The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Cross-Lingual Propagation for Deep Sentiment Analysis

Xin Dong
Rutgers University
New Brunswick, NJ, USA
xd48@rutgers.edu

Gerard de Melo
Rutgers University
New Brunswick, NJ, USA
gdm@demelo.org

Abstract

Across the globe, people are voicing their opinion in social media and various other online fora. Given such data, modern deep learning-based sentiment analysis methods excel at determining the sentiment polarity of what is being said about companies, products, etc. Unfortunately, such deep methods require significant training data, while for many languages, resources and training data are scarce. In this work, we present a cross-lingual propagation algorithm that yields sentiment embedding vectors for numerous languages. We then rely on a dual-channel convolutional neural architecture to incorporate them into the network. This allows us to achieve gains in deep sentiment analysis across a range of languages and domains.

1 Introduction

Motivation. As more and more users come online across the globe, increasing numbers of people are voicing their opinion in social media, blogs, review sites, and other online fora. Given such valuable data, modern deep learning-based sentiment analysis methods excel at determining the sentiment polarity of what is being said about companies, products, etc. (Wang et al. 2015). Unfortunately, such deep methods require substantial amounts of training data, because multiple levels of computation, each with additional weights and parameters, need to be learned, typically via end-to-end training.

This is a significant problem for many of the world's languages, for which resources may be too costly to obtain and training data is scarce, especially when one considers that new training data is needed for each domain and genre. A model trained on movie reviews, for instance, will fare very poorly on the task of assessing digital camera reviews, let alone social media postings such as tweets.

Contributions. In this work, we present a cross-lingual propagation algorithm to overcome these challenges and enable improved deep sentiment analysis across a range of languages and domains. Our approach relies on word vectors that are cross-lingually projected from a source language such as English to any number of target languages. Our key contributions are as follows:

1. We present an approach to project sentiment information across languages.
2. We propose encoding this information in embedding vectors that capture sentiment properties along multiple dimensions and allow the model to adapt to different domains and circumstances. This is different from previous work on cross-lingual projection, which has considered generic sentiment polarity lexicons. Different words, however, may have strikingly different connotations in different contexts. For instance, *hot* is generally positive when referring to music, but tends to be negative when referring to the temperature in a hotel room.
3. We incorporate the induced embeddings into a custom convolutional neural network architecture and show that our approach can lead to consistent gains across different languages on diverse datasets from different domains.

2 Approach

2.1 Sentiment Embedding Induction

For many languages and domains, there is a paucity of available data and resources. In some cases, it may be challenging to obtain sufficient in-domain training data, both because there may be less data available online and because it may be somewhat harder to find annotators. Hence, a question that arises is whether one can assist deep networks by incorporating external cues that enable the model to generalize better. We conjecture that vector representations are a suitable means of injecting sentiment-related signals into neural models, as a sort of external prior. Generic word vectors as produced by word2vec (Mikolov et al. 2013a) are widely used to feed generic semantic information into a model. Preinitialization with such vectors often leads to noticeable gains compared to randomly initialized embedding matrices (Kim 2014).

In our study, we consider the question of whether further gains can be achieved by relying on cross-lingual induction to obtain more targeted signals pertaining to a word's sentiment rather than to its general semantics. For this, we first derive embeddings for English words and then use a graph-based propagation algorithm to project these to further languages. To obtain sentiment embeddings for English words, we consider the following strategies.

Sentiment Lexicons. Despite their inherent limitations, lexicon-driven sentiment analysis methods remain widespread. One of their advantages is that they may be better-suited at performing robustly across different domains

compared to supervised approaches, which may pick up dataset-specific correlations. The latter, for instance, may learn that mentions of the word *novel* in movie reviews often correlate with lower review scores due to movies not living up to the expectations of fans of the novel. We thus consider English sentiment lexicons as a simple baseline form of English vector representations. Specifically, we rely on a recent sentiment lexicon called VADER (Hutto and Gilbert 2014), and view the polarity scores that it assigns to words as components of simple 1-dimensional word vectors.

Domain-Specific Lexicon Induction. Generic sentiment lexicons do not account for the domain-specific nature of word polarity scores. A word that has positive connotations in one domain may have negative connotations in another domain. We hence consider the SocialSent Reddit community-specific data mined by the Stanford NLP group (Hamilton et al. 2016). Their study produced separate domain-specific scores for each of 250 different subcommunities of the Reddit social media forum site. Although this data is biased by its source and by their semi-automatic induction process, we consider it a valuable resource. Taken together, the 250 different lexicons can be used to induce 250-dimensional vector embeddings that reflect the distribution of a word's sentiment polarity across a large range of domains.

Transfer Learning. Finally, for a genuinely data-driven way of obtaining word-specific scores, we rely on a supervised approach based on annotated training data. Given a training collection consisting of n binary sentiment polarity classification tasks (e.g., with documents from n different domains), we learn n corresponding models. From these, we then extract word-level feature weights that are tied to specific prediction outcomes. Specifically, we train n linear models

$$f_i(\mathbf{x}) = \mathbf{w_i^\intercal x} + \mathbf{b_i} \tag{1}$$

for tasks $i = 1, \ldots, n$ using bag-of-words features. Then, each vocabulary word index j is assigned a new word vector $(w_{1,j}, \ldots, w_{n,j})$ that incorporates the linear coefficients for that word across the n different linear models.

Cross-Lingual Induction. Given the initial seed embeddings obtained using one of the aforementioned approaches, we next seek to produce vectors for other languages via cross-lingual projection. This is achieved by propagating weights across words in different languages. We start off with an initial vocabulary $V_0 \subset V$ as a subset, typically with English words, of a general multilingual vocabulary V. Vocabulary items are defined as tuples of languages and normalized surface forms, such that the Spanish word *con* (with) is treated as distinct from the French word *con* (idiot). For each $x \in V_0$, we assume as input a corresponding input vector $\tilde{\mathbf{v}}_x \in \mathbb{R}^n$, obtained using one of the methods introduced earlier, i.e. from a sentiment lexicon, multi-dimensional lexicon induction, or using our transfer learning procedure.

Our goal is to induce embeddings $\mathbf{v}$ for all $x \in V$. We assume a translation lexicon $T_{\mathrm{L}} = \{(x_1, x'_1, w_1), \ldots, (x_m, x'_m, w_m)\}$ that provides evidence of semantic relationships between words in V with weights w_i. Although many entries in T_{L} are expected to be pairings of translational equivalents, T_{L} is not a simple set of one-to-one mappings. Rather, due to polysemy and synonymy, each word may have multiple translations, and we consider such links between arbitrary language pairs. Additionally, T_{L} may also include monolingual links, which we shall use to incorporate connections between synonyms, orthographic variants, and semantically, morphologically, derivationally, or etymologically related words (e.g., "*ensalzar*" and "*ensalzan*" in Spanish).

Given this data, our training objective is to minimize:

$$-\sum_{x \in V} \mathbf{v_x^\intercal} \left[\frac{\mathbf{1}}{\sum\limits_{(\mathbf{x},\mathbf{x'},\mathbf{w}) \in \mathbf{T_L}} \mathbf{w}} \sum_{(\mathbf{x},\mathbf{x'},\mathbf{w}) \in \mathbf{T_L}} \mathbf{w} \mathbf{v_{x'}} \right] + C \sum_{x \in V_0} \|\mathbf{v_x} - \tilde{\mathbf{v}}_\mathbf{x}\|_\mathbf{2} \tag{2}$$

The first part seeks to ensure that sentiment embeddings of words accord with those of their connected words, in terms of the dot product, while the second part ensures that the deviation from the initial word vectors $\tilde{\mathbf{v}}_x$ is minimal (for some very high constant C). Hence, words in the initial vocabulary will receive vectors $\mathbf{v}$ that do not diverge significantly from the original $\tilde{\mathbf{v}}_x$. New words, in contrast, are constrained to have vectors close to those of their neighbors in the graph. For optimization, we preinitialize $\mathbf{v_x} = \tilde{\mathbf{v}}_\mathbf{x}$ for all $x \in V_0$, and then rely on stochastic gradient descent steps. As a result, the sentiment vector signal gradually propagates from words in the original vocabulary to other words in the lexicon. As a side effect, this procedure also increases the coverage of our vectors in the original source language (English).

2.2 Network Architecture

To feed our cross-lingual embeddings into our training, we rely on a custom network architecture, illustrated in Fig. 1. This architecture incorporates an extra channel for the sentiment embeddings. The channel with regular word embeddings enables the model to learn salient patterns and exploit the nearest neighbour and linear substructure properties of standard word embeddings. Our hypothesis is that a separate sentiment channel, with dedicated convolutional filters and pooling, allows for better exploiting the information brought to the table by the sentiment embeddings.

Dual-Channel Inputs and Convolutional Filters. The input of the DC-CNN consists of two sentence matrices $\mathbf{S} \in \mathbb{R}^{s \times d}$ and $\mathbf{S'} \in \mathbb{R}^{s \times d'}$, the rows of which represent the words of the input sentence after tokenization. In the case of $\mathbf{S}$, i.e., the regular channel, each word is represented by its conventional word vector representation. In the case of $\mathbf{S'}$, i.e., the sentiment channel, each word is represented by a sentiment vector embedding. Here, s refers to the length of a sentence, and d and d' represent the dimensionality of the regular and sentiment word vectors, respectively.

As the two sentence matrices are similar to a two channel image, we can perform convolutional operations on both of them via linear filters. Given rows representing discrete words, we rely on several weight matrices $\mathbf{W} \in \mathbb{R}^{h \times d}$ and $\mathbf{W'} \in \mathbb{R}^{h \times d'}$, respectively, for different region sizes

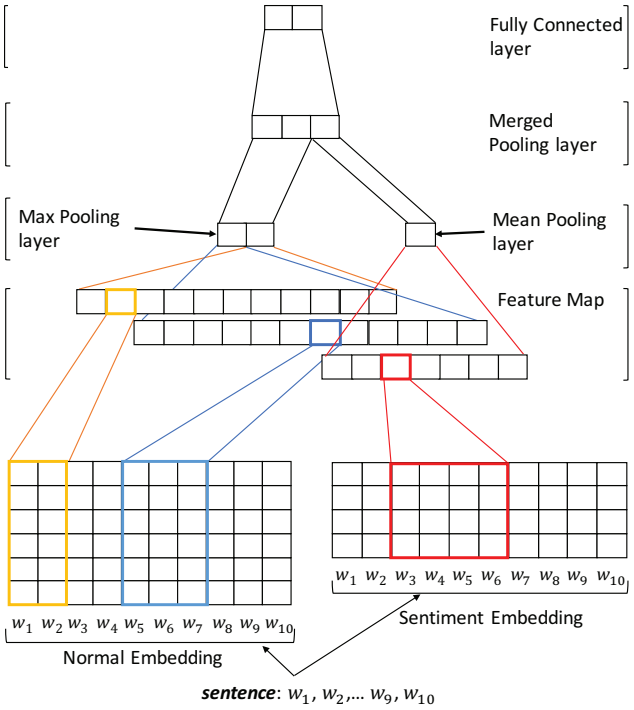

Figure 1: Dual-Channel Convolutional Neural Network architecture.

h. We use the notation $\mathbf{S_{i:j}}$, $\mathbf{S'_{i:j}}$ to denote the respective sub-matrices of $\mathbf{S}$, $\mathbf{S}'$ from row i to row j.

Supposing that these two weights matrices have a filter size of h, for the normal weight matrix, a wide convolution (Kalchbrenner, Grefenstette, and Blunsom 2014) is induced such that out-of-range submatrix values S_i where $i < 1$ or $i > s$ are taken to be zero. Thus, applying the filter on sub-matrices of $\mathbf{S}$ yields the output sequence $\mathbf{o} \in \mathbb{R}^{s+h-1}$ of the convolution operator:

$$o_i = \mathbf{W} \odot \mathbf{S_{i:i+h-1}} \tag{3}$$

where $\odot$ is taken to denote the sum of element-wise multiplication. In contrast, narrow convolutions (Kalchbrenner, Grefenstette, and Blunsom 2014) are used for the sentiment weight matrix, so $\mathbf{S}'$ yields the following output sequence $\mathbf{o}' \in \mathbb{R}^{s-h+1}$ computed as:

$$o'_j = \mathbf{W}' \odot \mathbf{S'_{j:j+h-1}} \tag{4}$$

Wide convolutions ensure that filters can cover words at the margins of the normal weight matrix, whereas the number of sentiment word vectors present in the sentiment embedding is relatively small, so narrow convolutions are sufficiently effective on the sentiment weight matrix.

Next, the c_i and c'_j in feature maps $\mathbf{c} \in \mathbb{R}^{s+h-1}$ and $\mathbf{c}' \in \mathbb{R}^{s-h+1}$ are computed as:

$$c_i = f(o_i + b) \tag{5}$$

$$c'_j = f(o'_j + b) \tag{6}$$

where $i = 1, \ldots, s+h-1$, $j = 1, \ldots, s-h+1$, $b \in \mathbb{R}$ is a bias term, and f is an activation function.

Pooling and Prediction. Subsequently, 1d-max pooling is applied to the $\mathbf{c}$, while 1d-mean pooling is applied to the $\mathbf{c}'$ to extract a scalar from each feature map. The rationale for invoking 1d-mean pooling on the sentiment feature map is to capture the average sentiment polarity instead of the most prominent sentiment features obtained by 1d-max pooling, because the overall polarity of a sentence is not only dependent on individual sentiment values of words. In general, the model shown in Fig. 1 is able to use multiple filters to obtain multiple features in the normal channel, while only using one filter in the sentiment channel to avoid disrupting the normal channel overly.

Finally, the outputs of the pooling functions can be concatenated into a fixed-length vector, which is passed to a fully connected softmax layer to generate the final output probabilities.

Loss Function and Training. Our loss function is the cross-entropy function

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c \in C} y_{i,c} \ln \hat{y}_{i,c}, \tag{7}$$

where n is the number of training examples, C is the set of (two) classes, $y_{i,c}$ are ground truth labels for a given training example and class c, and $\hat{y}_{i,c}$ are corresponding label probabilities predicted by the model, as emitted by the softmax layer. We train our model using Adam optimization (Kingma and Ba 2014) for better robustness across different datasets. More details about our training regime are provided in the Experiments section, which follows next.

3 Experiments

We now turn to our extensive empirical evaluation, which assesses the effectiveness of using cross-lingual projections of three different sources of sentiment word vectors.

3.1 Experimental Setup

Table 1: Dataset Descriptions with abbreviations as follows. MR: movie reviews, FR: Amazon food reviews, HR: hotel reviews, RR: restaurant reviews, TR: Television series reviews.

Language	Source	Domain	train	test
en	SST	MR	6,920	1,821
	AFF	FR	5,945	1,189
es	SE16-T5	RR	2,070	881
nl	SE16-T5	RR	1,317	575
de	TA	RR	1,687	481
ru	TA	HR	2,387	682
it	TA	HR	3,437	982
ja	TA	RR	1,435	411
cs	TA	RR	1722	491
fr	AC	TR	2,737	782

Datasets. For evaluation, we use real-world datasets for several different languages, taken from 5 different sources that cover a range of different domains. These are summarized in Table 1. In our experimental setup, these are all cast as binary

polarity classification tasks, for which we use accuracy as our evaluation metric.

- The Stanford Sentiment Treebank (**SST**) dataset (Socher et al. 2013) consists of movie reviews taken from the Rotten Tomatoes website, including binary labels.
- TripAdvisor (**TA**) is a well-known travel website. To obtain non-English evaluation data, we crawled German, Russian, Italian, Czech, and Japanese reviews of restaurants and hotels from the respective local versions of TripAdvisor. We removed three-star reviews, as these can be regarded as neutral, so reviews with a rating < 3 are considered negative, while those with a rating > 3 were deemed positive.
- The Allocine (**AC**) dataset[1] consists of reviews of French TV series. The data comes annotated with binary labels.
- The SemEval-2016 Task 5 (**SE16-T5**) dataset (Pontiki et al. 2016) provides Spanish and Dutch reviews of restaurants. The task targeted aspect-based sentiment analysis, so we converted the entity-level annotations to sentence-level polarity labels via voting. Since the number of entities per sentence is often one or very low, this process is reasonably precise. In any case, it enables us to compare the ability of different models to learn to recognize pertinent words.
- The Amazon Fine Food Reviews **AFF** (McAuley and Leskovec 2013) dataset consists of 568,454 food reviews left by Amazon users up to October 2012. We extracted a part of it and preprocessed it as for TripAdvisor.

Given the lack of provided test splits for TA, AFF, and AC, we randomly partitioned each into training/validation/test splits with a 80%/10%/20% ratio. Additionally, 10% of the training sets from SE16-T5 were randomly extracted and reserved for validation, while SST provides its own validation set.

For transfer learning, our experiments rely on a multi-domain Amazon product review dataset (Blitzer et al. 2007). This dataset includes reviews for 25 different categories of products and is used to generate our sentiment embeddings using a series of linear models, as explained further below.

Neural Network Details. For CNNs, we make use of the CNN-non-static architecture and hyperparameters proposed in previous work (Kim 2014). The learning rate used to train all languages for it is 0.0006. For our DC-CNN models, the configuration of the regular channel is the same as for CNNs and the remaining hyperparameter values were tuned on the validation sets. An overview of further network parameters resulting from this tuning is given in Table 2.

For greater efficiency and better convergence properties, the training relies on mini-batches of 50 instances. Our implementation considers the maximal sentence length in each mini-batch and zero-pads all other sentences to this length, thus enabling uniform and fast processing of each mini-batch.

Table 2: DC-CNN Model Parameters Setting

(a) General configuration.

Description		Values
Normal Channel	filter region size	(3,4,5)
	feature maps	100
	pooling	1d-max pooling
Sentiment Channel	feature maps	100
	pooling	1d-mean pooling
dropout rate		0.5
optimizer		Adam
activation function		ReLU

(b) Learning rate α and filter region size h used in Sentiment Channel under 9 languages

	en	es	nl	ru	de
α	0.0004	0.001	0.001	0.0004	0.0004
h	5	5	5	5	20
	cs	it	fr	ja	
α	0.001	0.0004	0.0004	0.0004	
h	5	5	5	20	

3.2 Embeddings

The standard pre-trained word vectors used for English are the GloVe (Pennington, Socher, and Manning 2014) ones trained on 840 billion tokens of Common Crawl data[2], while for other languages, we rely on Facebook's fastText Wikipedia embeddings (Bojanowski et al. 2016) as input representations. All of these are 300-dimensional. The vectors are either fed to the CNN, or to the regular channel of the DC-CNN during model initialization, while unknown words are initialized with zeros. All words, including the unknown ones, are fine-tuned during the training process.

In terms of sentiment embeddings, we draw on several forms of external data. For sentiment lexicon embeddings, we rely on VADER (Hutto and Gilbert 2014) to induce one-dimensional vectors. We also rely on the SocialSent (Hamilton et al. 2016) sentiment lexicons to construct 250-dimensional embeddings, as detailed in Section 2.1.

For the transfer learning approach, we train 25 linear SVM models to extract word coefficients for each domain of the multi-domain Amazon review dataset, as well as another model for all domains together, yielding a 26-dimensional sentiment embedding. However, just naïvely using bag-of-word features can lead to counter-intuitive weights. If a word like "*pleased*" in one domain mainly occurs after the word "*not*", while the reviews in another domain primarily used "*pleased*" in its unnegated form, then this word would be assessed as possessing opposite polarities in different domains. To avoid this, we generally treat occurrences of "*not* ⟨word⟩" as a single feature "*not_*⟨word⟩".

For cross-lingual projection, we extract links between words from a 2017 dump of the English edition of Wiktionary, which covers not just English but a broad range of languages. We restrict the vocabulary table T_{L} to include the languages in Table 1, mining corresponding translation, synonymy, morphology, derivation, and etymological links from Wiktionary (de Melo 2014). Since the same pair of words may occur multiple times in this data (for different semantic relationships), we define the weights w_i for a given pair in

[1] https://www.irit.fr/ Tim.Van-De-Cruys/tal/tp/tp3/tp3.pdf

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

T_{L} to be a count of the number of links we have for that pair. V_0 is defined as the set of all English words in T_{L}, using the vectors from the aforementioned English embeddings where available, and assigning all other English words a zero-valued vector, based on the assumption that they are neutral.

Table 3 compares the coverage of different embeddings with respect to our evaluation datasets. Unsurprisingly, generic word embeddings from GloVe (for English) and fastText (for other languages), denoted as G/F, have the largest relative coverage, due to being trained on massive amounts of text. However, our multilingual sentiment embeddings also fare quite well on a number of languages, approaching the coverage of generic word embeddings. The coverage is lower for languages with complex morphology or long compounds as tokens, as is the case for Czech and German. For comparison, we as well list the coverage of the Polyglot lexicon induction method (PG) from (Chen and Skiena 2014).

We have also applied our algorithm using the entirety of Wiktionary as T_{L}, which includes numerous further languages in the long tail. This provides us with sentiment embeddings for over 50 languages (even more if one counts embeddings with smaller vocabulary sizes). For many of these, to the best of our knowledge, no existing sentiment analysis resources exist other than the work by (Chen and Skiena 2014). For this data, please refer to http://gerard.demelo.org/sentiment/.

3.3 Results and Discussion

Comparing Embeddings for CNNs. Our main results are summarized in Table 4. The simplest baseline is to use a CNN model with randomly initialized word vectors. In comparison, CNNs with standard GloVe/fastText embeddings (G/F), where GloVe is used for English, and fastText is used for all other languages, obtain substantial gains across all languages. Thus, word vectors do tend to convey pertinent word semantics signals that enable models to generalize better.

We next consider the benefits of our multilingual sentiment embeddings when applying regular CNNs. For this, we simply concatenate the regular word embeddings with the different forms of sentiment embeddings that we have produced, including those from the sentiment lexicon VADER (V), from SocialSent (SS), and from transfer learning from Amazon reviews (A). As a baseline, we consider the Polyglot sentiment lexicons (PG) from (Chen and Skiena 2014).

The results of using our embeddings as opposed to regular embeddings are somewhat mixed. Using cross-lingual inductions based on VADER (V) and Amazon-based transfer learning (A) leads to small improvements on several languages. However, the results are far from consistent. In several cases, appending sentiment information to the word embeddings results in slightly degraded scores, e.g. for Spanish, although all input information that was previously there continues to be provided to the model. This suggests that a simple concatenation may harm the model's ability to benefit from the semantic relatedness information between words that are provided by regular word vectors. This risk seems to be more pronounced for larger-dimensional sentiment embeddings. However, we also see that our approach of inducing multi-dimensional sentiment embeddings generally outperforms the Polyglot baseline ("PG").

Sentiment Embeddings with Dual-Channel Approach. Next, we consider our DC-CNN architecture with its dual-channel mechanism. In this approach, the sentiment embeddings are provided to the model in a separate channel, with designated convolutional filters and pooling layers. Thus, the model can exploit the two kinds of information independently, and learn a suitable way to aggregate them to produce an overall output classification.

In this case, we observe that incorporating additional sentiment embeddings leads to fairly consistent and occasionally quite pronounced gains over CNNs with just the GloVe/fastText vectors. This demonstrates not only that the sentiment embeddings tend to provide important complementary signals but also that a dual-channel approach is best-suited to incorporate such signals into deep neural models.

We again also find that our approach of inducing multi-dimensional sentiment embeddings outperforms the "PG" baseline of inducing sentiment lexicons (Chen and Skiena 2014). Overall, our data-driven transfer learning approach of learning sentiment polarities on Amazon reviews using a series of linear models (A) tends to be the best choice, with notable gains across a large number of languages.

Analysis. For further analysis, we consider the special setting of relying on transfer learning via the Amazon embeddings, but without allowing the model to adjust them during training (denoted as SA in Table 4). While on a few languages, the results remain similar, on several languages, we notice a significant degradation. Hence, we can conclude that although our sentiment embeddings provide useful sentiment information, it is important to allow the model to adjust them to cater to the domain-specific meanings and corpus-specific correlations for a given evaluation dataset. Our DC-CNN architecture facilitates this domain adaptation process.

Finally, in Table 5, we provide some examples of sentences that were misclassified by the CNN with regular embeddings but correctly classified by our DC-CNN model using G $\|$ A embeddings. We encounter words such as "*ribald*", "*compelling*", but also "*contrived* and "*clink*". It appears that the sentiment embeddings enable the model to recognize the polarity of words that may not have had sufficiently strong polarity associations in the training set.

Influence of Training Set Size. To look into the effect of sentiment embeddings on training sets of different sizes, we use the English and Czech datasets as instructive examples. We split each of their training portions into 5 parts and plot the results for growing training set sizes, evaluated on the full test set. The results of the CNN model (for regular word vectors only as well as for concatenations) and of the DC-CNN model (for a separate channel with sentiment embeddings) are plotted in Fig. 2. We observe that the achieved gains tend to be even more pronounced on smaller training sets. This shows that the sentiment embeddings are particularly useful when domain-specific training data is scarce, although a modest amount of training data is still needed for the model to be able to adapt the sentiment vectors to the target domain.

Table 3: Coverage of different resources. PG: Polyglot; G: GloVe; F: fastText; All: the number of words appearing in the dataset.

Embedding	*en*		*es*	*nl*	*ru*	*de*	*cs*	*it*	*fr*	*ja*
	MR	FR	RR	RR	HR	RR	RR	HR	TR	RR
PG	2,523	1,252	686	584	851	746	533	1,121	1,470	110
Our method	14,793	8,835	4,342	2,445	12,618	3,674	4,421	6,945	6,945	3,732
G/F	16,510	10,225	5,304	3,913	17,267	10,425	11,115	9,007	9,895	5,729
All	17,516	11,362	5,974	4,794	19,881	12,453	14,952	11,109	12,498	11,305

Table 4: Accuracy on 9 language datasets using 12 embedding alternatives, where d denotes the embedding dimensionality, and the embedding types are abbreviated as follows. R: Randomly initialized embedding; +: Embeddings are concatenated; ‖: dual-channel; SS: SocialSent Embedding; V: VADER Embedding; A: Transfer learning using embeddings resulting from supervised training on Amazon reviews; SA: static Amazon Embedding

	Embedding	d	*en*		*es*	*nl*	*ru*	*de*	*cs*	*it*	*fr*	*ja*
			MR	FR	RR	RR	HR	RR	RR	HR	TR	RR
Baselines	R	300	80.78	86.54	81.50	75.30	90.18	88.09	90.00	93.18	87.21	78.59
	G/F	300	85.99	88.73	85.13	77.57	93.84	92.10	92.46	95.92	91.82	76.89
	G/F + PG	301	86.11	88.90	85.02	77.91	93.84	92.10	93.28	95.36	91.43	75.18
Concatenation (Our Embeddings)	G/F + V	301	86.33	88.81	84.45	78.26	**94.28**	92.93	92.87	**96.91**	91.56	75.18
	G/F + SS	550	85.45	88.14	83.31	76.87	91.50	91.48	91.85	94.80	90.41	75.67
	G/F + A	326	86.55	89.23	84.56	78.96	93.40	**93.56**	93.28	96.34	92.33	75.91
Dual Channel Baselines	G/F ‖ R	300/26	85.78	89.07	84.79	78.09	93.40	92.31	93.08	95.78	91.82	76.64
	G/F ‖ PG	300/1	85.72	88.73	85.13	77.39	93.11	91.68	93.08	95.78	91.30	76.64
Our Full Approach	G/F ‖ V	300/1	85.78	88.98	84.45	77.39	93.11	92.31	93.28	95.64	91.82	77.13
	G/F ‖ SS	300/250	86.11	88.73	84.56	77.91	**94.28**	92.10	**93.69**	96.77	91.94	85.40
	G/F ‖ A	300/26	86.60	**89.49**	**85.93**	**79.30**	93.26	92.31	**93.69**	96.48	**92.97**	**88.08**
Analysis	G/F ‖ SA	300/26	**86.82**	88.81	84.45	78.43	93.84	91.89	93.08	95.92	92.07	77.62

Cross-Domain Generalization. Finally, we evaluated the cross-domain generalization abilities of our sentiment embedding approach. For English, we have two different datasets, MR and FR, and hence can evaluate how well a model trained on one dataset performs on another. For the sentiment embeddings, we use the Amazon ones, as these performed best in our previous experiments on in-domain data. Training on MR and evaluating on the test set for FR, we achieve 69.24% when training using GloVe embeddings only, and 74.85% when training using our dual-channel approach with additional Amazon sentiment embeddings. This result provides further corroboration of our hypothesis that a sentiment embedding approach leads to substantially better generalization.

4 Related Work

Cross-Lingual Sentiment Analysis. The majority of research on sentiment analysis has focused on the English language. One way of supporting further languages is to use machine translation, as has been investigated for subjectivity (Banea et al. 2008) and sentiment polarity (Demirtas and Pechenizkiy 2013). However, this may be overly computationally intensive when analyzing the vast quantities of data posted online. Moreover, Duh et al. argued that even perfect machine translation incurs a degradation in the result quality for sentiment analysis (Duh, Fujino, and Nagata 2011), while showing that regular adaptation methods do not work well in this setting. Haas & Versley provided empirical support in line with these claims (Haas and Versley 2015). Another option, proposed by Vilares et al., is to forgo supervision from training data, instead relying on rules applied to syntactic dependencies (Vilares, Gómez-Rodríguez, and Alonso 2017). Wan presented a bilingual co-training approach that jointly trains a system on two languages, considering each language an independent view (Wan 2009).

An alternative strategy is to use cross-lingual projection, which involves transferring annotations from a source language resource to some target language by exploiting translational equivalence (de Melo and Weikum 2010; Gutiérrez et al. 2016) or parallel corpora (de Melo and Weikum 2009). There are several English-language sentiment lexicons, many of which have been compiled manually (Hu and Liu 2004) or via crowdsourcing (Mohammad and Turney 2013). While these are costly to produce, one can subsequently use cross-lingual projection techniques to effectively translate such lexicons to new languages. Mihalcea et al. proposed an approach to achieve this for subjectivity lexicons (Mihalcea, Banea, and Wiebe 2007). Boyd-Graber & Resnik proposed a cross-lingual probabilistic generative model for sentiment analysis (Boyd-Graber and Resnik 2010). Balamurali et al. use cross-lingual projection by means of word sense disambiguation (Balamurali, Joshi, and Bhattacharyya 2012), but the approach hinges on the existence of multilingual wordnets that map words in different languages to a shared interlingual representation. In terms of broad multilingual support, the most relevant previous work is that of Chen & Skiena, which used joint cross-lingual propagation to create sentiment lexicons for dozens of languages (Chen and Skiena 2014). We compare against these in our experiments.

Table 5: Examples of English SST sentences misclassified by CNNs with regular embeddings but correctly classified by DC-CNNs using G || A embeddings. Words covered by our Transfer Learning embeddings after cross-lingual expansion with a non-zero vector are given in italics.

Classification	Sentence
positive	*Though Mama takes a bit too long to find its rhythm and a* third-act *plot development is somewhat melodramatic, its ribald humor and touching nostalgia are sure to please anyone in search of a Jules and Jim for the new millennium .*
positive	*An utterly compelling 'who wrote it' in which the reputation of the most famous author who ever lived comes into question.*
negative	*Feels like one of those contrived,* only-in - *Hollywood productions where name actors deliver big performances created for the sole purpose of generating Oscar talk.*
negative	*This* pep-talk *for faith, hope and charity does little to offend, but if saccharine earnestness were a crime, the film's producers would be in the clink for life.*

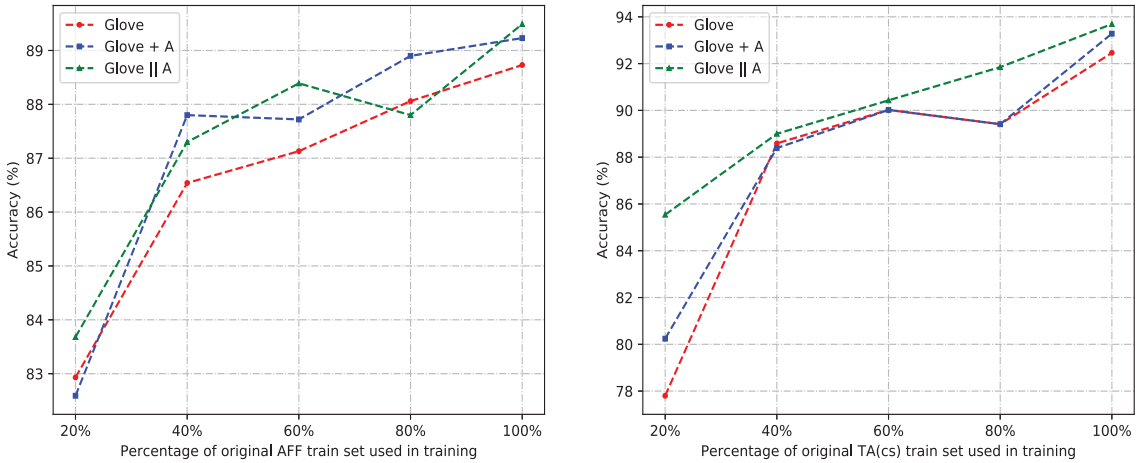

Figure 2: Effectiveness of three embedding alternatives on two languages for varying training set sizes.

An important shortcoming of sentiment lexicons is that they neglect the domain-specific nature of word sentiment polarities. For instance, a word such as *scary* tends to be negative, but may also correlate with positive movie review scores. Our work, in contrast, focuses on multi-dimensional word representations for deep neural networks.

Word Embeddings. Word embedding methods such as word2vec (Mikolov et al. 2013a) are now ubiquitously used across a wide range of tasks in the broad area of text mining and natural language processing, including in models for sentiment analysis (Socher et al. 2013; Kim 2014; dos Santos and Gatti 2014). Cross-lingual distributed representations have been studied as well. These are typically produced either by aligning multiple monolingual word embedding models using techniques such as linear projections (Mikolov et al. 2013b) or CCA (Faruqui and Dyer 2014), by jointly training in multiple languages via parallel corpora (Klementiev, Titov, and Bhattarai 2012; Luong, Pham, and Manning 2015), or by exploiting multilingual semantic resources (de Melo 2015; de Melo 2017). However, the co-occurrence-based training objectives of methods such as word2vec do not consider sentiment specifically. Our work, in contrast, focuses on representations that capture sentiment-specific cues rather than generic word semantics.

Mining Sentiment Information. There are various monolingual methods to mine sentiment information. For instance, one can collect reviews that come with associated ratings, and use supervised learning to learn feature weights (Thelwall et al. 2010). One can also apply distant supervision exploiting the presence of emoticons or hashtags in online social media (Tang et al. 2014). In our work, we as well start off with such approaches to obtain initial English data, and then rely on a cross-lingual induction procedure to transfer the acquired representations to new language.

5 Conclusions

We have investigated the use of cross-lingually induced sentiment representations to boost the effectiveness of deep neural models for sentiment analysis, incorporated into the network via a separate channel. Extensive experiments on 9 different languages confirm the effectiveness of this approach, leading to substantial gains across a series of datasets from heterogeneous domains. Our approach has allowed us to generate sentiment embeddings for over 50 languages. Please refer to http://gerard.demelo.org/sentiment/ to obtain a copy of our data.

Acknowledgments

This research is supported in part by the DARPA SocialSim program.

References

Balamurali, A.; Joshi, A.; and Bhattacharyya, P. 2012. Cross-lingual sentiment analysis for indian languages using linked wordnets. *Proc. COLING 2012* 73–82.

Banea, C.; Mihalcea, R.; Wiebe, J.; and Hassan, S. 2008. Multilingual subjectivity analysis using machine translation. In *Proc. EMNLP 2008*, 127–135.

Blitzer, J.; Dredze, M.; Pereira, F.; et al. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. ACL 2007*.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Boyd-Graber, J., and Resnik, P. 2010. Holistic sentiment analysis across languages: Multilingual supervised latent dirichlet allocation. In *Proc. EMNLP 2010*, 45–55.

Chen, Y., and Skiena, S. 2014. Building sentiment lexicons for all major languages. In *Proc. ACL 2014*, 383–389.

de Melo, G., and Weikum, G. 2009. Extracting sense-disambiguated example sentences from parallel corpora. In *Proc. Workshop on Definition Extraction at RANLP 2009*.

de Melo, G., and Weikum, G. 2010. Towards universal multilingual knowledge bases. In *Proc. GWC 2010*.

de Melo, G. 2014. Etymological Wordnet: Tracing the history of words. In *Proc. LREC 2014*.

de Melo, G. 2015. Wiktionary-based word embeddings. In *Proc. MT Summit XV*.

de Melo, G. 2017. Inducing conceptual embedding spaces from Wikipedia. In *Proc. WWW 2017*.

Demirtas, E., and Pechenizkiy, M. 2013. Cross-lingual polarity detection with machine translation. In *Proc. WISDOM 2013*, 9:1–9:8.

dos Santos, C. N., and Gatti, M. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proc. COLING 2014*.

Duh, K.; Fujino, A.; and Nagata, M. 2011. Is machine translation ripe for cross-lingual sentiment classification? In *Proc. ACL-HLT 2011*, 429–433.

Faruqui, M., and Dyer, C. 2014. Improving vector space word representations using multilingual correlation. In *Proc. EACL 2014*, 462–471.

Gutiérrez, E. D.; Shutova, E.; Lichtenstein, P.; de Melo, G.; and Gilardi, L. 2016. Detecting cross-cultural differences using a multilingual topic model. *TACL* 2016:4.

Haas, M., and Versley, Y. 2015. Subsentential sentiment on a shoestring: A crosslingual analysis of compositional classification. In *Proc. NAACL-HLT 2015*, 694–704.

Hamilton, W. L.; Clark, K.; Leskovec, J.; and Jurafsky, D. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proc. EMNLP 2016*, 595–605.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proc. KDD 2004*, 168–177.

Hutto, C., and Gilbert, E. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proc. ICWSM-14*.

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klementiev, A.; Titov, I.; and Bhattarai, B. 2012. Inducing crosslingual distributed representations of words. In *Proc. COLING 2012*.

Luong, M.-T.; Pham, H.; and Manning, C. 2015. Bilingual word representations with monolingual quality in mind. In *Proc. NAACL Workshop on Vector Space Modeling for NLP*.

McAuley, J. J., and Leskovec, J. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proc. WWW 2013*, 897–908.

Mihalcea, R.; Banea, C.; and Wiebe, J. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proc. ACL 2007*, 976–983.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS 2013*, 3111–3119.

Mohammad, S. M., and Turney, P. D. 2013. Crowdsourcing a word-emotion association lexicon. *Comp. Int.* 29(3):436–5.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proc. EMNLP*.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Mohammad, A.; et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. *Proc. SemEval*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. EMNLP 2013*, 1631–1642.

Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; and Qin, B. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proc. ACL 2014*.

Thelwall, M.; Buckley, K.; Paltoglou, G.; Cai, D.; and Kappas, A. 2010. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.* 61(12):2544–2558.

Vilares, D.; Gómez-Rodríguez, C.; and Alonso, M. A. 2017. Universal, unsupervised (rule-based), uncovered sentiment analysis. *Knowledge-Based Systems* 118:45–55.

Wan, X. 2009. Co-training for cross-lingual sentiment classification. In *Proc. ACL-IJCNLP 2009*, 235–243.

Wang, L.; Liu, K.; Cao, Z.; Zhao, J.; and de Melo, G. 2015. Sentiment-aspect extraction based on Restricted Boltzmann Machines. In *Proc. ACL 2015*.