Enriching Word Embeddings with a Regressor Instead of Labeled Corpora

Mohamed Abdalla,[1,2] **Magnus Sahlgren,**[3] **Graeme Hirst**[1,2]
[1]Vector Institute, Toronto, Canada
[2]Department of Computer Science, University of Toronto
[3]Swedish Institute of Computer Science, RISE SICS, Stockholm
msa@cs.toronto.edu, magnus.sahlgren@ri.se, gh@cs.toronto.edu

Abstract

We propose a novel method for enriching word-embeddings without the need of a labeled corpus. Instead, we show that relying on a regressor – trained with a small lexicon to predict pseudo-labels – significantly improves performance over current techniques that rely on human-derived sentence-level labels for an entire corpora. Our approach enables enrichment for corpora that have no labels (such as Wikipedia). Exploring the utility of this general approach in both sentiment and non-sentiment-focused tasks, we show how enriching embeddings, for both Twitter and Wikipedia-based embeddings, provide notable improvements in performance for: binary sentiment classification, SemEval Tasks, embedding analogy task, and, document classification. Importantly, our approach is notably better and more generalizable than other state-of-the-art approaches for enriching both labeled and unlabeled corpora.

Introduction

Word embeddings (*i.e.,* word vectors, distributed representations) are dense numeric sequences that represent words and 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 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 additional information 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.

Incorporating additional information during embedding creation, termed "enrichment", is a well-studied area of research. In sentiment-analysis, there exist many different structures for combining the traditional context loss with a sentiment loss, with the aim of balancing the learning of sentiment and the learning of context (Tang et al. 2016; Lan et al. 2016; Ren et al. 2016; Faruqui et al. 2015). However, the vast majority of these techniques require the entire dataset to be labeled at the sentence level as either positive or negative. This requirement is limiting as many commonly used datasets (*e.g.,* news corpora and Wikipedia) do not come with sentiment labels.

In this work, we introduce a more effective and generalizable way of incorporating sentiment during the creation of word embeddings. Our approach, replacing sentence level labels with pseudo-labels predicted by a regressor, can be used: (i) to allow for the enrichment of text corpora that originally did not have labels, and (ii) in conjunction with a variety of different enrichment architectures.

For this work, we demonstrate the effect of our approach on Tang et al. (2016)'s architecture, although it can be applied on other architectures (Lan et al. 2016; Ren et al. 2016). Our approach improves performance on a diverse set of tasks and enables us to enrich unlabeled corpora, which we show has the same positive effects. We quantitatively evaluate our proposed techniques in comparison with previous ones on a variety of tasks, including both sentiment and non-sentiment tasks to see whether the increase in performance for sentiment-related tasks comes at a price of performance in other unrelated tasks.

Previous Work

Word Embeddings

Word embeddings are dense vector representations of words from a corpus. They range from low-rank approximations of co-occurrence matrices (*e.g.*, Sahlgren 2005; Bullinaria and Levy 2012; Pennington, Socher, and Manning 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, Alvarez-Melis, and Jaakkola 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. 2013b) that attempts to predict the context words given the current word.

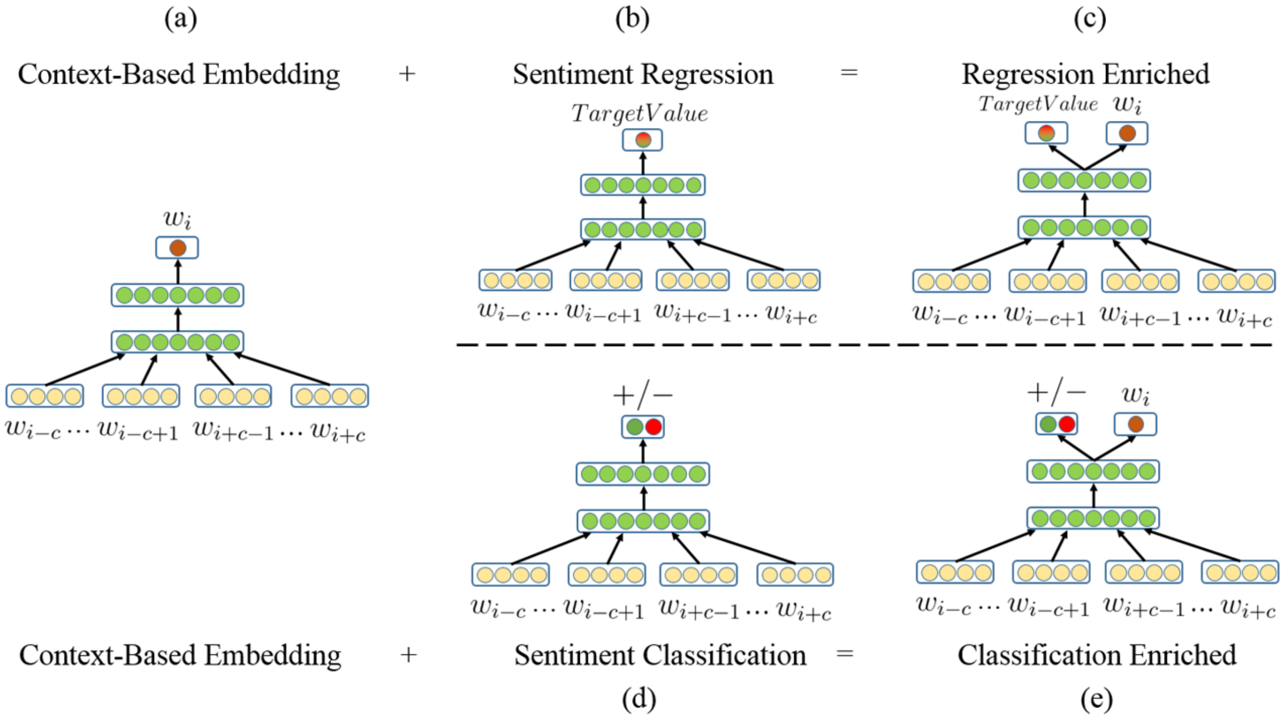
(a)
Context-Based Embedding
+
(b)
Sentiment Regression
=
(c)
Regression Enriched
$TargetValue$
$TargetValue$
w_i
w_i
$w_{i-c} \cdots w_{i-c+1}$ $w_{i+c-1} \cdots w_{i+c}$
$w_{i-c} \cdots w_{i-c+1}$ $w_{i+c-1} \cdots w_{i+c}$
$w_{i-c} \cdots w_{i-c+1}$ $w_{i+c-1} \cdots w_{i+c}$
$+/-$
$+/-$
w_i
$w_{i-c} \cdots w_{i-c+1}$ $w_{i+c-1} \cdots w_{i+c}$
$w_{i-c} \cdots w_{i-c+1}$ $w_{i+c-1} \cdots w_{i+c}$
Context-Based Embedding
+
Sentiment Classification
=
Classification Enriched
(d)
(e)

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.

Enriched Word Embeddings

Some past work has sought to improve quality or 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. 2016; Lan et al. 2016; Ren et al. 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 tests, retrofitting the word vectors using the paraphrase database (PPDB) (Ganitkevitch, Van Durme, and Callison-Burch 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 to learn the meaning and sentiment of phrases and sentences. The majority of enrichment approaches work by combining the traditional embedding model with an additional loss function, showing that the incorporated loss function serves to improve the capability of the embeddings to analyze sentiment (Tang et al. 2016; Lan et al. 2016; Ren et al. 2016). Our approach of having labels be predicted by a regressor can also be applied on such a class of algorithms, opening up enrichment for many different corpora.

Data

Affective Norms for English Words

Affective Norms for English Words (ANEW) is a representation of human emotions in a vector space with 3 underlying axes (Bradley and Lang 1999). The first axis, *valence*, ranges from unpleasant to pleasant; the second axis, *arousal*, ranges from calm to excited; the third axis, *dominance*, from in-control to out-of-control. We present an example of ANEW in Table 1. Warriner, Kuperman, and Brysbaert (2013) extended ANEW to 13,915 words from the original 1,000. We follow Abdalla and Hirst (2017) by replacing the need for using sentence-level sentiment labels to classify whether the current word and its context came from a positive or negative tweet by instead using automatically calculated sentiment values using this fine-grained extended ANEW.

Twitter Data

In order to emulate previous work for comparison, we follow the same procedure for the procurement and preprocessing of data. Following Tang et al. (2016), we scrape Twitter for positive and negative tweets, defined as those containing positive or negative emoticons respectively, as manual labeling of a large number of sentences from other sources

	Low stimulus	**High stimulus**
Arousal	*boring* (2.29)	*lust* (6.88)
Dominance	*rejection* (2.17)	*leader* (7.88)
Valence	*suicide* (1.25)	*triumphant* (8.82)

Table 1: Examples of words on differing locations on ANEW axes. Associated ANEW value for each axis is presented in parenthesis.

is not feasible. We scraped 5 million positive and 5 million negative tweets.

Wikipedia Data

To demonstrate that our method allows for the enrichment of unlabeled datasets, we also compared our proposed model against CBOW when trained on the more traditional Wikipedia dataset. The dataset gathered is a collection of all the English articles as of 2017-12-17.

Methods

Here we present the techniques used to learn sentiment-enriched embeddings. For consistency with previous work, 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. Where possible we used the parameters described by Tang et al.. 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.

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, in line 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.

Sentiment-Based Embeddings

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.. 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.

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 two approaches tested using ANEW: (i) valence regression, (ii) 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 approach, and 3 regressors that predicts each ANEW axis for the second 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 second approach) of the word. This predicted values will serve as the "gold" label during training of the enriched embedding.
3. *Approach (i)*: When training the embedding, 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 (ii): When training the embedding, we use the approach described above three times, one for each ANEW dimension. The entire sentiment loss, where loss is denoted by $\mathbb{L}$, is an equal split between MSE loss for each of the axes:

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

Enriching Context-Based Embeddings with Sentiment

Model names and details are defined in Table 2.

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:

$$\mathbb{L}_{\text{combined}} = \alpha\mathbb{L}_{\text{context}} + (1-\alpha)\mathbb{L}_{\text{classification}} \qquad (5)$$

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(e).

Hybrid Regression Models MODELS: *All variations of SE-HyReg*

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

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

As 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.

Experiments and Results

Twitter Experiments

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) Sentiment-related tasks: (i) binary sentiment classification, (ii) SemEval 2013 tweet classification, and (2) Non–sentiment-related tasks (downstream) tasks: (i) analogy evaluation, (ii) document classification.

In Table 2, we define the baseline and experimental network setups used for experimentation and assign each of them a name by which they will be referred to in later sections.

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. Like previous work, we trained classifiers to predict whether a word has positive or negative sentiment. The following lexicons were used: BL-Lexicon (Hu and Liu 2004)[1], MPQA (Wilson, Wiebe, and Hoffmann 2005)[2], and NRC-Lexicon (Mohammad and Turney 2013)[3].

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.

Our results mimic the trend observed by Tang et al. for the corresponding models. The absolute difference in performance between our results and theirs can 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 of 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. Tang et al.'s performs better than traditional CBOW, but when the non-linearity is removed (SE-HyPred-S) performance increases even more (across the board). Faruqui et al. (2015)'s CBOW+PPDB performs well on MPQA dataset but is outperformed by our novel approaches on the other datasets. We see that SE-HyPred-S and our two novel models (SE-HyReg-VWS and SE-HyReg-VADWS) both outperform all other models which serve as our baseline. Their performance is close to each other often within a range of 1–2; however, SE-HyReg-VWS and SE-HyReg-VADWS can be applied to any English corpus while SE-HyPred-S requires sentence-level labeling (which is not available for most corpora).

SemEval — Sentence-Level Sentiment Classification Having confirmed that enrichment improves performance when it comes to word-level sentiment, 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 test datasets provided by SemEval 2013. We must note that not all of the individual tweets in either of the sets could be obtained, as some of the

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

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

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

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-VWS	–	Valence regression on current context window + word
SE-HyReg-VADWS	–	Regression with all ANEW on current context window + word

Table 2: The models used in our experiment, and the names that we use for each. The first three are baselines, and other three are our experimental models. 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 $htanh$.

Embedding	10-fold CV		
	BL	MPQA	NRC
Word2Vec (CBOW)	68.4	66.5	65.0
CBOW+PPDB	76.1	**74.4**	65.8
SE-HyPred	75.1	70.4	66.6
SE-HyPred-S	**77.6**	73.2	**69.9**
SE-HyReg-VWS	75.7	74.1	68.7
SE-HyReg-VADWS	76.3	73.1	68.0

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

Embedding	Binary F1	Ternary F1
Word2Vec (CBOW)	69.2	54.2
CBOW+PPDB	65.6	52.4
SE-HyPred	**74.6**	52.8
SE-HyPred-S	73.3	55.3
SE-HyReg-VWS	72.5	54.2
SE-HyReg-VADWS	73.2	**55.8**

Table 4: Results for sentence-level SemEval tweet sentiment classification (for both the binary and ternary setups). We show the macro-F1 score. The best scores are bolded, and the second-best performing values are underlined.

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.

Here, SE-HyPred outperforms the newer models. SE-HyReg-VWS and SE-HyReg-VADWS outperforms both CBOW and Faruqui et al.'s CBOW+PDDB. In the ternary case the newer models outperform all of the baselines with our SE-HyReg-VADWS preforming the best. Once again removal of the non-linearity results in a significant increase in performance (SE-HyPred v.s. SE-HyPred-S). Here, CBOW+PPDB results in decreased performance from vanilla CBOW.

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 the effect of sentiment enrichment on non-sentiment tasks to see whether the enriched embeddings can be used for unrelated tasks.

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:

$$\text{arg max}_{\vec{v} \in V} cos(\vec{v}, \vec{v}_2 - \vec{v}_1 + \vec{v}_3) \tag{7}$$

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, Leśniak, and Czarnecki 2017), and the problem is further exacerbated by the fact that Twitter data itself does not discuss all of the topics and relationships represented in the dataset.

The trend observed between the new models (SE-HyPred-S, SE-HyReg-VWS, and SE-HyReg-VADWS) in relation to the baseline models also hold for this non-sentiment task. That is, the novel models greatly outperform all of the baselines for all of the three measures, even more so than the relative difference in sentiment tasks. The difference in relative performance for these three performing models is negligible in comparison to the difference to the baseline mod-

Embedding	P@1	P@5	P@10
Word2Vec (CBOW)	362 (2.15%)	918 (5.46%)	1243 (7.40%)
CBOW+PPDB	410 (2.44%)	1087 (6.47%)	1495 (8.89%)
SE-HyPred	204 (1.21%)	611 (3.64%)	888 (5.29%)
SE-HyPred-S	693 (4.12%)	1910 (11.37%)	2707 (16.11%)
SE-HyReg-VWS	641 (3.82%)	1868 (11.12%)	2462 (15.73%)
SE-HyReg-VADWS	**704 (4.19%)**	**2024 (12.05%)**	**2875 (17.11%)**

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 are bolded, and the second-best performing values are underlined.

els. Faruqui et al.'s CBOW+PPDB results in modest gains in performance over vanilla CBOW. Interestingly, SE-HyPred results in a decrease of performance (unlike in sentiment tasks). We believe this is caused by the non-linearity rather than the fact that the learning scheme is classification, as the other classification model SE-HyPred-S performs quite well.

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.

Once again, the trend observed between the new models (SE-HyPred-S, SE-HyReg-VWS, and SE-HyReg-VADWS) in relation to the baseline models also hold for this non-sentiment task. However, unlike the previous task the difference is not quite as pronounced. Here, SE-HyPred results in increased performance, whereas Faruqui et al.'s CBOW+PPDB results in decreased performance.

Embedding	F1 (Accuracy)
Word2Vec (CBOW)	21.1 (74.5)
CBOW+PPDB	19.6 (72.6)
SE-HyPred	41.8 (79.9)
SE-HyPred-S	**46.7 (81.9)**
SE-HyReg-VWS	46.5 (84.4)
SE-HyReg-VADWS	45.1 (84.2)

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

Wikipedia Experiments

Having demonstrated the competitive performance of regressor models (compared to models requiring human-generated labels), we sought to show the generalizability of our approach to corpora without labels (such as Wikipedia). We see that the performance trends are largely consistent (Tables 7–10). For this section, we were unable to compare to the works of Tang et al. (2016),Lan et al. (2016), and Ren et al. (2016) as all their approaches require labels assigned at the sentence level.

Word-Level Binary Sentiment Classification In this section, we study whether enriching the Wikipedia corpus results in performance improvement as was the case with the Twitter corpus. The methodology of this section is exactly the same as that of the Twitter experiment. Results are presented in Table 8, in which we see the general trend of improved performance repeat itself. Unlike the Twitter dataset, the improvement is not as pronounced, but this can be because of the very strong baseline. We see here that enriching only for Valence (SE-HyReg-VWS) results in improved performance than all three dimensions of ANEW (SE-HyReg-VADWS).

SemEval — Sentence-Level Sentiment Classification In this section, we tested the effect of enriching Wikipedia-sourced embeddings using our techniques on the SemEval test. As before, the experimental set-up is exactly that of the Twitter experiment.

The results, Table 9, for the binary case mimic the improvement seen in the Twitter experiments, although given the higher baseline there is a less dramatic improvement by the newer models. In the ternary classification case, however, enrichment seems to have no real effect. The best-performing model on the Wikipedia-trained embeddings performs worse than the best Twitter-trained embeddings and we think that the vocabulary of the respective datasets plays a large role in this. The embedding trained on the Twitter corpus is more likely to have the common misspellings and acronyms required for improved performance.

Embedding Analogy Evaluation We now look at how enriching the Wikipedia corpus affects performance on the embedding analogy task. The experimental methodology is exactly the same as that of Twitter experiment.

All embedding models trained on the Wikipedia Corpus perform better than those trained on the Twitter corpus. This

Embedding	P@1	P@5	P@10
Word2Vec (CBOW)	681 (3.48%)	**6119 (31.31%)**	**7432 (38.03%)**
SE-HyReg-VWS	**748 (3.83%)**	5869 (30.03%)	7170 (36.69%)
SE-HyReg-VADWS	732 (3.75%)	5756 (29.45%)	7042 (36.03%)

Table 7: Results from the Google analogy evaluation. Both the total count, and percentage of total examples in test-set are shown. The best scores are bolded, and the second-best performing values are underlined.

Embedding	10-fold CV		
	BL	MPQA	NRC
Word2Vec (CBOW)	73.41	73.88	62.27
SE-HyReg-VWS	**76.06**	**76.75**	**62.97**
SE-HyReg-VADWS	74.99	76.53	62.76

Table 8: The accuracies from the word-level binary sentiment classification task on the Wikipedia corpus. The best scores are bolded, and the second-best performing values are underlined.

Embedding	Binary F1	Ternary F1
Word2Vec (CBOW)	69.2	51.2
SE-HyReg-VWS	**73.2**	**51.5**
SE-HyReg-VADWS	69.8	51.1

Table 9: Results for sentence-level SemEval tweet sentiment classification (for both the binary and ternary setups). We show the macro-F1 score. The best scores are bolded, and the second-best performing values are underlined.

is expected given that the types of relationships tested by this task are not likely to be the topic of discussion on Twitter. Table 7 shows that enriching the model results in slightly better P@1 percision, but slightly lower on P@5 and P@10.

Document Classification Performance For this experiment, we used the same experimental setup described in the Twitter variant. Table 10 shows that enrichment here results in improved performance. Enriching for Valence alone seems to perform better than enriching for all ANEW dimensions.

Embedding	F1 (Accuracy)
Word2Vec (CBOW)	66.32 (91.4)
SE-HyReg-VWS	**72.51 (92.0)**
SE-HyReg-VADWS	66.47 (91.4)

Table 10: Results of the document classification task. Since the dataset is not balanced, both Macro-F1 and Accuracy are presented. The best scores are bolded, and the second-best performing values are underlined.

Conclusion

We have proposed a novel method to enrich word embeddings without the need for labeled corpora. Instead we show that using a regressor to predict pseudo-labels is just as effective an approach, and at the same time increases the "generalizability" of past approaches. Using our work we can modify previous approaches (Tang et al. 2016; Lan et al. 2016; Ren et al. 2016), those which rely on sentence level labels, to work on corpora without such labels. For this work we took Tang et al. (2016), and studied the effect of enrichment on both sentiment and non–sentiment-related tasks. We later applied our model to the Wikipedia corpus, something not possible with the current approaches — an application that is both novel and impactful. We show that there are improvements to be had both on sentiment and non-sentiment tasks by enriching with our algorithm.

We hypothesize that having sentiment framed as regression instead of classification (as is the case with the work of Tang et al. (2016)) would be more likely to achieve a sentiment gradient in the embedding space. This would also allow for the handling of neutral cases, which is not feasible or practical with previous techniques, yet is crucial for an actual model of human emotion, as not all documents have a sentiment value. However, we have not proven this claim and more work needs to be done to confirm our observations.

Not only does replacing the need for human-generated labels enable enrichment of other corpora, it also suggests that enrichment for other automatically generated information may be possible. Future work can study expanding prediction to other tasks.

Additionally, there is still much work to be done to study the effect of enrichment on the underlying word vector space. The reason for improved performance in the analogy task (even in non-sentiment categories, *e.g.,* plural) is not something we can yet explain. Our experiments on the Wikipedia dataset are a good step in this direction but more still needs to be done. We also hope to study whether the positive effect of enrichment holds with other methods of embedding creation.

More work is needed to determine whether such improvements would hold for classification problems with more than two classes, and what the limit would be before a drop-off in performance.

Acknowledgments

The work was financially supported by Vetenskåpsradet (the Swedish Research Council), and by the Natural Sciences and Engineering Research Council of Canada.

References

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

Bradley, M. M., and Lang, P. J. 1999. 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.

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

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

Faruqui, M.; Dodge, J.; Jauhar, S. K.; Dyer, C.; Hovy, E.; and Smith, N. A. 2015. 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*, 1606–1615. Association for Computational Linguistics.

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

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

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

Hashimoto, T.; Alvarez-Melis, D.; and Jaakkola, T. 2016. Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics* 4:273–286.

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

Jastrzebski, S.; Leśniak, D.; and Czarnecki, W. M. 2017. How to evaluate word embeddings? On importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv:1702.02170*.

Lan, M.; Zhang, Z.; Lu, Y.; and Wu, J. 2016. Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In *(IJCNN), 2016 International Joint Conference on Neural Networks*, 3172–3179.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 142–150. Association for Computational Linguistics.

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

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26*, 3111–3119. Curran Associates, Inc.

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

Nakov, P.; Rosenthal, S.; Kozareva, Z.; Stoyanov, V.; Ritter, A.; and Wilson, T. 2013. 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, 312–320.

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

Ren, Y.; Zhang, Y.; Zhang, M.; and Ji, D. 2016. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In *(AAAI), 30th AAAI conference on Artificial Intelligence*.

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

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

Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 151–161. Association for Computational Linguistics.

Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; and Zhou, M. 2016. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28(2):496–509.

Warriner, A. B.; Kuperman, V.; and Brysbaert, M. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior Research Methods* 45(4):1191–1207.

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