Lowering the Cost of Improved Cross-Lingual Sentiment Analysis

by

Mohamed Mohamed Saad Atia Abdalla

Submitted in conformity with the requirements
for the Degree of Master of Science.

Department of Computer Science
University of Toronto

April 17, 2018

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Abstract

In this work[1], we take a two pronged approach to improving cross-lingual sentiment analysis. First, using a new technique, we show that word embeddings can be enriched for sentiment information without the need of a labeled corpora; this enrichment improves performance across the board in both sentiment and non-sentiment-related tasks. Second, we study how a linear transformation that can effectively leverage embeddings and English sentiment knowledge without the need for accurate translation. Across a series of diverse tasks, we show that this transformation is effective in analyzing languages with scarce data, despite the sub-par translation accuracy, thus making sentiment and related analyses for many languages inexpensive.

[1]This work is based on two papers: i)(Abdalla and Hirst, 2017) and ii)(Abdalla et al., 2018)

Acknowledgements

This work was financially supported by Vetenskapsrådet (the Swedish Research Council), through the grant "The advantage of country comparisons — Towards a new method for estimating language effects in cross-cultural surveys" (Principal Investigator: Stefan Dahlberg, University of Gothenburg) and by the Natural Sciences and Engineering Research Council of Canada, through the grant "Applied computational models of discourse, argument, and text" (Principal Investigator: Graeme Hirst, University of Toronto).

I am grateful to Professor Graeme Hirst for his supervision and guidance. I am grateful to Professor Frank Rudzicz (University of Toronto), Magnus Sahlgren (RISE-SICS Research Institutes of Sweden, Swedish Insitute of Computer Science), and Amaru Cuba Gyllensten (RISE-SICS) for technical help and detailed discussions throughout my Master's work. To Professor Stefan Dahlberg (University of Gothenberg), and Moustafa Abdalla (University of Oxford) I am grateful for their broad discussion and advice regarding my work.

1 Introduction

Word embeddings (*i.e.,* word vectors, distributed representations) aim to represent words as a sequence of numbers that can subsequently be used as input for a wide variety of statistical machine learning models and techniques for various tasks. The complexity of such encodings varies from the very simple (*e.g.,* one-hot encoding) to the relatively complex (*i.e.,* automatically generated embeddings by newer machine learning techniques).

While word embeddings have worked well in the past for a variety of NLP tasks, because of the distributional hypothesis (Firth, 1957), there remains room for improvement. For example, sentiment words with opposite emotional values are often used in the same context, and thus have very close representations in a language's vector space, closer than their antonymy implies. If we could incorporate sentiment during the creation of our embeddings, we would add some degree of separation between words that occur in the same context but with opposite meanings.

1.1 Incorporating Sentiment Information into Word Embeddings Without The Need for Labeled Corpora

In the first part of this work, we introduce a more effective and generalizable way of incorporating sentiment during the creation of word embeddings, which improves performance over previous approaches (Tang et al., 2016) on a diverse set of tasks and does not require labeling of the entire corpus. We also test the effect of our approach and past sentiment-enriched embeddings on non-sentiment related tasks to see whether the increase in performance for sentiment-related tasks comes at a price of performance in other unrelated tasks.

Most common approaches to incorporating sentiment in embedding creation have the loss function be composed of a context loss and a sentiment loss. Tang et al. (2016) make use of this approach and make use of distant-supervision (Go et al., 2009) by looking at sentence level labeling. Inspired by Tang et al., we adopt a similar approach, combining the loss of context with that of sentiment, extending their model to allow for training without the need for human labeling of sentences. We contribute two novel ideas: i) the generalization of the combined context+sentiment approach by substituting the need for sentence-level sentiment labeling with context-based sentiment scores that are automatically calculated; ii) the removal of Tang et al.'s *htanh* layer, as we show that it degrades performance in most tasks.

We quantitatively evaluate our proposed techniques in comparison with previous ones on a variety of tasks, including both sentiment and non-sentiment tasks.

1.2 Cross Lingual Sentiment Analysis without (Good) Translation

The second part of this work deals with leveraging the wealth of sentiment information available for the English language to aid in the analysis of sentiment where data is more scarce. Since many applications for the analysis of sentiment rely on a treasure-trove of labeled data, they cannot readily be applied to other languages. Therefore, many approaches which deal with other languages often: i) experiment with small datasets that are limited in domain or size of training and testing sets (Lee and Renganathan, 2011; Tan and Zhang, 2008), or ii) attempt to elucidate sentiment lexicons for their respective languages (Mohammad et al., 2016).

A growing number of publications attempt to leverage labeled English data to compensate for the relative lack of training material in the other languages. This is usually done through the use of either bilingual lexicons (Balamurali et al., 2012), machine translation (MT) systems (Salameh et al., 2015; Zhou et al., 2016), or more recently through the usage of bilingual vector space embeddings (Chen et al., 2016).

Unfortunately, in many cases, such data is still expensive to obtain. Many languages do not have good, or sometimes any, MT systems, and the cost of producing word alignments or sentence alignments for training bilingual word embeddings (BWE) (Zou et al., 2013; Bengio and Corrado, 2015) or similar techniques (Jain and Batra, 2015) which are frequently used can be too expensive or not feasible. Here, we introduce a high-performance, low-cost approach to cross-lingual sentiment classification, which aims to serve as a benchmark for more expensive methods, and we demonstrate that limited training data suffices for effective cross-lingual sentiment analysis.

Our approach relies on the simple vector space translation matrix method (Mikolov et al., 2013b), which computes a matrix to convert from the vector space of one language to that of another. We noted that sentiment is highly preserved even in the face of poor translation accuracy. We observed that a sentiment classifier trained only with word vectors from English (hereafter referred to as the target language) performs well on unseen words from other unseen languages (referred to as the source languages) translated into the English vector space through the simple matrix method, even with very poor translation scores.[2]

We quantitatively evaluate our methods by training on English and testing on words from Spanish and Chinese. We experiment with differing amounts of data during training to show the robustness of our observation. We then apply the fine-grained sentiment regressor to the task of review classification as done by Chen et al. (2016), and show that our naïve algorithm achieves results similar to their benchmark but at a lower cost.

[2]This work involves translations in both directions between English and the other languages studied. However, we will consistently take the perspective of the translation matrix introduced in section 3.2 and thus refer to English as the target language and the others as source languages.

2 Previous Work

2.1 Word Embeddings

Word embeddings are a dense vector representation for words of a corpus. They range from low-rank approximations of co-occurrence matrices (e.g. Sahlgren (2005); Bullinaria and Levy (2012); Pennington et al. (2014)) to those created using shallow neural networks (e.g. Mikolov et al. (2013a)). The latter approach has been shown to be connected to the former approach (Hashimoto et al., 2016), and all embeddings are heavily influenced by the distributional hypothesis (Sahlgren, 2008).

The specific algorithm that we improve upon is a model that attempts to predict the current word given the context (surrounding words), termed continuous bag of words (CBOW) (Mikolov et al., 2013a), which contrasts with the common Skip-Gram approach (Mikolov et al., 2013c) that attempts to predict the context words given the current word. We made this decision as using this very common setup allowed us to compare our results with those of previous work.

2.2 Enriched Word Embeddings

Some past work has sought to improve the quality and utility of word embeddings by incorporating external non-context information into the embeddings, a technique we refer to as "embedding enrichment". Previous work has enriched embeddings with different external information ranging from semantic information (Faruqui et al., 2015) to sentiment information (Maas et al., 2011; Socher et al., 2011; Tang et al., 2014, 2016).

Faruqui et al. (2015)'s approach uses semantic lexicons to retrofit word-vectors by encouraging linked words to have similar vector representations. In this work we included the model which performed best on their sentiment analysis, retrofitting the word vectors using the paraphrase database (PPDB) (Ganitkevitch et al., 2013), for comparison in our battery of tests.

Focused on sentiment enrichment alone, Maas et al. (2011) make use of a probabilistic document modeling approach, constraining words that express similar sentiment to have a more similar representation. Socher et al. (2011) make use of manually labeled data and the semantic principle of composition, that the meaning of a complex expression can be determined through understanding of the parts of which it is composed, to learn the meaning and sentiment of phrases and sentences. Tang et al. (2014) make use of distant supervision and a massive Twitter dataset in order to learn word embeddings. They then extend their work by combining the traditional embedding model (CBOW) with their previous work, showing that the incorporated loss function serves to improve the embeddings' capability to analyze sentiment (Tang et al., 2016).

2.3 Cross-Lingual Word Embeddings

When working with more than one language, we seek to satisfy two objectives: i) mono-lingually, similar words of the same language have similar embeddings; and ii) cross-lingually, similar words across languages also have similar embeddings. Satisfying these two criteria would allow us to use algorithms trained for the embeddings of a single language (such as English with a wealth of labeled data) for other languages as well. Below we discuss algorithms to achieve the second cross-lingual objective, their costs, performance, and the rationale underlying our algorithm design.

2.3.1 Offline Alignment

The simplest approach to achieving the cross-lingual objective is to train each monolingual objective separately (create a model for each language), and then learn a transformation to enforce the second objective. This approach uses a dictionary of paired words in order to learn a transformation or 'alignment' from the vector space of one language to that of another.

First introduced by Mikolov et al. (2013b), and later extended by Faruqui and Dyer (2014), offline alignment is fast and low cost, but does not achieve a high translation accuracy. A big drawback of these approaches is that using a dictionary ignores the polysemic nature of languages. It is also not clear or proven that a single transformation would be able to capture the relationship between all the words in a cross-lingual setting.

We opt to use offline alignment to show that such a low-cost approach does, in fact, capture a significant part of the relationship between words of different languages when it comes to sentiment. That is, a single transformation (linear in the case of our work) is enough to learn a projection which allows one to use labeled English data to aid in sentiment analysis.

2.3.2 Parallel-Only

An alternative approach to offline alignment is the parallel-only approach. Approaches which fall into this group, such as bilingual compositional model (BiCVM) (Hermann and Blunsom, 2013) and bilingual auto-encoder (BAE) (Sarath Chandar et al., 2014), rely purely on sentence-aligned parallel data to train a model with similar representations. Such approaches can be effective, but require data that is expensive to collect and is not always available. Another drawback is that these approaches can be affected by the writing style of the parallel text (Bengio and Corrado, 2015).

2.3.3 Jointly-Trained Model

Combining the offline alignment and parallel-only algorithms is a third class of jointly-trained approaches. These approaches jointly optimize the monolingual objective at the same time as the

cross-lingual objective, making use of both monolingual and parallel data. When looking at the cost of parallel data, approaches like those of Klementiev et al. (2012) and Zou et al. (2013) use word-aligned data in order to learn the fine-grained cross-lingual features and tend to be quite slow, while that of Bengio and Corrado (2015) relies on sentence-aligned data and is faster than other jointly-trained models. While these models are cheaper than parallel-only approaches, it is still quite expensive to obtain a good set of word or sentence alignments for many languages, a problem we sought to avoid.

A lower-cost alternative to these expensive jointly trained models was proposed by Duong et al. (2016) and later used to project multiple languages in the same vector space (Duong et al., 2017). The model involved creating and making use of translations produced using a bilingual dictionary during training. Using expectation-maximization-inspired training, sentence translations were produced by selecting translations of words based on context to deal with polysemy, and this approach demonstrated improvements on the simple linear transformation method. However, even this model uses a significantly larger amount of data than the methods used in this work, with its smallest dictionary being composed of 35,000 word pairs compared to our largest model of 9500 words for both translation and sentiment regression.

2.4 Cross-Lingual Sentiment Analysis

Previous approaches to cross-lingual sentiment analysis can be classified into two main categories: either i) rely on parallel corpora to train BWE's (use pre-trained embeddings) (Chen et al., 2016; Sarath Chandar et al., 2014; Tang and Wan, 2014), or ii) use translation systems (Zhou et al., 2015, 2016) in order to obtain aligned inputs to learn to extract features which work on both languages. This allows them to have their training and testing data in the same vector space. However many languages have no MT system, and it is extremely expensive to create one on a language-by-language basis.

3 Data

3.1 Affective Norms for English Words

For fine-grained sentiment regression we used Affective Norms for English Words (ANEW) (Bradley and Lang, 1999b). The creators of ANEW aimed to provide emotional ratings for a relatively large number of words in the English language.

ANEW falls inside an assumed framework where all human emotion is located in a space with three basic underlying dimensions, in which the entire range of human emotions can be arranged.

	Low Stimulus	High Stimulus
Arousal	Relaxed (2.39)	Infatuation (7.02)
Dominance	Victim (2.69)	Confident (7.68)
Valence	Death (1.61)	Beauty (7.82)

Table 1: Examples of words on each end of the spectrum for each of the three dimensions. Numeric stimulus value is placed within the brackets.

The first dimension, *valence*, ranges from *pleasant* to *unpleasant*; the second dimension, *arousal*, ranges from *calm* to *excited*; and the third dimension, *dominance*, ranges from *in-control* to *out-of-control*.

Bradley and Lang (1999b) used a nonverbal pictographic measure, the Self-Assessment Manikin (SAM) (Bradley and Lang, 1994), to measure stimuli across these three dimensions. The figures in the SAM consist of bipolar scales depicting different values along each of the three emotional dimensions. For example, when considering valence SAM ranges from a frowning unhappy figure to a smiling happy figure, and similarly for the other dimensions. Using this test, Bradley and Lang were able to arrive at a numerical value for each dimension ranging from 1 to 9; where 1 is the low stimulus (*unpleasant, calm, in-control*) and 9 is the high stimulus (*pleasant, excited, out-of-control*).

Warriner et al. (2013) used crowd-sourced numerical ratings directly and anchored their scales in the same direction as previous work done by Bradley and Lang. Warriner et al. also showed that the usage of numerical ratings through online crowd-sourcing highly correlates with SAM results from ANEW.

3.2 Word Lists

Translation Word List For the process of learning a translation matrix from one language to the other, a lexicon of approximately $10,000$ English words was obtained online by scraping the most commonly used words as determined by n-gram frequency analysis in Google's "Trillion Word Corpus"[3]. The lexicon was then translated using Google Translate[4] in order to obtain corresponding words in Spanish and traditional Chinese. For alignment lists of smaller sizes during experimentation, a random subset of the larger list was selected. During the selection we discarded any words which were not in the target language vector space and whose translation was not in the source language(s) vector space(s).

[3] https://github.com/first20hours/google-10000-english

[4] https://translate.google.com/

Binary Sentiment Word List For the task of binary sentiment classification we used a list[5] curated by Hu and Liu (2004b) containing both positive and negative English opinion words (or sentiment words). Google Translate was used to translate the list into the other languages to obtain cross-lingual word pairs. During training and testing, we made sure to balance the dataset and to discard words that were not in the vector space of the target language or whose translation was not in the vector space of the source language(s).

3.3 Twitter Data

In order to emulate previous work for comparison, we follow the same procedure for the procurement of data. Following Hu et al. (2013), Tang et al. (2016), we scrape Twitter for positive and negative tweets, defined as containing a positive or negative emoticon, as manual labeling of a large amount of sentences from other sources is not feasible. We scraped 5 million positive and 5 million negative tweets.

The Twitter data was preprocessed in the following ways:

- The tweets were tokenized using NLTK's tweet tokenizer (Loper and Bird, 2002), as opposed to the previously used TwitterNLP (Gimpel et al., 2011), but we believe that the differences between the two tokenizers are not significant.
- URLs, hashtags, and "@" symbols were removed.
- Tweets fewer than 7 words long (after the previous steps) were removed.

3.4 Word Embeddings

For the second task of this paper, we used precomputed word embeddings in multiple languages.

English Vector Space Model For English, we used a model pre-trained on part of the Google News dataset (which is composed of approximately 100 billion words).[6] The words are represented by 300-dimensional vectors.

Spanish Vector Space Model For the Spanish word embeddings, we opted to use a model pre-trained on the Spanish Billion Word Corpus (Cardellino, 2016). It consists of just under 1.5 billion words compiled from a variety of Spanish resources. As with the English model, the words are represented by 300-dimensional vectors.

[5] https://github.com/williamgunn/SciSentiment

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

Chinese Vector Space Model For Chinese word embeddings, we learned our own vector representations using a Wikimedia dump[7] of around 250,000 articles composed of around 150 million words from articles in both simplified and traditional Chinese. We used OpenCC[8] to translate the articles in simplified Chinese to traditional. To segment the text into tokens we used Jieba[9]. Finally, to create the actual word embedding model we used Gensim (Řehřek and Sojka, 2011) with the minimum count set to 1, using continuous bag of words (CBOW), a window of 8, and vector dimension set to 300.

3.5 Review Data

In this subsection, we discuss the data used to replicate the review classification task done by Chen et al. (2016) as a means of validating the usability of our model. This experiment was done using only English and Chinese because there was no Spanish data for this task and the Arabic review data-set that Chen et al. (2016) used was not freely available.

Labeled English Reviews Following Chen et al. (2016), we obtained a balanced dataset of 700,000 reviews of businesses on Yelp from Zhang et al. (2015) with their sentiment ratings as labels ranging from 1 for very negative to 5 for very positive.

Labeled Chinese Reviews Here we use a dataset from Lin et al. (2015). Their work provides hotel reviews, with labels ranging from 1 for very negative to 5 for very positive. In order to fairly compare our work with that of Chen et al. (2016), we use 10,000 reviews for model selection, and another unseen 10,000 as our test set.

4 Enriched Embedding

4.1 Methods

Here we present the techniques used to learn sentiment-enriched embeddings. For consistency with previous work (Tang et al., 2016), we will first describe the techniques used to capture traditional context-based word embeddings, followed by the techniques used to encode sentiment polarity. We will then describe how we combine the two models together to enrich the context-based sentiment embeddings. The structure of described models is presented in Figure 1.

[7] https://dumps.wikimedia.org/zhwiki/latest/
[8] https://github.com/BYVoid/OpenCC
[9] https://github.com/fxsjy/Jieba

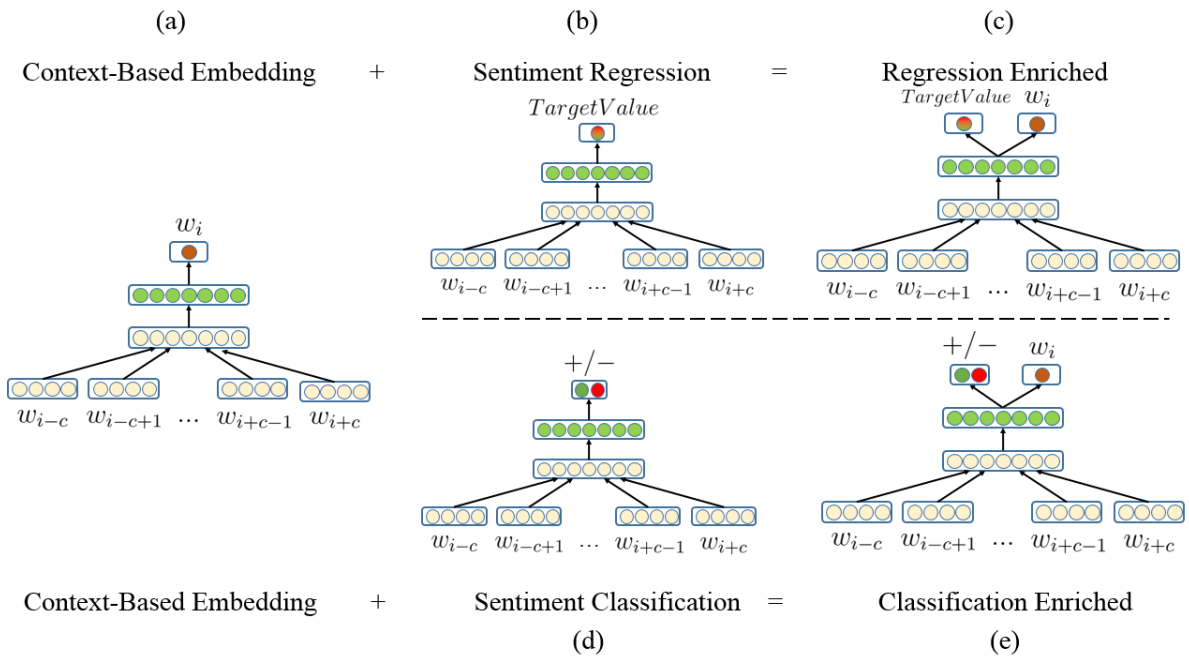


Figure 1: The neural network architectures used in experiments. (a) Context-Based Embeddings (CBOW). (b) Sentiment Regression (ANEW). (c) Combined context, sentiment regression loss. (d) Sentiment Classification (Tweets). (e) Combined context, sentiment classification loss.

Unlike previous work, we make use of the TensorFlow library (Abadi et al., 2016). Code for our both the traditional CBOW and enriched sentiment model are provided with the supplemental data.

4.1.1 Context-Based Embeddings

We will focus specifically on the CBOW technique and how it was modified and extended for this work. The traditional CBOW approach attempts to predict a word w_i given a context h_i, which is composed of $\{w_{i-c}, ..., w_{i-1}, w_{i+1}, ..., w_{i+c}\}$ where c is the context size. That is, given the surrounding context words preceding and following a given word, we try to automatically predict the current word.

The lookup layer maps each word to the corresponding continuous vector representation using a lookup table. For our CBOW implementation, the output of the lookup layer would be the mean of the context vectors (Equation 1), but for the sentiment-embedding models the output is the concatenation of the extracted context vectors into a new vector as is with previous work (Equation 2).

$$O_{lookup} = \sum_{i}^{2c} \frac{e_i}{2c} \tag{1}$$

$$O_{lookup} = [e_{i-c}, e_{i-c+1}, ..., e_{i+c-1}, e_{i+c}] \tag{2}$$

In either case, the output of the lookup layer is then passed to a linear layer, such that:

$$O_{l1} = W \times O_{lookup} + b \tag{3}$$

We experimented with and without the addition of an *htanh* non-linearity such that:

$$htanh(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases} \tag{4}$$

As with most embedding creation, we rely on noise-contrastive estimation to speed-up the training process, instead of the normal softmax.

4.1.2 Sentiment-Based Embeddings

4.1.2.1 Binary Classification

The network used to encode sentiment in the case of binary classification, Figure 1 (d), is a re-implementation of the method that is described by Tang et al. (2016). During training, the gold labels would be a $[1,0]$ if a tweet was positive and $[0,1]$ if negative. The initial layers are all equivalent to those described above, and the final layer is a softmax layer with a cross-entropy error between the gold and predicted distributions as the loss of this network.

4.1.2.2 ANEW Regression

We wanted to replicate improvements of previous work without having the corpus limitation that came with the approach. Previous work (Abdalla and Hirst, 2017) has shown that it is possible to predict the ANEW values of a word given embeddings. Here we describe three approaches tested using ANEW: i) Valence Regression, ii)Valence Window Regression, iii) Full ANEW Regression. The approaches work as follows:

1. Given an un-enriched word embedding model, and the ANEW lexicon, train a simple linear regressor that predicts the valence of a word given the vector representation of the word for the first two approaches, and three regressors that predicts each ANEW axis for the third approach. We used linear SVM as our regressor.

2. For each word in the vocabulary, use the trained regressor(s) to predict the valence (and arousal and dominance for the third approach) of the word. These predicted values will serve as the "gold" label during training of the enriched embedding.

3. *Approach i)*: When training the embedding, instead of binary classification, treat this as a regression problem where the task is to predict the associated valence from Step 2, given h_i. The error function used for this task is the mean squared error (MSE).

   Approach ii): When training the embedding, instead of binary classification, we again treat this as a regression problem. However, instead of predicting the valence of w_i, here we attempt to predict the average valence of each word in context including the current word (*i.e.*, the average of predicted values for each word in the set $\{w_i$ and $h_i\}$). The error function used for this task is the mean squared error (MSE).

   Approach iii):When training the embedding, instead of simple regression as with the previous approaches, here we use the window approach described above three times, one for each ANEW dimension. The entire sentiment loss, where any loss is denoted by $\mathbb{L}$, is an equal split between MSE loss for each of the axes:

$$\mathbb{L}_{\text{senti}} = \left(\frac{1}{3}\mathbb{L}_{\text{arousal}} + \frac{1}{3}\mathbb{L}_{\text{dominance}} + \frac{1}{3}\mathbb{L}_{\text{valence}}\right) \tag{5}$$

4.1.3 Enriching Context-Based Embeddings with Sentiment

4.1.3.1 Hybrid Classification Models

MODELS: *All variations of SE-HyPred*

In this model, the context-based embeddings are combined with the original binary classification sentiment embeddings. The combined loss function is:

$$\mathbb{L}_{\text{combined}} = \alpha\mathbb{L}_{\text{context}} + (1-\alpha)\mathbb{L}_{\text{classification}} \tag{6}$$

Where $\alpha = 0.5$ for consistency with previous work. All of the layers except for the final predictive layers are shared as shown in Figure 1(c).

4.1.3.2 Hybrid Regression Models

MODELS: *All variations of SE-HyReg*

In these models, Figure 1(e), the context-based embeddings are combined with the variety of regression-based sentiment models described in Section 4.1.2.2. The combined loss function is:

$$\mathbb{L}_{\text{combined}} = \alpha\mathbb{L}_{\text{context}} + (1-\alpha)\mathbb{L}_{\text{regression}} \tag{7}$$

As with before, $\alpha = 0.5$, and all of the layers except for the final predictive layers are shared between both the context-based and sentiment-based embeddings.

4.1.4 Enriching Context-Based Embeddings with Semantic Information

We had briefly highlighted, in Section 2 (Previous Work), enrichment algorithms that made use of semantic information to enrich embeddings (as opposed to sentiment information). However, as sentiment is often encoded in semantics, we describe the approach that had the best performance on sentiment tasks from the work of Faruqui et al. (2015).

The best-performing model (on their tests of sentiment analysis) was the model retrofitted with PPDB. Given traditionally created word-vectors, the enrichment of these word vectors is then done by iteratively transforming the vector space such that the words which are paraphrases of each other also have closer representation in the vector embedding space. We used 10 iterations, as recommended by the authors. In the remainder of this work, we refer to this model as CBOW+PPDB.

4.1.5 Network Parameters

In this section, we describe the network parameters used for the results presented in the experiments. For consistency with previous work, and to enable fair comparisons, we used the same parameters for all of the shared networks (*i.e.,* same context window, and layer parameters, *etc.,* so that without enrichment they would have encoded the same information, thereby measuring only the effects of the enrichment process). Where possible we used the parameters described by Tang et al. (2016). Where such parameters were not defined, we used ones we thought made sense given the data and models at hand, making sure to stay consistent throughout all of the networks.

The word embeddings were initialized from a random uniform distribution $U(-0.01, 0.01)$. The weights of the linear layers were initialized from a random uniform distribution function $U(\frac{-0.01}{\text{layer\_length}}, \frac{0.01}{\text{layer\_length}})$. The window size was set to 7 (3 preceding words, and 3 following words). The embedding size was set to 50. AdaGrad was used for parameter updating, with an initial value of 0.1.

The minimum occurrence requirement (often used to filter non-words and misspellings) was set to 10. The threshold for down-sampling high-frequency words was set to 10^{-3}. 64 words were negatively sampled. The batch size was set to 200, and we did 5 iterations (epochs) over the corpus.

4.2 Experiments and Results

We conducted experiments to determine whether the sentiment-enriched embeddings improve performance for sentiment-related tasks (*e.g.,* binary word sentiment classification), and what their effect is on traditional tasks that are not explicitly sentiment-related (*e.g.,* document classification).

The experiments are split into two main categories: (1) Word-level (upstream) tasks: i) Binary sentiment classification, ii) Fine-grained sentiment regression, iii) Analogy evaluation, and (2) Sentence-level (downstream) tasks: i) SemEval 2013 tweet classification, ii) Document classification.

Table 2 defines the different network setups used for experimentation and assigns each of them a name by which they will be referred to in later sections.

Embedding	*htanh*	Sentiment Output
Word2Vec (CBOW)	–	Classification
CBOW+PPDB	–	Vectors retrofitted using PPDB, (Faruqui et al., 2015)
SE-HyPred	+	Classification, (Tang et al., 2016)
SE-HyPred-S	–	Classification
SE-HyReg-V	+	Valence regression on current word
SE-HyReg-VW	+	Valence regression on current context window + word
SE-HyReg-VWS	–	Valence regression on current context window + word
SE-HyReg-VADWS	–	Regression with all ANEW on current context window + word
SE-HyReg-VADWS (R=1)	–	Regression with all ANEW on current context window + word

Table 2: The models used in our experiment. The *htanh* column uses '+' to denote the inclusion of the non-linearity, and '–' for exclusion. 'SE-HyPred' denotes 'S'entiment 'E'nriched embeddings where the task at hand was classification ('PRED'diction). 'SE-HyReg' denotes 'S'entiment 'E'nriched embeddings where the task at hand was 'Reg'ression. 'V' = regression on valence, 'A' = regression on arousal, 'D' = regression on Dominance, 'W' = window approach, 'S' = no *htanh*, 'R'- if we recurse (use ANEW classifier trained on enriched embeddings).

4.2.1 Sentiment Tasks

4.2.1.1 Word-Level Binary Sentiment Classification

In this task, we tested whether word embeddings enriched with sentiment information resulted in improved performance for predicting word-level sentiment classification. This was a task also done in previous work, allowing us to compare our results with predecessors to verify implementation and findings. A classifier whose task was to predict whether a word is positive or negative was trained on the following three lexicons: BL-Lexicon (Hu and Liu, 2004a)[10], MPQA (Wilson et al.,

[10]https://www.cs.uic.edu/∼liub/FBS/sentiment-analysis.html

2005)[11], and NRC-Lexicon (Mohammad and Turney, 2013)[12].

We performed both 5-fold and 10-fold cross-validation to ensure consistency in result comparison with previous work. We trained supervised classifiers (linear SVMs), and present the averaged training accuracy in Table 3. The classes within the data are heavily unbalanced, and therefore we balanced the dataset before training and testing.

Embedding	5-fold CV			10-fold CV		
	BL	**MPQA**	**NRC**	**BL**	**MPQA**	**NRC**
Word2Vec (CBOW)	66.9	65.9	64.9	68.4	66.5	65.0
CBOW+PPDB	75.8	**74.2**	65.7	76.1	**74.4**	65.8
SE-HyPred	75.1	70.4	66.6	75.1	70.4	66.6
SE-HyPred-S	**77.3**	73.0	**70.0**	**77.6**	73.2	**69.9**
SE-HyReg-V	68.7	66.9	63.1	69.0	67.1	63.0
SE-HyReg-VW	69.5	67.0	63.7	69.4	67.0	63.8
SE-HyReg-VWS	76.0	<u>74.1</u>	<u>68.7</u>	75.7	<u>74.1</u>	<u>68.7</u>
SE-HyReg-VADWS	76.3	72.5	68.1	76.3	73.1	68.0
SE-HyReg-VADWS (R=1)	<u>76.8</u>	73.9	<u>68.7</u>	<u>77.4</u>	<u>74.1</u>	<u>68.7</u>

Table 3: The accuracies from the word-level binary sentiment classification task. The best scores in each column are bolded, and the second-best performing values are underlined.

Our results mimic the trend observed by Tang et al. (2016) for the first two models. The absolute difference in performance between our results and theirs could be attributed to several reasons. Firstly, although both methods use Twitter data, the exact tweets used, their topic of conversation and such, is not something we could control for and therefore might have had an impact on the results. Additionally, Tang et al. state only that they used a "trained supervised classifier" without specifying which classifier. This could also be a cause behind the difference in performance. Last, we balanced our testing and training dataset, but it is not clear whether the previous authors have done the same.

However, although the specific numbers are different, the general trend between models that appeared in the previous paper is preserved. Removing the non-linearity from the network (approaches containing "-S") tends to perform better than models with the *htanh* layer. We also see that considering the average of the current context (approaches containing "-W") leads to a small improvement over only the current word. SE-HyPred-S (implementation of previous work, but after removal of *htanh*), and SE-HyReg-VADWS both outperform the baseline and the best of pre-

[11] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

[12] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

vious work. The latter does so while not requiring the entire corpus to be labeled. Faruqui et al. (2015)'s CBOW+PPDB performs well on MPQA but is outperformed by the baseline on NRC, and the our techniques for both BL and NRC.

4.2.1.2 Word-Level Fine-Grained Sentiment Regression

The previous experiment showed that the new methods improved binary sentiment classification, but we sought to establish whether this improvement in performance (for binary-sentiment classification) was present at fine-grained level. To do this, we regressed on the valence dimension of ANEW lexicon. We used a linear model (SVR), 5-fold CV, and present the results in Table 4.

Embedding	Valence MSE	Binary F1-Score	Ternary F1-Score
Word2Vec (CBOW)	1.46	72.0	56.5
CBOW+PPDB	1.36	69.1	56.6
SE-HyPred	1.35	**75.7**	56.9
SE-HyPred-S	1.28	<u>74.8</u>	**59.7**
SE-HyReg-V	1.43	71.4	54.3
SE-HyReg-VW	1.36	72.7	53.1
SE-HyReg-VWS	**1.25**	74.1	57.6
SE-HyReg-VADWS	**1.25**	<u>74.8</u>	<u>57.7</u>
SE-HyReg-VADWS (R=1)	<u>1.26</u>	74.1	57.4

Table 4: Results for both: i) word-level fine grained sentiment regression (mean-squared error (MSE)) (first column) and ii) sentence-level SemEval tweet sentiment classification (for both the binary and ternary setups), macro-F1 score (second and third column respectively). The best scores in each column are bolded, and the second best performing values are underlined.

Again, removing the *htanh* layer (approaches containing the "-S" suffix) improves performance. As aforementioned, models which take into account the average sentiment of the context instead of just the current word (approaches containing the "-W" suffix) also tend to perform better. Unlike the results in the binary classification task, we see that regressing leads to the better performing models (as opposed to a tie in performance). Here Faruqui et al. (2015)'s CBOW+PPDB does not beat the baseline of Tang et al. (2016)'s best model.

4.2.1.3 SemEval — Sentence-Level Sentiment Classification

Having confirmed that enrichment improves performance when it comes to word-level sentiment tasks, we show that the improvements in performance carry over to the sentence-level sentiment

tasks. To do this, we attempted the SemEval Task 2 — Sentiment analysis in Twitter (Nakov et al., 2013), involving the sentiment classification of sentences.

In order to do sentence-level sentiment analysis, we used the principle of compositionality (Frege, 1948) to construct sentence-level features. The compositionality principle states that the meaning of a sentence, or other expression, is determined by the smaller units from which it is composed (words in this case). Thus we used max, average, and minimum pooling layers to construct sentence representations from the individual words.

The data used was the training and development datasets provided by SemEval 2013. As the test-dataset "SemEval-2013Test" was not publicly available, we changed the experimental protocol. We trained classifiers on the training dataset "SemEval-2013Train", using 5 fold cross-validation, selecting the highest performing fold, and testing on the development set "SemEval-2013Dev" (which we treated as our test-set). We must note that not all of the specific tweets in either of the sets could be obtained, as the original tweets had either been deleted or had their access policy changed.

We performed two different classification tasks: i) Binary positive/negative sentiment classification and ii) ternary positive/neutral/negative sentiment classification. For the results, shown in Table 4, we present Macro-F1 which is defined as the average of F1-Scores across all of the categories.

We reached numbers close to those of Tang et al. (2016), and observe a similar trend (but smaller in magnitude) when it comes to improvement over the baseline. Interestingly, we see that the original model developed by Tang et al. does indeed perform the best in the binary test, but not by a large margin over the new methods (less than 2 points). However the original model ('SE-HyPred') does not perform as well in the ternary case when compared to the new models. As before Faruqui et al. (2015)'s CBOW+PPDB does not beat the baseline of Tang et al. (2016)'s best model.

4.2.2 Non-Sentiment Tasks

We have shown that word embeddings enriched with sentiment information during creation result in more meaningful embeddings when it comes to tasks that are directly related to sentiment analysis. However, previous work did not study the effect of enriching such embeddings on non-sentiment related tasks. It may be that the gains in sentiment-related tasks come at a price of the general embedding quality (given that the loss function weighs context and sentiment equally). The following tasks study this effect to see whether the enriched embeddings can be used for unrelated tasks.

Embedding	P@1	P@5	P@10
Word2Vec (CBOW)	49 (0.28%)	306 (1.72%)	533 (3.00%)
CBOW+PPDB	66 (0.37%)	274 (1.54%)	488 (2.75%)
SE-HyPred	38 (0.21%)	150 (0.85%)	246 (1.39%)
SE-HyPred-S	127 (0.72%)	484 (2.73%)	854 (4.81%)
SE-HyReg-V	42 (0.24%)	232 (1.31%)	371 (2.09%)
SE-HyReg-VW	32 (0.18%)	140 (0.79%)	266 (1.50%)
SE-HyReg-VWS	207 (1.17%)	611 (3.44%)	1016 (5.73%)
SE-HyReg-VADWS	**226 (1.27%)**	**700 (3.95%)**	**1160 (6.54%)**
SE-HyReg-VADWS (R=1)	181 (1.02%)	674 (3.80%)	1079 (6.08%)

Table 5: Results from the Google analogy evaluation. Both the total count, and percentage of total examples in test-set are shown. The best scores in each column are bolded, and the second-best performing values are underlined.

4.2.2.1 Embedding Analogy Evaluation

The first non-sentiment task we studied was Google's Embedding Analogy Task. Embeddings are tested for their ability to predict the fourth word from the first three words, such that the first and second word have a relationship to each other that is equivalent to that of the third and fourth word: (*e.g., Athens* is to *Greece* as *Madrid* is to *Spain*). This can mathematically be represented as attempting to find vector v such that:

$$\arg\max_{\vec{v}\in V} cos(\vec{v}, \vec{v}_2 - \vec{v}_1 + \vec{v}_3) \tag{8}$$

Table 5 presents both the total count and the percentage of the entire dataset to be captured. The notation P@N is used to denote the number or percentage of times the correct vector is within the N-closest vectors to $\vec{v}_2 - \vec{v}_1 + \vec{v}_3$. We removed words that were not found in the training data and thus had no trained embedding. The results are quite poor, which is expected for normal embedding trained on "proper" English text (Jastrzebski et al., 2017), and we believe the problem is further exacerbated by the fact that Twitter data itself does not discuss all of the topics/relationships which are represented in the dataset.

We observe that the models introduced by Tang et al. actually resulted in a decrease in embedding quality for this task. Faruqui et al. (2015)'s CBOW+PPDB increased performance slightly for P@1, but decreased for the other measures, and was largely eclipsed by the newer techniques. As expected, the removal of *htanh* results in a notable increase in performance when compared to all models with the non-linearity. We also notice here that looking at singular values seems to per-

Embedding	F1 (Accuracy)
Word2Vec (CBOW)	21.1 (74.5)
CBOW+PPDB	19.6 (72.6)
SE-HyPred	41.8 (79.9)
SE-HyPred-S	<u>46.7 (81.9)</u>
SE-HyReg-V	37.1 (79.3)
SE-HyReg-VW	35.3 (77.4)
SE-HyReg-VWS	46.5 (84.4)
SE-HyReg-VADWS	45.1 (84.2)
SE-HyReg-VADWS (R=1)	**48.7 (85.4)**

Table 6: Results of the document classification task. Since the dataset is not balanced, both Macro-F1 and Accuracy are presented. The best scores in each column are bolded, and the second-best performing values are underlined.

form better than the window approach, but window + all dimensions results in the best performance against all models.

4.2.2.2 Document Classification Performance

We wanted to study whether the increased performance on the analogy task (hinting at an improved embedding in the general sense) would carry over to downstream non-sentiment related tasks. To study this, we considered the classic task of document classification. Given N classes, and unlabeled documents, we asked whether we could learn a classifier for the documents.

For this problem, we used the R8 dataset (Cardoso-Cachopo, 2007), which is composed of 7674 single-labeled Reuters news articles split into 8 topics. Cardoso-Cachopo removed any document which could have been assigned more than a single label. As the classes were heavily unbalanced during both training and testing, we present both the macro-F1 Score and the unweighted accuracy as well. Table 6 shows that again, the removal of *htanh* layer results in a significant increase in embedding performance. Here CBOW+PPDB performs worse than both baselines. We again see that regression on ANEW values outperforms previous approaches, and competes with improved versions of past approaches that require much labeled data.

5 Cross-Lingual Translation

5.1 Methods

5.1.1 Translation Matrix Technique

As described by Mikolov et al. (2013b), the translation matrix technique assumes that we are given a set of word pairs and their associated vector space representations. More specifically, we are given j word pairs, $\{x_i, z_i\}_{i=1}^{j}$ where $x_i \in \mathbb{R}^n$ is a word vector from the source language of word i and $z_i \in \mathbb{R}^m$ is the word vector representation of the corresponding translated word in the target language.

We then want to find a transformation matrix W such that Wx_i approximates z_i. We learn this by solving the following optimization problem:

$$\min_{W} \sum_{i=1}^{j} ||Wx_i - z_i||^2$$

Instead of solving with stochastic gradient descent, we instead opt to use the closed-form solution.

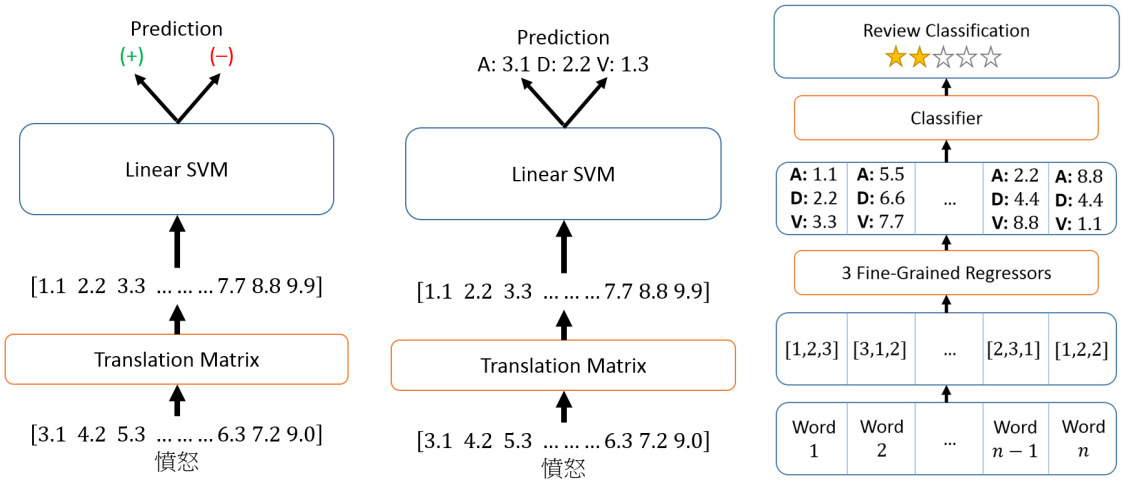
To translate a word from source language to target language, we can map it using $z = Wx$, and then find the closest word in that language space using cosine similarity as the measure of distance. The method of testing was Monte Carlo cross-validation run 10 times with a split of 90% training data and 10% test data.

5.1.2 Models

5.1.2.1 Binary Sentiment Analysis Model

The sentiment analysis model, Figure 2(a), is a simple linear support vector machine (SVM) classifier. Implemented using Sci-kit Learn's *SGDClassifier* function (Pedregosa et al., 2011), this model takes a word represented as a vector with dimension of 300 and outputs a prediction of either -1 or $+1$ (for negative and positive respectively). The classifier itself performs stochastic gradient descent (SGD) with $l2$ regularization to arrive at the best classification.

The training procedure of this model is quite simple and involves only the target language. The model is trained only on word embeddings from the target language but tested on embeddings returned by the translation of words originally from the source language(s). We made sure that the English translation of the test words had not been seen before in training. The training/testing split was changed to 80% and 20% from the previous 90%/10% to account for the smaller number of examples in the dataset (and the fact that we wanted to test on a representative sample of the data).



(a) Binary Sentiment Classification (b) Fine-Grained Sentiment Regression (c) Cross-lingual Review Classification

Figure 2: The three models used in the experiments. All the models are trained only on English word vectors, and are tested on Chinese and Spanish word vectors which are transformed to English word vectors by the matrix translation method.

5.1.2.2 Fine-Grained Sentiment Analysis Model

The fine-grained sentiment analysis model in Figure 2(b) is a regression model to predict the ANEW values for each of the three dimensions. For our experiment we built a regressor for each of the 3 dimensions whose input is a 300-dimension vector and whose output is a real number from 1 to 9. The regressor used was a Bayesian Ridge regressor, which estimates a probabilistic model of the regression problem. The prior for the parameter w is given by a spherical Gaussian:

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1} I_p) \tag{9}$$

The priors α and λ are chosen to be gamma distributions, the conjugate prior for the precision of the Gaussian. The resulting model is similar to that of the Ridge regression. The model was implemented using Sci-kit Learn's with hyperparameters alpha_1 and alpha_2 set to 1.

The training procedure of this model is quite simple and similar to that of the previous model. Again it only trains on the vector of the target language, and is tested purely on words from the source language whose translation into the target language were not seen in the training of the regressor (so that there may be no chance of skewing the results). The training/testing split here was 75%/25%.

5.1.2.3 Review Classification Model

For the task of review classification, another model Figure 2(c), was built to make use of the previously described fine-grained sentiment analysis model. The classifier used for this task is a logistic regression classifier. For a given review r_i in the target language, composed of n words, we construct a $1 \times Max\_length$ sentiment vector (where Max_length is the number of words of the longest target review). We construct this vector by passing in the word embedding for every word into the sentiment analysis model and placing the resulting values (of 1 to 9) into the constructed array. The array is then padded with 0's in order to make it of length Max_length.

For reviews in the source language, the process is similar, with the only change being that the words are first translated from their original vector space to that of the target language before being passed into the sentiment classifier. Once the review vector has been constructed, it is passed to the classifier to produce a classification of 1 to 5.

As with the previous classifier, the training procedure is quite simple and involves only the target language (*i.e.*, the classifier is trained only on reviews which are originally from the target language), but it is tested on reviews only from the source language.

5.2 Experiments and Results

5.2.1 Translation Accuracy

	1000 WORDS		**4500 WORDS**		**8500 WORDS**	
Translation	P@1	P@5	P@1	P@5	P@1	P@5
EN → ES	20.3	34.6	33.42	46.13	34.79	47.79
EN → CN	2.4	11.6	7.60	20.29	8.87	23.01

Table 7: Accuracy of the word translation method. P@1 and P@5 represent Top-1 and Top-5 accuracy respectively. The columns are split by number of words used in both testing and training the translation matrix.

We first measure the accuracy of the matrix translation method. The data used is described in section 2.1. Table 7 shows the effect of training size (number of words) on the accuracy of the translation matrix method. As expected (and previously shown by Mikolov), the translation accuracy increases with more training examples, as shown in Figure 3. It is also interesting to note that Chinese, which is less like English than Spanish is, also suffers a lower translation score across both categories. However, we see that this large drop is not represented significantly in later portions.

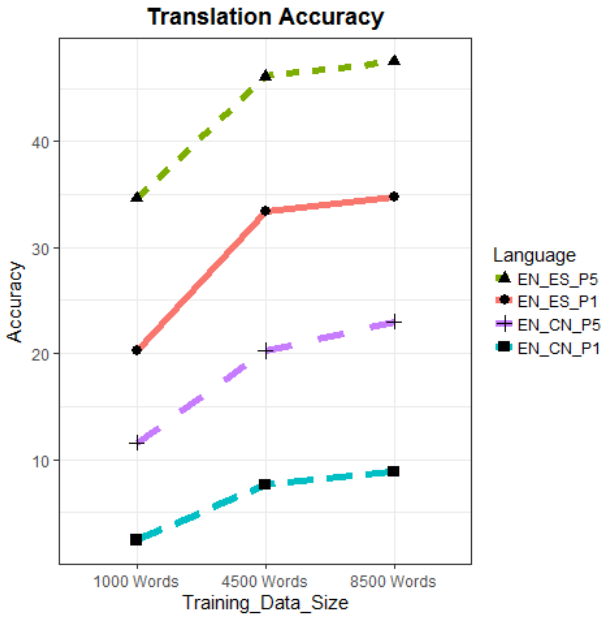


Figure 3: Top-1 and Top-5 Accuracy for Spanish and Chinese using various amounts of training data.

5.2.2 Binary Word Sentiment Classification

The second experiment tested the binary cross-lingual sentiment classification capabilities of the matrix translation method, *i.e.*, how well can we differentiate between positive and negative words of a language we have not seen before using a model trained only on English words? Here we used the binary sentiment word list described in section 2.1.2 in order to assess whether or not the translation matrix would preserve sentiment even with poor translation accuracy scores. Given that the classifier is trained only on the target language vectors, we used the translation matrices produced previously to translate a word from source to target language embedding space.

As we can see in Table 8, even with low translation accuracy, such as the 1000-word Chinese translation matrix, we are able to achieve good binary sentiment classification. Another observation to note is that a significant drop in translation accuracy results in only a relatively small drop in sentiment classification performance.

5.2.3 Fine-Grained Sentiment Analysis

In the third experiment, we tested the accuracy of cross-lingual regression when it comes to predicting a word's value in any of the 3 ANEW dimensions of *valence*, *arousal*, and *dominance*. That is to say, we attempted to predict the *valence*, *arousal*, and *dominance* of words in the source language, having only trained on the target language. As we saw in the last experiment, massive

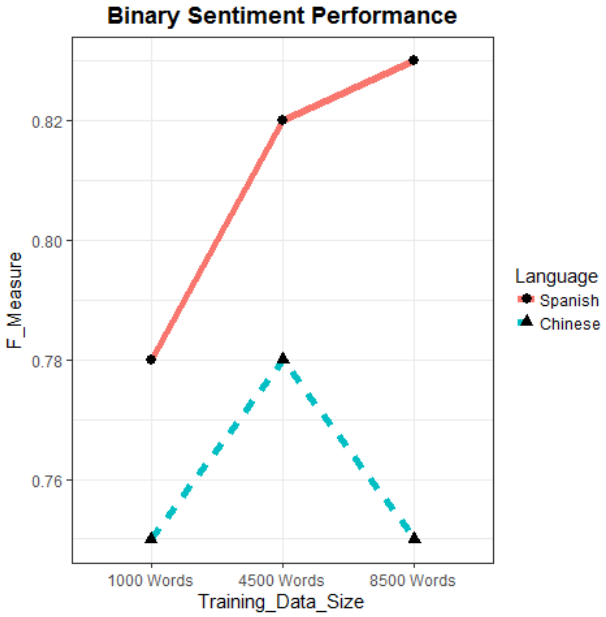
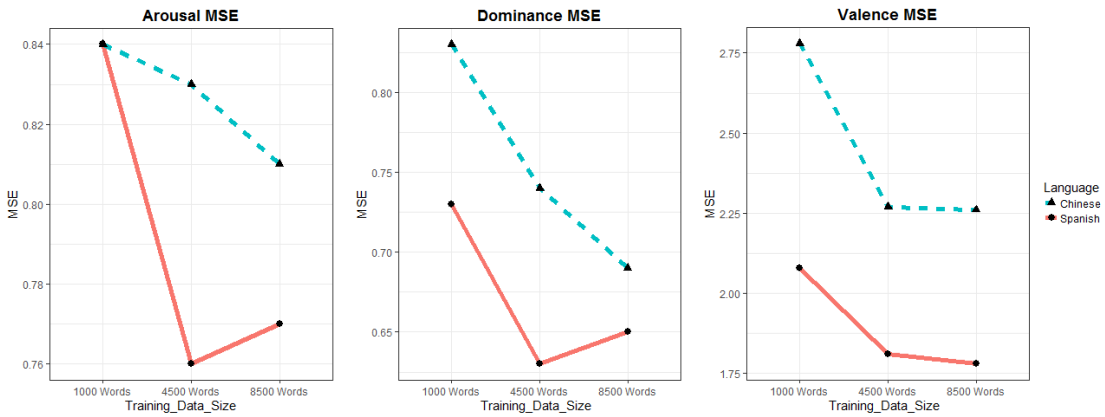


Figure 4: F-Measure of binary sentiment classifier with varying amounts of training data for both Spanish and Chinese.

	Spanish	**Chinese**
1000 Words		
Precision	0.77	0.76
Recall	0.79	0.74
F-Measure	0.78	0.75
4500 Words		
Precision	0.82	0.79
Recall	0.82	0.77
F-Measure	0.82	0.78
8500 Words		
Precision	0.83	0.80
Recall	0.83	0.77
F-Measure	0.83	0.75

Table 8: Results of the binary sentiment classification task for each language with each translation matrix.



(a) Arousal (b) Dominance (c) Valence

Figure 5: Mean squared error for each regression problem with varying amounts of training data for all three of the ANEW dimensions. (Lower is better).

drop-off in translation accuracy need not result in a massive drop-off in sentiment analysis. As this is a regression problem, Table 9 presents both the r^2 and the mean squared error (MSE) as measurements of model performance. Given the data's scale from 1 to 9 with an average standard deviation among participants for each word of 2.02, an average mean squared error of approximately 1 shows that our model has high predictive power.

5.2.4 Sentiment Classification of Reviews

In the fourth experiment, we sought to show that the regressor developed in section 3.3.2 could be used as a feature extractor in performing other tasks. To this end, we replicated the experiment by Chen et al. (2016), where we predicted hotel ratings from Chinese reviews using a model trained only on English restaurant reviews.

Chen et al. (2016) had two baseline models which they beat using their new model: i) a logistic regression classifier (line 1 in Table 10), and ii) a non-adversarial variation of adversarial deep averaging network (ADAN) (line 2), referred to as DAN (deep averaging network), which is one of the state-of-the-art neural models for sentiment classification. These were the only two models which did not make use of either labeled Chinese examples or a MT system, and therefore were chosen to serve as a fair comparison to our method. Both models use bilingual word embeddings as an input representation to map words from both languages into the same vector space. Our own model (line 3) uses logistic regression on sentence arrays created by predicting their ANEW values for each dimension to predict review scores.

We were able to match the accuracy of the baseline systems implemented by Chen et al. (2016) for Chinese reviews, Table 10.

	Spanish	Chinese
1000 Words		
Arousal	0.24 (0.84)	0.24 (0.84)
Dominance	0.31 (0.73)	0.23 (0.83)
Valence	0.48 (2.08)	0.32 (2.78)
4500 Words		
Arousal	0.33 (0.76)	0.28 (0.83)
Dominance	0.39 (0.63)	0.28 (0.74)
Valence	0.54 (1.81)	0.44 (2.27)
8500 Words		
Arousal	0.33 (0.77)	0.29 (0.81)
Dominance	0.38 (0.65)	0.31 (0.69)
Valence	0.54 (1.78)	0.43 (2.26)

Table 9: Results of the fine-grained sentiment regression task for each language with each translation matrix. Provided in the form of $r^2(MSE)$.

Approach	**Accuracy**
Logistic regression (BWE)	30.58%
Deep averaging network (DAN)	29.11%
Logistic regression (ANEW)	28.05%

Table 10: Model performance on sentiment extracted vectors versus previous approaches. Logistic regression on predicted sentiment (ANEW) values preformed similarly to both regression and DAN on BWEs.

These results demonstrate that our sentiment regressors encoded enough information into the sentence vectors to achieve similar results to the baseline models which took bilingual word embedding as input, and that the fine-grained sentiment model can be used to extract sentiment-based features for other tasks in languages where aligned data might be expensive to obtain.

6 Conclusion

In this work, we have introduced two new techniques: i) a novel and generalizable way of incorporating sentiment information into word embeddings without the need of a labeled corpora, and ii) a novel way of leveraging the wealth of labeled sentiment data in English to aid with sentiment analysis in languages with fewer resources.

We subsequently studied the effect of enrichment on both sentiment and non-sentiment related tasks, showing that the enriched embeddings seem to improve traditional (CBOW) embeddings both at the word level and the sentence level for all of the tasks tested. This observation suggests that sentiment information and other linguistic properties are not completely encoded by the distributional hypothesis; i.e. not fully extracted by looking at the direct neighbors of a word. In future work, we can extend the idea that representations of words can be further enhanced by incorporating external information (as evidenced with sentiment in this case). However, additional experiments are required to see if the observations hold for traditional text corpora (such as that of Wikipedia), although preliminary testing seems to indicate that they do.

Our work has also shown that the matrix translation method can be used to infer and predict cross-lingual sentiment. More notably we observe that: i) sentiment is preserved accurately even with sub-par translations, and ii) this low-cost approach also maintained fine-grained sentiment information between languages. These observations were further cemented by the experiments performed where we showed that with as low as 11% P@5 translation accuracy, we are still able to predict sentiment with 75% accuracy.

Throughout the experiments, we saw the general trend of reduced error and increased accuracy with more training data. In our fine-grained experiment we showed that it was possible to get a MSE lower than 2 on a scale from 1 to 9. However, the increases in accuracy start to diminish around 8500 words for both experiments. The cause of this leveling could be an inherent limitation of the either the translation technique used or the classification or regression algorithms used. In order to determine which, more experiments need to be done by exploring different translation algorithms (CCA (Faruqui and Dyer, 2014), Orthogonal transformation (Smith et al., 2017)), and studying whether using non-linear functions improves sentiment value prediction.

The most obvious extension to the work presented here is combining cross-lingual sentiment analysis with the mono-lingually enriched embeddings. In the next iteration of this work, we

hope to study the effect of translation on enriched word embeddings. Using enriched embeddings might allow us to break the plateau in performance observed with 8500 words. Studying the topological sentiment map produced by enriched embeddings might lead to insights regarding how the enrichment itself works.

7 Discussion

We hypothesize that having sentiment framed as regression instead of classification is conducive to achieving a sentiment gradient in the embedding space (when compared to pairwise classification). Framing the problem as regression also dealt with limitations with previous work; their inability to handle sentences with a neutral sentiment score (as it is unclear how one would accurately extract such labels from online datasets without manual human labeling) Tang et al. (2016). Yet the ability to do this is crucial for an actual model of human emotions, as not all documents carry sentiment value.

It was interesting for us to observe that our proposed techniques outperformed Faruqui et al. (2015)'s technique on Google's Analogy task; as we only enrich for sentiment, while they enrich using semantic information. However, the results of Faruqui et al. (2015)'s technique when compared to our own have to have certain factors taken into consideration. Their work did not study the effects of enrichment on text from social platforms which is different from that of newspapers (the source of their word-embeddings). The word-embeddings of their original work were of size 300, and trained on a massive set of documents when compared to our 50-dimensional word embeddings and 10 million tweets. All of these could be reasons why Faruqui et al. (2015)'s method did not perform as well.

The techniques used here could be used to study changes in sentiment for words in a single language over time, leading to new insights or re-affirming old ones. Future analysis could compare different transformations and their effect on sentiment analysis. The ability to produce a "stable" topological sentiment map could also be used to evaluate algorithms which create embeddings as well.

Both works lack the ability to handle polysemy and this subsequently impacts both the embedding quality but also sentiment prediction (as different words carry different sentiment values in different contexts). Future work could possibly try to incorporate sentiment into embeddings that are also enriched with polysemic information.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Mohamed Abdalla and Graeme Hirst. Cross-lingual sentiment analysis without (good) translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 506–515, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

Mohamed Abdalla, Magnus Sahlgren, and Graeme Hirst. Incorporating Sentiment Analysis into Word Embeddings without Labeled Corpora. *Submitted*, 2018.

AR Balamurali, Aditya Joshi, and Pushpak Bhattacharyya. Cross-lingual sentiment analysis for Indian languages using linked WordNets. In *Proceeding of International Conference on Computational Linguistics (COLING)*, pages 73–82. Association for Computational Linguistics, 2012.

Yoshua Bengio and Greg Corrado. BilBOWA: Fast bilingual distributed representations without word alignments. *arXiv preprint arXiv:1410.2455*, 2015.

Margaret M Bradley and Peter J Lang. Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49–59, 1994.

Margaret M Bradley and Peter J Lang. Affective Norms for English Words (ANEW): Instruction manual and affective ratings. Technical report, Technical Report C-1, The Center for Research in Psychophysiology, University of Florida, 1999a.

Margaret M Bradley and Peter J Lang. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical report, Center for Research in Psychophysiology, University of Florida, 1999b.

John Bullinaria and Joseph P. Levy. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior Research Methods*, 44:890–907, 2012.

Cristian Cardellino. Spanish Billion Words Corpus and Embeddings, March 2016. URL http://crscardellino.me/SBWCE/.

Ana Cardoso-Cachopo. Improving Methods for Single-label Text Categorization. PhD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

Xilun Chen, Ben Athiwaratkun, Yu Sun, Kilian Q Weinberger, and Claire Cardie. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*, 2016.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning crosslingual word embeddings without bilingual corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 1285–1295, 2016.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Multilingual training of crosslingual word embeddings. In *Proceedings of the European Chapter of the Association for Computational Linguistics* (EACL), pages 893–903, 2017.

Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (EACL), pages 462–471, 2014.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615. Association for Computational Linguistics, 2015.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

Gottlob Frege. Sense and reference. *The Philosophical Review*, 57(3):209–230, 1948.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, 2013.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.

Tatsunori Hashimoto, David Alvarez-Melis, and Tommi Jaakkola. Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics*, 4: 273–286, 2016. ISSN 2307-387X.

Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*, 2013.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004a.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD-2004), pages 168–177. ACM, 2004b.

Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 607–618. ACM, 2013.

Sarthak Jain and Shashank Batra. Cross-lingual sentiment analysis using modified BRAE. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 159–168, 2015.

Stanisław Jastrzebski, Damian Leśniak, and Wojciech Marian Czarnecki. How to evaluate word embeddings? On importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv:1702.02170*, 2017.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing crosslingual distributed representations of words. In *Proceedings of the International Conference on Computational Linguistics* (COLING), 2012.

Huey Yee Lee and Hemnaath Renganathan. Chinese sentiment analysis using maximum entropy. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology* (SAAIP), page 89, 2011.

Yiou Lin, Hang Lei, Jia Wu, and Xiaoyu Li. An empirical study on sentiment classification of Chinese review using word embedding. *arXiv preprint arXiv:1511.01665*, 2015.

Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Pro-*

cessing and Computational Linguistics-Volume 1, pages 63–70. Association for Computational Linguistics, 2002.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Quoc V Le, and Ilya Stuskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013c.

Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

Saif M Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. Sentiment lexicons for Arabic social media. In *Proceedings of the International Conference on Language Resources and Evaluation* (LREC), pages 33–37, 2016.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. Semeval-2013 task 2: Sentiment Analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 312–320, 2013.

F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

R Řehřek and P Sojka. Gensim - Python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 2011.

Magnus Sahlgren. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*, 2005.

Magnus Sahlgren. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):31–51, 2008.

Mohammad Salameh, Saif M Mohammad, and Svetlana Kiritchenko. Sentiment after translation: A case-study on Arabic social media posts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (HLT-NAACL), 2015.

AP Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems* (NIPS), 2014.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*, 2017.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics, 2011.

Songbo Tan and Jin Zhang. An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications*, 34(4):2622–2629, 2008.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565. Association for Computational Linguistics, 2014.

Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28 (2):496–509, 2016.

Xuewei Tang and Xiaojun Wan. Learning bilingual embedding model for cross-language sentiment classification. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences*

on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02, pages 134–141. IEEE Computer Society, 2014.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior Research Methods*, 45(4):1191–1207, 2013.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems* (NIPS), pages 649–657, 2015.

Huiwei Zhou, Long Chen, Fulin Shi, and Degen Huang. Learning bilingual sentiment word embeddings for cross-language sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics* (ACL), pages 430–440, 2015.

Xinjie Zhou, Xianjun Wan, and Jianguo Xiao. Cross-lingual sentiment classification with bilingual document representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (ACL), pages 1403–1412, 2016.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. Bilingual word embeddings for phrase-based machine translation. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 1393–1398, 2013.