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 aligning corresponding word vectors using Canonical Correlation Analysis (CCA) or the related nonlinear Kernel CCA. Evaluation results on sentiment classification tasks show that the DA embeddings substantially outperform both generic and 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 sets of raw text, have demonstrated remarkable success when used as features to a supervised learner in various applications such as the sentiment classification of text documents. There are, however, many applications with domain specific vocabularies and relatively small amounts of data. The performance of generic word embedding 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 a mobile health intervention treatment that encourages participants to engage in sobriety-related discussions. The goal of such treatments is to analyze content of participant's 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.

Such 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, may 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, content is generated spontaneously on a day to day basis, and language use is informal and unstructured. Fine-tuning the generic word embedding also leads to noisy outputs due to the highly non-convex training objective and the small amount of data. Since

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 37–42
Melbourne, Australia, July 15 - 20, 2018.

such data sets are common, a simple and effective method to adapt word embedding approaches is highly valuable. While existing work (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 to perform domain adaptation or improve on extrinsic tasks.

This paper proposes a method for obtaining high quality word embeddings that capture domain specific semantics and are suitable for tasks on the specific domain. The new Domain Adapted (DA) embeddings are obtained by combining generic embeddings and Domain Specific (DS) embeddings via CCA/KCCA. Generic embeddings are trained on large corpora and do not capture domain specific semantics, while DS embeddings are obtained from the domain specific data set via algorithms such as Latent Semantic Analysis (LSA) or other embedding methods. The two sets of embeddings are combined using a linear CCA (Hotelling, 1936) or a nonlinear kernel CCA (KCCA) (Hardoon et al., 2004). They are projected along the directions of maximum correlation, and a new (DA) embedding is formed by averaging the projections of the generic embeddings and DS embeddings. The DA embeddings are then evaluated in a sentiment classification setting. Empirically, it is shown that the CCA/KCCA combined DA embeddings improve substantially over the generic embeddings, DS embeddings and a concatenation-SVD (concSVD) based baseline.

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

2 Domain Adapted Word Embeddings

Training word embeddings directly on small data sets leads to noisy outputs while embeddings from generic corpora fail to capture specific local meanings within the domain. Here we combine DS and generic embeddings using CCA KCCA, which projects corresponding word vectors along the directions of maximum correlation.

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., running 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., running the 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 between $\bar{w}_{i,DS}$ and $\bar{w}_{i,G}$ to obtain ϕ_{DS} and ϕ_G such that

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

where ρ is the correlation between the projected word embeddings and $\mathbb{E}$ is the expectation 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 $\hat{\mathbf{w}}_{i,DA} = \alpha \bar{\mathbf{w}}_{i,DS} + \beta \bar{\mathbf{w}}_{i,G}$, where the parameters α and β can be obtained by solving the following optimization,

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

Solving (3) gives a weighted combination with $\alpha = \beta = \frac{1}{2}$, i.e., the new vector is equal to the

average of the two projections:

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

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.

3 Experimental Evaluation

This section evaluates DA embeddings in binary sentiment classification tasks on four standard data sets. Document embeddings are obtained via (i) a standard framework, i.e document embeddings are a 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 word embeddings to obtain sentence embeddings. Encoded sentences are then classified using a Logistic Regressor.

3.1 Datasets

The following balanced and imbalanced data sets are used for experimentation,

- **Yelp:** This is a balanced data set consisting of 1000 restaurant reviews obtained from Yelp. Each review is labeled as either 'Positive' or 'Negative'. There are a total of 2049 distinct word tokens in this data set.

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 (STD). Best results are attained by KCCA (GlvCC, LSA) and are highlighted in boldface.

- **Amazon:** In this balanced data set there are 1000 product reviews obtained from Amazon. Each product review is labeled either 'Positive' or 'Negative'. There are a total of 1865 distinct word tokens in this data set.

- **IMDB:** This is a balanced data set consisting of 1000 reviews for movies on IMDB. Each movie review is labeled either 'Positive' or 'Negative'. There are a total of 3075 distinct

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 using sentence embeddings from the InferSent encoder in the sentiment classification task. Metrics reported are average Precision, F-score and AUC along with the corresponding standard deviations (STD). Best results are obtained by KCCA (GlvCC, LSA) and are highlighted in boldface.

word tokens in this data set.

- **A-CHESS:** This 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).

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

3.2 Word embeddings and baselines:

This section briefly describes the various generic and DS embeddings used. We also compare against a basic DA embedding baseline in both the standard framework and while initializing the neural network baseline.

- **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.

- **concatenation-SVD 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 resultant word embeddings. These meta-embeddings proposed by (Yin and Schütze, 2016) have demonstrated considerable success in intrinsic tasks such as similarities, analogies etc.

Details about dimensions of the word embeddings and kernel hyperparameter tuning are found in the supplemental material.

The following neural network baselines are used in this work,

- **InferSent:**This is a bidirectional LSTM based sentence encoder (Conneau et al., 2017) that learns sentence encodings in a supervised fashion on a natural language inference (NLI) data set. The aim is to use the sentence encoder trained on the NLI data set to learn generic sentence encodings for use in transfer learning applications.

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

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

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

- **RNTN:** The Recursive Neural Tensor Network (**?**) baseline is a neural network based dependency parser that performs sentiment analysis. Since the data sets considered in our experiments have binary sentiments we compare against this baseline as well.

Note that InferSent is fine-tuned with a combination of GloVe common crawl embeddings and DA embeddings, and concSVD. The choice of GloVe common crawl embeddings is in keeping with the experimental conditions of the authors of InferSent. Since the data sets at hand do not contain all the tokens required to retrain InferSent, we replace word tokens that are common across our test data sets and InferSent training data with the DA embeddings and concSVD.

Since we have a combination of balanced and unbalanced test data sets, test metrics reported are Precision, F-score and AUC. We perform 10-fold cross validation to determine hyperparameters and so we report averages of the performance metrics along with the standard deviation.

4 Results and Discussion

From Tables 1 and 2 we see that DA embeddings perform better than concSVD as well as the generic and DS word embeddings, when used in a standard classification task as well as when used to initialize a sentence encoding algorithm. As expected, LSA DS embeddings provide better results than word2vec DS embeddings. Note that on the imbalanced A-CHESS data set, on the standard classification task, KCCA embeddings perform better than the other baselines across all three performance metrics. However from Table 2, GlvCC embeddings achieve a higher average F-score and AUC over KCCA embeddings that obtain the highest precision.

While one can argue that when evaluating a classifier, the F-score and AUC are better indicators of performance, it is to be noted that A-CHESS is highly imbalanced and precision is calculated on the minor (positive) class that is of most interest. Also note that, InferSent is retrained on the balanced NLI data set that is much larger in size than the A-CHESS test set. Certainly such a training set has more instances of positive samples. Thus when using generic word embeddings to initialize the sentence encoder, which uses the outputs in the classification task, the overall F-score and AUC are better.

From our hypothesis, KCCA embeddings are expected to perform better than the others because CCA/KCCA provides an intuitively better technique to preserve information from both the generic and DS embeddings. On the other hand the concSVD based embeddings do not exploit information in both the generic and DS embeddings. Furthermore, in their work (Yin and Schütze, 2016) propose to learn an 'ensemble' of meta-embeddings by learning weights to combine different generic word embeddings via a simple neural network. We determine the proper weight for combination of DS and generic embeddings in the CCA/KCCA space using the simple optimization problem given in Equation (3).

Thus, task specific DA embeddings formed by a proper weighted combination of DS and generic word embeddings are expected to do better than the concSVD embeddings and individual generic and/or DS embeddings and this is verified empirically. Also note that the LSA DS embeddings do better than the word2vec DS embeddings. This is expected due to the size of the test sets and the nature of the word2vec algorithm. We expect similar observations when using GloVe DS embeddings owing to the similarities between word2vec and GloVe.

5 Conclusion

This paper presents a simple yet effective method to learn Domain Adapted word embeddings that generally outperform generic and Domain Specific word embeddings in sentiment classification experiments on a variety of standard data sets. CCA/KCCA based DA embeddings generally outperform even a concatenation based methods.

Acknowledgments

We would like to thank Ravi Raju for lending computing support for training our neural network baselines. We also thank the anonymous reviewers for their feedback and suggestions.

References

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.

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* .

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* .

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.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika* 28(3/4):321–377.

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.

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.

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.