \mathbb{R}^{|V_{DS}|\times d_1}$ be the matrix whose columns are the domain specific word embeddings (obtained by, e.g., the LSA algorithm on the domain specific data set), where V_{DS} is its vocabulary and d_1 is the dimension of the embeddings. Similarly, let $\mathbf{W}_G \in \mathbb{R}^{|V_G|\times d_2}$ be the matrix of generic word embeddings (obtained by, e.g., GloVe algorithm on the Common Crawl data), where V_G is the vocabulary and d_2 is the dimension of the embeddings. Let $V_\cap = V_{DS} \cap V_G$. Let $\mathbf{w}_{i,DS}$ be the domain specific embedding of the word $i \in V_\cap$, and $\mathbf{w}_{i,G}$ be its generic embedding. For one dimensional CCA, let ϕ_{DS} and ϕ_G be the projection directions of $\mathbf{w}_{i,DS}$ and $\mathbf{w}_{i,G}$ respectively. Then the projected values are,

$$\begin{aligned} \bar{w}_{i,DS} &= \mathbf{w}_{i,DS}\,\phi_{DS} \\ \bar{w}_{i,G} &= \mathbf{w}_{i,G}\,\phi_G. \end{aligned} \tag{1}$$

CCA maximizes the correlation ρ between $\bar{w}_{i,DS}$ and $\bar{w}_{i,G}$ to obtain ϕ_{DS} and ϕ_G such that

$$\rho(\phi_{DS}, \phi_G) = \max_{\phi_{DS},\phi_G} \frac{\mathbb{E}[\bar{w}_{i,DS}\bar{w}_{i,G}]}{\sqrt{\mathbb{E}[\bar{w}_{i,DS}^2]\mathbb{E}[\bar{w}_{i,G}^2]}} \tag{2}$$

where the expectation is over all words $i \in V_\cap$.

The d-dimensional CCA with $d > 1$ can be defined recursively. Suppose the first $d-1$ pairs of canonical variables are defined. Then the d^{th} pair is defined by seeking vectors maximizing the same correlation function subject to the constraint that they be uncorrelated with the first $d-1$ pairs. Equivalently, matrices of projection vectors $\mathbf{\Phi}_{DS} \in \mathbb{R}^{d_1\times d}$ and $\mathbf{\Phi}_G \in \mathbb{R}^{d_2\times d}$ are obtained for all vectors in $\mathbf{W}_{DS}$ and $\mathbf{W}_G$ where $d \leq \min\{d_1, d_2\}$. Embeddings obtained by $\bar{\mathbf{w}}_{i,DS} = \mathbf{w}_{i,DS}\,\mathbf{\Phi}_{DS}$ and $\bar{\mathbf{w}}_{i,G} = \mathbf{w}_{i,G}\,\mathbf{\Phi}_G$ are projections along the directions of maximum correlation. The final domain adapted embedding for word i is given by $\hat{\mathbf{w}}_{i,DA} = \alpha\bar{\mathbf{w}}_{i,DS} + \beta\bar{\mathbf{w}}_{i,G}$. We next propose optimization based algorithms to determine α, β.

3.1 α, β that minimizes the sum of squared distances

One way to determine α and β is to find DA embeddings such that in the CCA transformed space, the new DA embeddings are as close as possible to both generic and DS embeddings. This is expressed by the following optimization problem,

$$\min_{\alpha,\beta} \|\bar{\mathbf{w}}_{i,DS} - (\alpha\bar{\mathbf{w}}_{i,DS} + \beta\bar{\mathbf{w}}_{i,G})\|_2^2 + \|\bar{\mathbf{w}}_{i,G} - (\alpha\bar{\mathbf{w}}_{i,DS} + \beta\bar{\mathbf{w}}_{i,G})\|_2^2. \tag{3}$$

Solving (3) gives $\alpha = \beta = \frac{1}{2}$, i.e., the new vector is equal to the average of the two projections:

$$\hat{\mathbf{w}}_{i,DA} = \frac{1}{2}\bar{\mathbf{w}}_{i,DS} + \frac{1}{2}\bar{\mathbf{w}}_{i,G}. \tag{4}$$

3.2 α, β to minimize the sum of cluster variance

A major goal of learning domain adapted embeddings is to use them in a downstream task such as sentiment analysis. To facilitate better sentiment analysis it helps if the cluster of positive and negative documents are tightly clustered. That is, we would like to minimize the sum of variance of each individual cluster of documents. This can be cast as a convex optimization problem that has a closed form solution as shown in the following theorem.

Theorem 1. *Let $\beta = 1 - \alpha$. Then, the optimal value of α that minimizes the sum of the variance of document clusters is given by the following set of equations, $\tilde{\alpha} = \frac{\frac{1}{k}\sum_{i=1}^{k}(d_{gp_i} - \hat{\mu}_p)^\top(\bar{\mu}_p - \bar{d}_{p_i}) + \frac{1}{N-k}\sum_{i=1}^{N-k}(d_{gn_i} - \hat{\mu}_n)^\top(\bar{\mu}_n - \bar{d}_{n_i})}{\frac{1}{k}\sum_{i=1}^{k}(\bar{\mu}_p - \bar{d}_{p_i})^\top(\bar{\mu}_p - \bar{d}_{p_i}) + \frac{1}{N-k}\sum_{i=1}^{N-k}(\bar{\mu}_n - \bar{d}_{n_i})^\top(\bar{\mu}_n - \bar{d}_{n_i})}$ $\alpha = \max(0, \min(\tilde{\alpha}, 1))$.*

Proof. Assume that a DA embedding is expressed as, $\hat{\mathbf{w}}_{i,DA} = \alpha\bar{\mathbf{w}}_{i,DS} + (1-\alpha)\bar{\mathbf{w}}_{i,G}$. Further, let each i^{th} document be expressed as the sum of constituent word embeddings,

$$\begin{aligned} d_i &= \sum_{j=1}^{n} \hat{\mathbf{w}}_{j,DA} \\ &= \sum_{j=1}^{n} (\bar{\mathbf{w}}_{j,G} + \alpha(\bar{\mathbf{w}}_{j,DS} - \bar{\mathbf{w}}_{j,G})) \\ &= d_{g_i} + \alpha\bar{d}_i. \end{aligned}$$

Suppose, there are N documents of which k are positive and $N-k$ are negative. Also, let μ_p, μ_n denote the cluster center of all positive and negative documents respectively. We can determine α that minimizes the sum of cluster variances by solving

$$\min_{\alpha\in[0,1]} \frac{1}{k}\sum_{i=1}^{k} ||d_{p_i} - \mu_p||_2^2 + \frac{1}{N-k}\sum_{i=1}^{N-k} ||d_{n_i} - \mu_n||_2^2 \tag{5}$$

Here μ_p and μ_n are centers of positive and negative document cluster centers. Taking means of clusters, we get $\mu_p = \frac{1}{k}\sum_{i=1}^{k}(d_{g_{p_i}} + \alpha\bar{d}_{p_i}) = \hat{\mu}_p + \alpha\bar{\mu}_p$. Similarly $\mu_n = \hat{\mu}_n + \alpha\bar{\mu}_n$.

$$\min_{\alpha\in[0,1]} \frac{1}{k}\sum_{i=1}^{k} ||(d_{g_{p_i}} - \hat{\mu}_p) - \alpha(\bar{\mu}_p - \bar{d}_{p_i})||_2^2 + \frac{1}{N-k}\sum_{i=1}^{N-k} ||(d_{g_{n_i}} - \hat{\mu}_n) - \alpha(\bar{\mu}_n - \bar{d}_{n_i})||_2^2. \quad (6)$$

The above problem is a very simple convex minimization problem. Differentiating w.r.t. α and setting it to 0, and projecting the resulting solution onto the interval $[0,1]$ we get the desired result. □

3.3 Kernel CCA

Because of its linear structure, the CCA in (2) may not always capture the best relationships between the two matrices. To account for nonlinearities, a kernel function, which implicitly maps the data into a high dimensional feature space, can be applied. For example, given a vector $\mathbf{w} \in \mathbb{R}^d$, a kernel function K is written in the form of a feature map φ defined by $\varphi : \mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_d) \mapsto \varphi(\mathbf{w}) = (\varphi_1(\mathbf{w}), \ldots, \varphi_m(\mathbf{w}))(d < m)$ such that given $\mathbf{w}_a$ and $\mathbf{w}_b$

$$K(\mathbf{w}_a, \mathbf{w}_b) = \langle\varphi(\mathbf{w}_a), \varphi(\mathbf{w}_b)\rangle.$$

In kernel CCA, data is first projected onto a high dimensional feature space before performing CCA. In this work the kernel function used is a Gaussian kernel, i.e.,

$$K(\mathbf{w}_a, \mathbf{w}_b) = \exp\Big(-\frac{||\,\mathbf{w}_a - \mathbf{w}_b\,||^2}{2\sigma^2}\Big).$$

The implementation of kernel CCA follows the standard algorithm described in several texts such as (Hardoon et al., 2004); see reference for details.

Data Set		Embedding	Avg Precision	Avg F-score	Avg AUC
Yelp	$\mathbf{W}_{DA}$	KCCA(Glv, LSA)	85.36± 2.8	81.89±2.8	82.57±1.3
		CCA(Glv, LSA)	83.69± 4.7	79.48±2.4	80.33±2.9
		KCCA(w2v, LSA)	87.45± 1.2	83.36±1.2	84.10±0.9
		CCA(w2v, LSA)	84.52± 2.3	80.02±2.6	81.04±2.1
		KCCA(GlvCC, LSA)	**88.11± 3.0**	**85.35±2.7**	**85.80±2.4**
		CCA(GlvCC, LSA)	83.69± 3.5	78.99±4.2	80.03±3.7
		KCCA(w2v, DSw2v)	78.09± 1.7	76.04±1.7	76.66±1.5
		CCA(w2v, DSw2v)	86.22± 3.5	84.35±2.4	84.65±2.2
		concSVD(Glv, LSA)	80.14± 2.6	78.50±3.0	78.92±2.7
		concSVD(w2v, LSA)	85.11± 2.3	83.51±2.2	83.80±2.0
		concSVD(GlvCC, LSA)	84.20± 3.7	80.39±3.7	80.83±3.9
	$\mathbf{W}_G$	GloVe	77.13± 4.2	72.32±7.9	74.17±5.0
		GloVe-CC	82.10± 3.5	76.74±3.4	78.17±2.7
		word2vec	82.80± 3.5	78.28±3.5	79.35±3.1
	$\mathbf{W}_{DS}$	LSA	75.36± 5.4	71.17±4.3	72.57±4.3
		word2vec	73.08± 2.2	70.97±2.4	71.76±2.1
Amazon	$\mathbf{W}_{DA}$	KCCA(Glv, LSA)	86.30±1.9	83.00±2.9	83.39±3.2
		CCA(Glv, LSA)	84.68±2.4	82.27±2.2	82.78±1.7
		KCCA(w2v, LSA)	87.09±1.8	82.63±2.6	83.50±2.0
		CCA(w2v, LSA)	84.80±1.5	81.42±1.9	82.12±1.3
		KCCA(GlvCC, LSA)	**89.73±2.4**	85.47±2.4	85.56±2.6
		CCA(GlvCC, LSA)	85.67±2.3	83.83±2.3	84.21±2.1
		KCCA(w2v, DSw2v)	85.68±3.2	81.23±3.2	82.20±2.9
		CCA(w2v, DSw2v)	83.50±3.4	81.31±4.0	81.86±3.7
		concSVD(Glv, LSA)	82.36±2.0	81.30±3.5	81.51±2.5
		concSVD(w2v, LSA)	87.28±2.9	**86.17±2.5**	**86.42±2.0**
		concSVD(GlvCC, LSA)	84.93±1.6	77.81±2.3	79.52±1.7
	$\mathbf{W}_G$	GloVe	81.58±2.5	77.62±2.7	78.72±2.7
		GloVe-CC	79.91±2.7	81.63±2.8	81.46±2.6
		word2vec	84.55±1.9	80.52±2.5	81.45±2.0
	$\mathbf{W}_{DS}$	LSA	82.65±4.4	73.92±3.8	76.40±3.2
		word2vec	74.20±5.8	72.49±5.0	73.11±4.8
IMDB	DA	KCCA(Glv, LSA)	73.84±1.3	73.07±3.6	73.17±2.4
		CCA(Glv, LSA)	73.35±2.0	73.00±3.2	73.06±2.0
		KCCA(w2v, LSA)	**82.36±4.4**	**78.95±2.7**	**79.66±2.6**
		CCA(w2v, LSA)	80.66±4.5	75.95±4.5	77.23±3.8
		KCCA(GlvCC, LSA)	54.50±2.5	54.42±2.9	53.91±2.0
		CCA(GlvCC, LSA)	54.08±2.0	53.03±3.5	54.90±2.1
		KCCA(w2v, DSw2v)	60.65±3.5	58.95±3.2	58.95±3.7
		CCA(w2v, DSw2v)	58.47±2.7	57.62±3.0	58.03±3.9
		concSVD(Glv, LSA)	73.25±3.7	74.55±3.2	73.02±4.7
		concSVD(w2v, LSA)	53.87±2.2	51.77±5.8	53.54±1.9
		concSVD(GlvCC, LSA)	78.28±3.2	77.67±3.7	74.55±2.9
	$\mathbf{W}_G$	GloVe	64.44±2.6	65.18±3.5	64.62±2.6
		GloVe-CC	50.53±1.8	62.39±3.5	49.96±2.3
		word2vec	78.92±3.7	74.88±3.1	75.60±2.4
	$\mathbf{W}_{DS}$	LSA	67.92±1.7	69.79±5.3	69.71±3.8
		word2vec	56.87±3.6	56.04±3.1	59.53±8.9
A-CHESS	DA	**KCCA(Glv, LSA)**	32.07±1.3	39.32±2.5	**65.96±1.3**
		CCA(Glv, LSA)	32.70±1.5	35.48±4.2	62.15±2.9
		KCCA(w2v, LSA)	33.45±1.3	**39.81±1.0**	65.92±0.6
		CCA(w2v, LSA)	33.06±3.2	34.02±1.1	60.91±0.9
		KCCA(GlvCC, LSA)	36.38±1.2	34.71±4.8	61.36±2.6
		CCA(GlvCC, LSA)	32.11±2.9	36.85±4.4	62.99±3.1
		KCCA(w2v, DSw2v)	25.59±1.2	28.27±3.1	57.25±1.7
		CCA(w2v, DSw2v)	24.88±1.4	29.17±3.1	57.76±2.0
		concSVD(Glv, LSA)	27.27±2.9	34.45±3.0	61.59±2.3
		concSVD(w2v, LSA)	29.84±2.3	36.32±3.3	62.94±1.1
		concSVD(GlvCC, LSA)	28.09±1.9	35.06±1.4	62.13±2.6
	$\mathbf{W}_G$	GloVe	30.82±2.0	33.67±3.4	60.80±2.3
		GloVe-CC	**38.13±0.8**	27.45±3.1	57.49±1.2
		word2vec	32.67±2.9	31.72±1.6	59.64±0.5
	$\mathbf{W}_{DS}$	LSA	27.42±1.6	34.38±2.3	61.56±1.9
		word2vec	24.48±0.8	27.97±3.7	57.08±2.5

Table 1: This table shows results from the classification task using sentence embeddings obtained from weighted averaging of word embeddings. Metrics reported are average Precision, F-score and AUC and the corresponding standard deviations. Best performing embeddings and corresponding metrics are highlighted in boldface.

4 Experimental Evaluation

In this section we evaluate DA embeddings in binary sentiment classification tasks on four standard data sets. Document embeddings are obtained via i) a standard bag-of-words framework, in which documents are expressed as the weighted combination of their constituent word embeddings and ii) by initializing a state-of-the-art-sentence encoding algorithm, InferSent (Conneau et al.,

2017) with DA word embeddings to obtain sentence embeddings. Encoded sentences are then classified using a logistic regressor. Performance metrics reported are average precision, F-score and AUC. All hyperparameters are tuned via 10 fold cross validation.

4.1 Data Sets

Experiments are conducted using four data sets which differ in vocabulary and content. All four arise in specific domains and hence illustrate the objective of this work. The four data sets are:

- The **Yelp data set** consists of 1000 restaurant reviews obtained from Yelp. Each review is associated with a 'positive' or 'negative' label. There are a total of 2049 distinct word tokens in this data set.

- The **Amazon data set** consists of 1000 product reviews with 'positive' or 'negative' labels obtained from Amazon. It has 1865 distinct tokens.

- The **IMDB data set** consists of 1000 movie reviews with binary 'positive' and 'negative' labels obtained from IMDB. It has 3075 distinct tokens.

- The **A-CHESS data set** is a proprietary data set[1] obtained from a study involving users with alcohol addiction. Text data is obtained from a discussion forum in the A-CHESS mobile app (Quanbeck et al., 2014). There are a total of 2500 text messages, with 8% of the messages indicative of relapse risk. Since this data set is part of a clinical trial, an exact text message cannot be provided as an example. However, the following messages illustrate typical messages in this data set, *"I've been clean for about 7 months but even now I still feel like maybe I won't make it."* Such a message is marked as 'threat' by a human moderator. On the other hand there are other benign messages that are marked 'not threat' such as *"30 days sober and counting, I feel like I am getting my life back."* The aim is to eventually automate this process since human moderation involves considerable effort and time. This is an unbalanced data set (8% of the messages are marked 'threat') with a total of 3400 distinct work tokens.

The first three data sets are obtained from (Kotzias et al., 2015).

4.2 Word embeddings, baselines and parameter settings

The following generic and DS word embeddings are used,

- **Generic word embeddings:** Generic word embeddings used are GloVe[2] from both Wikipedia and common crawl and the word2vec (Skip-gram) embeddings[3]. These generic embeddings will be denoted as Glv, GlvCC and w2v.

- **DS word embeddings:** DS embeddings are obtained via Latent Semantic Analysis (LSA) and via retraining word2vec on the test data sets using the implementation in gensim[4]. DS embeddings via LSA are denoted by LSA and DS embeddings via word2vec are denoted by DSw2v. We also retrained GloVe on our test datasets to obtain domain specific word embeddings. Since, the performance of retrained Glove embeddings was similar to word2vec we shall not present the results of Glove based DS embeddings in this paper.

- **concatenation-SVD (concSVD) baseline:** Generic and DS embeddings are concatenated to form a single embeddings matrix. SVD is performed on this matrix and the resulting singular vectors are projected onto the d largest singular values to form word embeddings. The resultant word embeddings called meta-embeddings proposed by (Yin and Schütze, 2016) have demonstrated considerable success in intrinsic tasks such as similarities, analogies etc.

Dimensions of generic, DS and DA word embeddings are provided in the supplement.
Kernel parameter estimation A rule-of-thumb for estimating the kernel parameter σ is to set σ equal to the median of pairwise distance between data points (Flaxman et al., 2016). We use this rule

[1] Center for Health Enhancement System Services at UW-Madison

[2] https://nlp.stanford.edu/projects/glove/

[3] https://code.google.com/archive/p/word2vec/

[4] https://radimrehurek.com/gensim/

to set the value of σ for all of our experiments that use kernel CCA. CCA is performed using the readily available installation in python. KCCA used in this work is implemented in python and follows closely the implementation developed by (Bilenko and Gallant, 2016).

Data Set	Embedding	Avg Precision	Avg F-score	Avg AUC
Yelp	GlvCC	86.47±1.9	83.51±2.6	83.83±2.2
	KCCA(GlvCC, LSA)	**91.06±0.8**	**88.66±2.4**	**88.76±2.4**
	CCA(GlvCC, LSA)	86.26±1.4	82.61±1.1	83.99±0.8
	concSVD(GlvCC,LSA)	85.53±2.1	84.90±1.7	84.96±1.5
	RNTN	83.11±1.1	-	-
Amazon	GlvCC	87.93±2.7	82.41±3.3	83.24±2.8
	KCCA(GlvCC, LSA)	**90.56±2.1**	**86.52±2.0**	**86.74±1.9**
	CCA(GlvCC, LSA)	87.12±2.6	83.18±2.2	83.78±2.1
	concSVD(GlvCC, LSA)	85.73±1.9	85.19±2.4	85.17±2.6
	RNTN	82.84±0.6	-	-
IMDB	GlvCC	54.02±3.2	53.03±5.2	53.01±2.0
	KCCA(GlvCC, LSA)	59.76±7.3	**53.26±6.1**	56.46±3.4
	CCA(GlvCC, LSA)	53.62±1.6	50.62±5.1	**58.75±3.7**
	concSVD(GlvCC, LSA)	52.75±2.3	53.05±6.0	53.54±2.5
	RNTN	**80.88±0.7**	-	-
A-CHESS	GlvCC	52.21±5.1	**55.26±5.6**	**74.28±3.6**
	KCCA(GlvCC, LSA)	**55.37±5.5**	50.67±5.0	69.89±3.1
	CCA(GlvCC, LSA)	54.34±3.6	48.76±2.9	68.78±2.4
	concSVD(GlvCC, LSA)	40.41±4.2	44.75±5.2	68.13±3.8
	RNTN	-	-	-

Table 2: This table shows results obtained by initializing InferSent encoder with different embeddings in the sentiment classification task. Metrics reported are average Precision, F-score and AUC along with the corresponding standard deviations. Best performing embeddings and corresponding metrics are highlighted in boldface We use $\alpha = 0.5$ for all of our experiments here.

4.3 Results from standard classification tasks

Table 1 presents results from the standard classification task. In this approach, we use a bag-of-words approach to combine word embeddings weighted by their term frequency counts. The resulting encoding $v = \gamma^\top \mathbf{W}$. Here $\gamma \in \mathbb{R}^{|V|}$ represents the weights for all the words in the sentence/document, and $\mathbf{W}$ is the matrix whose columns are word embeddings. A logistic regression classifier is then trained on the training data and used to predict the sentiment labels on the test data sets. From this table, it can be inferred that DA embeddings obtained by applying KCCA on GlvCC generic and LSA DS embeddings provide the best performing results on all data sets. Note that in these experiments $\alpha = \frac{1}{2}$ (3.1). On the Amazon data set, concSVD achieves slightly better average F-score (86.17) and average AUC (86.42) over average F-score (85.47) and average AUC (85.56) obtained by KCCA (GlvCC, LSA). However, KCCA (GlvCC, LSA) achieves an average precision of 89.73 while concSVD achieves an average precision of 87.28. On the A-CHESS data set, owing to the imbalance in the classes, the best performing embedding is one that achieves maximum precision. From the table we can determine that KCCA (GlvCC, LSA) achieves the highest average precision of 36.38.

4.4 Results from InferSent encoding for classification

In this section DA embeddings are used to initialize a state-of-the-art sentence encoding algorithm, InferSent. The resultant sentence embeddings are then classified using a logistic regression classifier. Table 2 presents results from classifying sentences obtained from InferSent. First, the pre-trained encoder[5] initialized with GloVe common crawl embeddings is used to obtain vector representations of the input data. Next, InferSent is fine-tuned with a combination of GloVe common crawl embeddings and DA embeddings. DA embeddings are only obtained for a small subset of a vocabulary, so the combination is obtained by using the common crawl embeddings for the rest of the vocabulary. The same procedure is repeated with concSVD embeddings. Additionally, embeddings are compared against a classic sentiment classification algorithm, the Recursive Neural Tensor Network (RNTN) (Socher et al., 2013). This is a dependency parser based sentiment analysis algorithm. Since the focus of this work is not on sentiment analysis algorithms per se, but on domain adaptation of word embeddings for extrinsic tasks, this is used as a baseline for comparison. From table 2 it can be inferred that KCCA(GlvCC, LSA) embeddings perform better than all other baselines for Yelp, Amazon and A-CHESS data sets. On the IMDB data set, RNTN performs best. This could be a case of (GlvCC, LSA) being bad initial guess embeddings for the IMDB data set. Performance of GlvCC embeddings from table 1 further support this conjecture. Also, InferSent produces superior sentence embeddings than simple averaging hence results from table 2 are better than results in table 1.

4.5 Results from using α that minimizes the sum of cluster variances

As described in Theorem (1), α can be selected to minimizes variance of document clusters when learning DA embeddings. Since from tables 1

[5] https://github.com/facebookresearch/InferSent

and 2 we see that the best performing DA embedding is obtained by KCCA, results for this embedding alone are presented in table 3. Furthermore, empirically we did not observe much difference in CCA DA embeddings obtained using $\alpha = 0.5$ and α that minimizes the sum of cluster variances. From tables 2 and 3 observe that on the Yelp, Amazon and IMDB data sets, there is not much of difference in performance metrics for $\alpha = 0.5$ and the α obtained from Theorem (1). However, on the A-CHESS data set, α as obtained from Theorem (1) does better than $\alpha = 0.5$. This result is not surprising given that the word sentiments on the A-CHESS data set is highly atypical. This supports our hypothesis that using only generic embeddings such as the GloVe common crawl is not sufficient when analyzing datasets such as the A-CHESS dataset.

Data Set	Embedding	α	Avg Precision	Avg F-score	Avg AUC
Yelp	KCCA(Glv, LSA)	0.25	84.75±2.2	80.02±2.5	81.13±2.0
	KCCA(w2v, LSA)	0.45	87.74±2.2	83.57±2.6	84.27±2.4
	KCCA(GlvCC, LSA)	0.6	88.84±2.3	85.36±2.3	85.93±2.0
Amazon	KCCA(Glv, LSA)	0.35	85.63±1.3	84.64±1.9	84.84±1.6
	KCCA(w2v, LSA)	0.54	87.15±2.0	84.27±1.9	84.79±1.6
	KCCA(GlvCC, LSA)	0.4	90.42±2.2	87.48±2.3	87.92±2.0
IMDB	KCCA(Glv, LSA)	0.35	72.10±1.8	72.63±2.3	73.01±2.1
	KCCA(w2v, LSA)	0.4	83.01±1.6	79.10±1.2	79.96±2.0
	KCCA(GlvCC, LSA)	0.45	58.56±1.8	53.29±1.7	60.56±1.9
A-CHESS	KCCA(Glv, LSA)	0.4	37.32±1.6	41.64±2.8	66.13±2.1
	KCCA(w2v, LSA)	0.55	35.06±0.9	43.44±1.4	68.60±1.3
	KCCA(GlvCC, LSA)	0.75	38.65±3.1	43.03±2.2	67.26±2.2

Table 3: This table shows results using KCCA DA embeddings within a BoW framework. Since from tables 1 and 2 we see that the best performing DA embedding is obtained by KCCA, results for this embedding alone are presented in this table. α used minimizes the sum of cluster variances as shown in Theorem (1). Note that on the A-CHESS dataset the value of α is large. This observation supports our hypothesis that on domain specific data sets such as A-CHESS, using only generic embeddings such as the GloVe common crawl, as features for classification or to initialize algorithms such as InferSent is not sufficient.

5 Discussion and Conclusion

In this paper DA embeddings are obtained by optimizing a combination of generic and DS embeddings that are projected along directions of maximum correlation. The resulting DA embeddings are evaluated on sentiment classification tasks from four different data sets. Results show that while actual performance metrics vary from database to database, the optimized DA embeddings outperform both the generic and the DS word embeddings in a standard classification framework; as well as outperform concatenation based combination embeddings. This is a positive results since CCA/KCCA provides a principled formulation for combining multiple embeddings. In contrast, concatenating embeddings followed by SVD is an ad-hoc procedure and does not exploit correlations among multiple embeddings. The need for such DA embeddings is motivated by the limitations of performance of generic embeddings on data sets such as A-CHESS. Initializing InferSent with DA embeddings further improves the output from InferSent. This is encouraging because several NLP tasks such as Sentiment Analysis, POS tagging, etc., use algorithms that must be initialized with word embeddings. Initializing such algorithms with embeddings customized to a particular domain or data set will improve performance of these algorithms. Future work will explore effectiveness of using our approach in other downstream applications such as question/answering, machine translation.

References

KR Anoop, Ramanathan Subramanian, Vassilios Vonikakis, KR Ramakrishnan, and Stefan Winkler. 2015. On the utility of canonical correlation analysis for domain adaptation in multi-view headpose estimation. In *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, pages 4708–4712.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2425–2433.

Natalia Y Bilenko and Jack L Gallant. 2016. Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging. *Frontiers in neuroinformatics* 10.

John Blitzer, Mark Dredze, Fernando Pereira, et al. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*. volume 7, pages 440–447.

John Blitzer, Sham Kakade, and Dean Foster. 2011. Domain adaptation with coupled subspaces. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. pages 173–181.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364* .

Paramveer Dhillon, Jordan Rodu, Dean Foster, and Lyle Ungar. 2012. Two step cca: A new spectral method for estimating vector models of words. *arXiv preprint arXiv:1206.6403* .

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. Association for Computational Linguistics.

Joseph Firth, John Torous, Jennifer Nicholas, Rebekah Carney, Simon Rosenbaum, and Jerome Sarris. 2017. Can smartphone mental health interventions reduce symptoms of anxiety? a meta-analysis of randomized controlled trials. *Journal of Affective Disorders* .

Seth Flaxman, Dino Sejdinovic, John P Cunningham, and Sarah Filippi. 2016. Bayesian learning of kernel embeddings. *arXiv preprint arXiv:1603.02160* .

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. pages 748–756.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation* 16(12):2639–2664.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483* .

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pages 597–606.

Erika B Litvin, Ana M Abrantes, and Richard A Brown. 2013. Computer and mobile technology-based interventions for substance use disorders: An organizing framework. *Addictive behaviors* 38(3):1747–1756.

Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Deep multilingual correlation for improved word embeddings. In *HLT-NAACL*. pages 250–256.

Yong Luo, Jian Tang, Jun Yan, Chao Xu, and Zheng Chen. 2014. Pre-trained multi-view word embedding using two-side neural network. In *AAAI*. pages 1982–1988.

Siamak Mehrkanoon and Johan AK Suykens. 2017. Regularized semipaired kernel cca for domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

John A Naslund, Lisa A Marsch, Gregory J McHugo, and Stephen J Bartels. 2015. Emerging mhealth and ehealth interventions for serious mental illness: a review of the literature. *Journal of mental health* 24(5):321–332.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Andrew Quanbeck, Ming-Yuan Chih, Andrew Isham, Roberta Johnson, and David Gustafson. 2014. Mobile delivery of treatment for alcohol use disorders: A review of the literature. *Alcohol research: current reviews* 36(1):111.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pages 806–813.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 1701–1708.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *HLT-NAACL*. pages 589–598.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1351–1360.