# Learning Word Vectors for Sentiment Analysis

**Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang,**
**Andrew Y. Ng,** and **Christopher Potts**
Stanford University
Stanford, CA 94305
[amaas, rdaly, ptpham, yuze, ang, cgpotts]@stanford.edu

## Abstract

Unsupervised vector-based approaches to semantics can model rich lexical meanings, but they largely fail to capture sentiment information that is central to many word meanings and important for a wide range of NLP tasks. We present a model that uses a mix of unsupervised and supervised techniques to learn word vectors capturing semantic term–document information as well as rich sentiment content. The proposed model can leverage both continuous and multi-dimensional sentiment information as well as non-sentiment annotations. We instantiate the model to utilize the document-level sentiment polarity annotations present in many online documents (e.g. star ratings). We evaluate the model using small, widely used sentiment and subjectivity corpora and find it out-performs several previously introduced methods for sentiment classification. We also introduce a large dataset of movie reviews to serve as a more robust benchmark for work in this area.

## 1 Introduction

Word representations are a critical component of many natural language processing systems. It is common to represent words as indices in a vocabulary, but this fails to capture the rich relational structure of the lexicon. Vector-based models do much better in this regard. They encode continuous similarities between words as distance or angle between word vectors in a high-dimensional space. The general approach has proven useful in tasks such as word sense disambiguation, named entity recognition, part of speech tagging, and document retrieval (Turney and Pantel, 2010; Collobert and Weston, 2008; Turian et al., 2010).

In this paper, we present a model to capture both semantic and sentiment similarities among words. The semantic component of our model learns word vectors via an unsupervised probabilistic model of documents. However, in keeping with linguistic and cognitive research arguing that expressive content and descriptive semantic content are distinct (Kaplan, 1999; Jay, 2000; Potts, 2007), we find that this basic model misses crucial sentiment information. For example, while it learns that *wonderful* and *amazing* are semantically close, it doesn't capture the fact that these are both very strong positive sentiment words, at the opposite end of the spectrum from *terrible* and *awful*.

Thus, we extend the model with a supervised sentiment component that is capable of embracing many social and attitudinal aspects of meaning (Wilson et al., 2004; Alm et al., 2005; Andreevskaia and Bergler, 2006; Pang and Lee, 2005; Goldberg and Zhu, 2006; Snyder and Barzilay, 2007). This component of the model uses the vector representation of words to predict the sentiment annotations on contexts in which the words appear. This causes words expressing similar sentiment to have similar vector representations. The full objective function of the model thus learns semantic vectors that are imbued with nuanced sentiment information. In our experiments, we show how the model can leverage document-level sentiment annotations of a sort that are abundant online in the form of consumer reviews for movies, products, etc. The technique is suffi-

ciently general to work also with continuous and multi-dimensional notions of sentiment as well as non-sentiment annotations (e.g., political affiliation, speaker commitment).

After presenting the model in detail, we provide illustrative examples of the vectors it learns, and then we systematically evaluate the approach on document-level and sentence-level classification tasks. Our experiments involve the small, widely used sentiment and subjectivity corpora of Pang and Lee (2004), which permits us to make comparisons with a number of related approaches and published results. We also show that this dataset contains many correlations between examples in the training and testing sets. This leads us to evaluate on, and make publicly available, a large dataset of informal movie reviews from the Internet Movie Database (IMDB).

## 2 Related work

The model we present in the next section draws inspiration from prior work on both probabilistic topic modeling and vector-spaced models for word meanings.

Latent Dirichlet Allocation (LDA; (Blei et al., 2003)) is a probabilistic document model that assumes each document is a mixture of latent topics. For each latent topic $T$, the model learns a conditional distribution $p(w|T)$ for the probability that word $w$ occurs in $T$. One can obtain a $k$-dimensional vector representation of words by first training a $k$-topic model and then filling the matrix with the $p(w|T)$ values (normalized to unit length). The result is a word–topic matrix in which the rows are taken to represent word meanings. However, because the emphasis in LDA is on modeling topics, not word meanings, there is no guarantee that the row (word) vectors are sensible as points in a $k$-dimensional space. Indeed, we show in section 4 that using LDA in this way does not deliver robust word vectors. The semantic component of our model shares its probabilistic foundation with LDA, but is factored in a manner designed to discover word vectors rather than latent topics. Some recent work introduces extensions of LDA to capture sentiment in addition to topical information (Li et al., 2010; Lin and He, 2009; Boyd-Graber and Resnik, 2010). Like LDA, these methods focus on modeling sentiment-imbued topics rather than embedding words in a vector space.

Vector space models (VSMs) seek to model words directly (Turney and Pantel, 2010). Latent Semantic Analysis (LSA), perhaps the best known VSM, explicitly learns semantic word vectors by applying singular value decomposition (SVD) to factor a term–document co-occurrence matrix. It is typical to weight and normalize the matrix values prior to SVD. To obtain a $k$-dimensional representation for a given word, only the entries corresponding to the $k$ largest singular values are taken from the word's basis in the factored matrix. Such matrix factorization-based approaches are extremely successful in practice, but they force the researcher to make a number of design choices (weighting, normalization, dimensionality reduction algorithm) with little theoretical guidance to suggest which to prefer.

Using term frequency (tf) and inverse document frequency (idf) weighting to transform the values in a VSM often increases the performance of retrieval and categorization systems. Delta idf weighting (Martineau and Finin, 2009) is a supervised variant of idf weighting in which the idf calculation is done for each document class and then one value is subtracted from the other. Martineau and Finin present evidence that this weighting helps with sentiment classification, and Paltoglou and Thelwall (2010) systematically explore a number of weighting schemes in the context of sentiment analysis. The success of delta idf weighting in previous work suggests that incorporating sentiment information into VSM values via supervised methods is helpful for sentiment analysis. We adopt this insight, but we are able to incorporate it directly into our model's objective function. (Section 4 compares our approach with a representative sample of such weighting schemes.)

## 3 Our Model

To capture semantic similarities among words, we derive a probabilistic model of documents which learns word representations. This component does not require labeled data, and shares its foundation with probabilistic topic models such as LDA. The sentiment component of our model uses sentiment annotations to constrain words expressing similar

sentiment to have similar representations. We can efficiently learn parameters for the joint objective function using alternating maximization.

### 3.1 Capturing Semantic Similarities

We build a probabilistic model of a document using a continuous mixture distribution over words indexed by a multi-dimensional random variable $\theta$. We assume words in a document are conditionally independent given the mixture variable $\theta$. We assign a probability to a document $d$ using a joint distribution over the document and $\theta$. The model assumes each word $w_i \in d$ is conditionally independent of the other words given $\theta$. The probability of a document is thus

$$p(d) = \int p(d, \theta) d\theta = \int p(\theta) \prod_{i=1}^{N} p(w_i|\theta) d\theta. \quad (1)$$

Where $N$ is the number of words in $d$ and $w_i$ is the $i^{th}$ word in $d$. We use a Gaussian prior on $\theta$.

We define the conditional distribution $p(w_i|\theta)$ using a log-linear model with parameters $R$ and $b$. The energy function uses a word representation matrix $R \in \mathbb{R}^{(\beta \times |V|)}$ where each word $w$ (represented as a one-on vector) in the vocabulary $V$ has a $\beta$-dimensional vector representation $\phi_w = Rw$ corresponding to that word's column in $R$. The random variable $\theta$ is also a $\beta$-dimensional vector, $\theta \in \mathbb{R}^{\beta}$ which weights each of the $\beta$ dimensions of words' representation vectors. We additionally introduce a bias $b_w$ for each word to capture differences in overall word frequencies. The energy assigned to a word $w$ given these model parameters is

$$E(w; \theta, \phi_w, b_w) = -\theta^T \phi_w - b_w. \quad (2)$$

To obtain the distribution $p(w|\theta)$ we use a softmax,

$$p(w|\theta; R, b) = \frac{\exp(-E(w; \theta, \phi_w, b_w))}{\sum_{w' \in V} \exp(-E(w'; \theta, \phi_{w'}, b_{w'}))} \quad (3)$$

$$= \frac{\exp(\theta^T \phi_w + b_w)}{\sum_{w' \in V} \exp(\theta^T \phi_{w'} + b_{w'})}. \quad (4)$$

The number of terms in the denominator's summation grows linearly in $|V|$, making exact computation of the distribution possible. For a given $\theta$, a word $w$'s occurrence probability is related to how closely its representation vector $\phi_w$ matches the scaling direction of $\theta$. This idea is similar to the word vector inner product used in the log-bilinear language model of Mnih and Hinton (2007).

Equation 1 resembles the probabilistic model of LDA (Blei et al., 2003), which models documents as mixtures of latent topics. One could view the entries of a word vector $\phi$ as that word's association strength with respect to each latent topic dimension. The random variable $\theta$ then defines a weighting over topics. However, our model does not attempt to model individual topics, but instead directly models word probabilities conditioned on the topic mixture variable $\theta$. Because of the log-linear formulation of the conditional distribution, $\theta$ is a vector in $\mathbb{R}^{\beta}$ and not restricted to the unit simplex as it is in LDA.

We now derive maximum likelihood learning for this model when given a set of unlabeled documents $D$. In maximum likelihood learning we maximize the probability of the observed data given the model parameters. We assume documents $d_k \in D$ are i.i.d. samples. Thus the learning problem becomes

$$\max_{R,b} p(D; R, b) = \prod_{d_k \in D} \int p(\theta) \prod_{i=1}^{N_k} p(w_i|\theta; R, b) d\theta. \quad (5)$$

Using *maximum a posteriori* (MAP) estimates for $\theta$, we approximate this learning problem as

$$\max_{R,b} \prod_{d_k \in D} p(\hat{\theta}_k) \prod_{i=1}^{N_k} p(w_i|\hat{\theta}_k; R, b), \quad (6)$$

where $\hat{\theta}_k$ denotes the MAP estimate of $\theta$ for $d_k$. We introduce a Frobenious norm regularization term for the word representation matrix $R$. The word biases $b$ are not regularized reflecting the fact that we want the biases to capture whatever overall word frequency statistics are present in the data. By taking the logarithm and simplifying we obtain the final objective,

$$\nu ||R||_F^2 + \sum_{d_k \in D} \lambda ||\hat{\theta}_k||_2^2 + \sum_{i=1}^{N_k} \log p(w_i|\hat{\theta}_k; R, b), \quad (7)$$

which is maximized with respect to $R$ and $b$. The hyper-parameters in the model are the regularization

weights ($\lambda$ and $\nu$), and the word vector dimensionality $\beta$.

### 3.2 Capturing Word Sentiment

The model presented so far does not explicitly capture sentiment information. Applying this algorithm to documents will produce representations where words that occur together in documents have similar representations. However, this unsupervised approach has no explicit way of capturing which words are predictive of sentiment as opposed to content-related. Much previous work in natural language processing achieves better representations by learning from multiple tasks (Collobert and Weston, 2008; Finkel and Manning, 2009). Following this theme we introduce a second task to utilize labeled documents to improve our model's word representations.

Sentiment is a complex, multi-dimensional concept. Depending on which aspects of sentiment we wish to capture, we can give some body of text a sentiment label $s$ which can be categorical, continuous, or multi-dimensional. To leverage such labels, we introduce an objective that the word vectors of our model should predict the sentiment label using some appropriate predictor,

$$\hat{s} = f(\phi_w). \tag{8}$$

Using an appropriate predictor function $f(x)$ we map a word vector $\phi_w$ to a predicted sentiment label $\hat{s}$. We can then improve our word vector $\phi_w$ to better predict the sentiment labels of contexts in which that word occurs.

For simplicity we consider the case where the sentiment label $s$ is a scalar continuous value representing sentiment polarity of a document. This captures the case of many online reviews where documents are associated with a label on a star rating scale. We linearly map such star values to the interval $s \in [0, 1]$ and treat them as a probability of positive sentiment polarity. Using this formulation, we employ a logistic regression as our predictor $f(x)$. We use $w$'s vector representation $\phi_w$ and regression weights $\psi$ to express this as

$$p(s = 1|w; R, \psi) = \sigma(\psi^T \phi_w + b_c), \tag{9}$$

where $\sigma(x)$ is the logistic function and $\psi \in \mathbb{R}^\beta$ is the logistic regression weight vector. We additionally introduce a scalar bias $b_c$ for the classifier.

The logistic regression weights $\psi$ and $b_c$ define a linear hyperplane in the word vector space where a word vector's positive sentiment probability depends on where it lies with respect to this hyperplane. Learning over a collection of documents results in words residing different distances from this hyperplane based on the average polarity of documents in which the words occur.

Given a set of labeled documents $D$ where $s_k$ is the sentiment label for document $d_k$, we wish to maximize the probability of document labels given the documents. We assume documents in the collection and words within a document are i.i.d. samples. By maximizing the log-objective we obtain,

$$\max_{R,\psi,b_c} \sum_{k=1}^{|D|} \sum_{i=1}^{N_k} \log p(s_k|w_i; R, \psi, b_c). \tag{10}$$

The conditional probability $p(s_k|w_i; R, \psi, b_c)$ is easily obtained from equation 9.

### 3.3 Learning

The full learning objective maximizes a sum of the two objectives presented. This produces a final objective function of,

$$\begin{aligned} &\nu||R||_F^2 + \sum_{k=1}^{|D|} \lambda||\hat{\theta}_k||_2^2 + \sum_{i=1}^{N_k} \log p(w_i|\hat{\theta}_k; R, b) \\ &+ \sum_{k=1}^{|D|} \frac{1}{|S_k|} \sum_{i=1}^{N_k} \log p(s_k|w_i; R, \psi, b_c). \end{aligned} \tag{11}$$

$|S_k|$ denotes the number of documents in the dataset with the same rounded value of $s_k$ (i.e. $s_k < 0.5$ and $s_k \geq 0.5$). We introduce the weighting $\frac{1}{|S_k|}$ to combat the well-known imbalance in ratings present in review collections. This weighting prevents the overall distribution of document ratings from affecting the estimate of document ratings in which a particular word occurs. The hyper-parameters of the model are the regularization weights ($\lambda$ and $\nu$), and the word vector dimensionality $\beta$.

Maximizing the objective function with respect to $R$, $b$, $\psi$, and $b_c$ is a non-convex problem. We use alternating maximization, which first optimizes the

word representations ($R$, $b$, $\psi$, and $b_c$) while leaving the MAP estimates ($\hat{\theta}$) fixed. Then we find the new MAP estimate for each document while leaving the word representations fixed, and continue this process until convergence. The optimization algorithm quickly finds a global solution for each $\hat{\theta}_k$ because we have a low-dimensional, convex problem in each $\hat{\theta}_k$. Because the MAP estimation problems for different documents are independent, we can solve them on separate machines in parallel. This facilitates scaling the model to document collections with hundreds of thousands of documents.

## 4 Experiments

We evaluate our model with document-level and sentence-level categorization tasks in the domain of online movie reviews. For document categorization, we compare our method to previously published results on a standard dataset, and introduce a new dataset for the task. In both tasks we compare our model's word representations with several bag of words weighting methods, and alternative approaches to word vector induction.

### 4.1 Word Representation Learning

We induce word representations with our model using 25,000 movie reviews from IMDB. Because some movies receive substantially more reviews than others, we limited ourselves to including at most 30 reviews from any movie in the collection. We build a fixed dictionary of the 5,000 most frequent tokens, but ignore the 50 most frequent terms from the original full vocabulary. Traditional stop word removal was not used because certain stop words (e.g. negating words) are indicative of sentiment. Stemming was not applied because the model learns similar representations for words of the same stem when the data suggests it. Additionally, because certain non-word tokens (e.g. "!" and ":-)" ) are indicative of sentiment, we allow them in our vocabulary. Ratings on IMDB are given as star values ($\in \{1, 2, ..., 10\}$), which we linearly map to $[0, 1]$ to use as document labels when training our model.

The semantic component of our model does not require document labels. We train a variant of our model which uses 50,000 unlabeled reviews in addition to the labeled set of 25,000 reviews. The unlabeled set of reviews contains neutral reviews as well as those which are polarized as found in the labeled set. Training the model with additional unlabeled data captures a common scenario where the amount of labeled data is small relative to the amount of unlabeled data available. For all word vector models, we use 50-dimensional vectors.

As a qualitative assessment of word representations, we visualize the words most similar to a query word using vector similarity of the learned representations. Given a query word $w$ and another word $w'$ we obtain their vector representations $\phi_w$ and $\phi_{w'}$, and evaluate their cosine similarity as $S(\phi_w, \phi_{w'}) = \frac{\phi_w^T \phi_{w'}}{||\phi_w|| \cdot ||\phi_{w'}||}$. By assessing the similarity of $w$ with all other words $w'$, we can find the words deemed most similar by the model.

Table 1 shows the most similar words to given query words using our model's word representations as well as those of LSA. All of these vectors capture broad semantic similarities. However, both versions of our model seem to do better than LSA in avoiding accidental distributional similarities (e.g., *screwball* and *grant* as similar to *romantic*) A comparison of the two versions of our model also begins to highlight the importance of adding sentiment information. In general, words indicative of sentiment tend to have high similarity with words of the same sentiment polarity, so even the purely unsupervised model's results look promising. However, they also show more genre and content effects. For example, the sentiment enriched vectors for *ghastly* are truly semantic alternatives to that word, whereas the vectors without sentiment also contain some content words that tend to have *ghastly* predicated of them. Of course, this is only an impressionistic analysis of a few cases, but it is helpful in understanding why the sentiment-enriched model proves superior at the sentiment classification results we report next.

### 4.2 Other Word Representations

For comparison, we implemented several alternative vector space models that are conceptually similar to our own, as discussed in section 2:

**Latent Semantic Analysis (LSA; Deerwester et al., 1990)** We apply truncated SVD to a tf.idf weighted, cosine normalized count matrix, which is a standard weighting and smoothing scheme for

| | Our model Sentiment + Semantic | Our model Semantic only | LSA |
|---|---|---|---|
| **melancholy** | bittersweet | thoughtful | poetic |
| | heartbreaking | warmth | lyrical |
| | happiness | layer | poetry |
| | tenderness | gentle | profound |
| | compassionate | loneliness | vivid |
| **ghastly** | embarrassingly | predators | hideous |
| | trite | hideous | inept |
| | laughably | tube | severely |
| | atrocious | baffled | grotesque |
| | appalling | smack | unsuspecting |
| **lackluster** | lame | passable | uninspired |
| | laughable | unconvincing | flat |
| | unimaginative | amateurish | bland |
| | uninspired | clichéd | forgettable |
| | awful | insipid | mediocre |
| **romantic** | romance | romance | romance |
| | love | charming | screwball |
| | sweet | delightful | grant |
| | beautiful | sweet | comedies |
| | relationship | chemistry | comedy |

Table 1: Similarity of learned word vectors. Each target word is given with its five most similar words using cosine similarity of the vectors determined by each model. The full version of our model (left) captures both lexical similarity as well as similarity of sentiment strength and orientation. Our unsupervised semantic component (center) and LSA (right) capture semantic relations.

VSM induction (Turney and Pantel, 2010).

**Latent Dirichlet Allocation (LDA; Blei et al., 2003)** We use the method described in section 2 for inducing word representations from the topic matrix. To train the 50-topic LDA model we use code released by Blei et al. (2003). We use the same 5,000 term vocabulary for LDA as is used for training word vector models. We leave the LDA hyperparameters at their default values, though some work suggests optimizing over priors for LDA is important (Wallach et al., 2009).

**Weighting Variants** We evaluate both binary (b) term frequency weighting with smoothed delta idf ($\Delta$t') and no idf (n) because these variants worked well in previous experiments in sentiment (Martineau and Finin, 2009; Pang et al., 2002). In all cases, we use cosine normalization (c). Paltoglou and Thelwall (2010) perform an extensive analysis of such weighting variants for sentiment tasks.

### 4.3 Document Polarity Classification

Our first evaluation task is document-level sentiment polarity classification. A classifier must predict whether a given review is positive or negative given the review text.

Given a document's bag of words vector $v$, we obtain features from our model using a matrix-vector product $Rv$, where $v$ can have arbitrary tf.idf weighting. We do not cosine normalize $v$, instead applying cosine normalization to the final feature vector $Rv$. This procedure is also used to obtain features from the LDA and LSA word vectors. In preliminary experiments, we found 'bnn' weighting to work best for $v$ when generating document features via the product $Rv$. In all experiments, we use this weighting to get multi-word representations

| Features | PL04 | Our Dataset | Subjectivity |
|---|---|---|---|
| Bag of Words (bnc) | 85.45 | 87.80 | 87.77 |
| Bag of Words (b$\Delta$t'c) | 85.80 | 88.23 | 85.65 |
| LDA | 66.70 | 67.42 | 66.65 |
| LSA | 84.55 | 83.96 | 82.82 |
| Our Semantic Only | 87.10 | 87.30 | 86.65 |
| Our Full | 84.65 | 87.44 | 86.19 |
| Our Full, Additional Unlabeled | 87.05 | 87.99 | 87.22 |
| Our Semantic + Bag of Words (bnc) | 88.30 | 88.28 | 88.58 |
| Our Full + Bag of Words (bnc) | 87.85 | 88.33 | 88.45 |
| Our Full, Add'l Unlabeled + Bag of Words (bnc) | 88.90 | 88.89 | 88.13 |
| Bag of Words SVM (Pang and Lee, 2004) | 87.15 | N/A | 90.00 |
| Contextual Valence Shifters (Kennedy and Inkpen, 2006) | 86.20 | N/A | N/A |
| tf.$\Delta$idf Weighting (Martineau and Finin, 2009) | 88.10 | N/A | N/A |
| Appraisal Taxonomy (Whitelaw et al., 2005) | 90.20 | N/A | N/A |

Table 2: Classification accuracy on three tasks. From left to right the datasets are: A collection of 2,000 movie reviews often used as a benchmark of sentiment classification (Pang and Lee, 2004), 50,000 reviews we gathered from IMDB, and the sentence subjectivity dataset also released by (Pang and Lee, 2004). All tasks are balanced two-class problems.

from word vectors.

#### 4.3.1 Pang and Lee Movie Review Dataset

The polarity dataset version 2.0 introduced by Pang and Lee (2004)[1] consists of 2,000 movie reviews, where each is associated with a binary sentiment polarity label. We report 10-fold cross validation results using the authors' published folds to make our results comparable with others in the literature. We use a linear support vector machine (SVM) classifier trained with LIBLINEAR (Fan et al., 2008), and set the SVM regularization parameter to the same value used by Pang and Lee (2004).

Table 2 shows the classification performance of our method, other VSMs we implemented, and previously reported results from the literature. Bag of words vectors are denoted by their weighting notation. Features from word vector learner are denoted by the learner name. As a control, we trained versions of our model with only the unsupervised semantic component, and the full model (semantic and sentiment). We also include results for a version of our full model trained with 50,000 additional unlabeled examples. Finally, to test whether our models' representations complement a standard bag of words, we evaluate performance of the two feature representations concatenated.

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data

Our method's features clearly outperform those of other VSMs, and perform best when combined with the original bag of words representation. The variant of our model trained with additional unlabeled data performed best, suggesting the model can effectively utilize large amounts of unlabeled data along with labeled examples. Our method performs competitively with previously reported results in spite of our restriction to a vocabulary of only 5,000 words.

We extracted the movie title associated with each review and found that 1,299 of the 2,000 reviews in the dataset have at least one other review of the same movie in the dataset. Of 406 movies with multiple reviews, 249 have the same polarity label for all of their reviews. Overall, these facts suggest that, relative to the size of the dataset, there are highly correlated examples with correlated labels. This is a natural and expected property of this kind of document collection, but it can have a substantial impact on performance in datasets of this scale. In the random folds distributed by the authors, approximately 50% of reviews in each validation fold's test set have a review of the same movie with the same label in the training set. Because the dataset is small, a learner may perform well by memorizing the association between label and words unique to a particular movie (e.g., character names or plot terms).

We introduce a substantially larger dataset, which

uses disjoint sets of movies for training and testing. These steps minimize the ability of a learner to rely on idiosyncratic word–class associations, thereby focusing attention on genuine sentiment features.

#### 4.3.2 IMDB Review Dataset

We constructed a collection of 50,000 reviews from IMDB, allowing no more than 30 reviews per movie. The constructed dataset contains an even number of positive and negative reviews, so randomly guessing yields 50% accuracy. Following previous work on polarity classification, we consider only highly polarized reviews. A negative review has a score $\leq 4$ out of 10, and a positive review has a score $\geq 7$ out of 10. Neutral reviews are not included in the dataset. In the interest of providing a benchmark for future work in this area, we release this dataset to the public.[2]

We evenly divided the dataset into training and test sets. The training set is the same 25,000 labeled reviews used to induce word vectors with our model. We evaluate classifier performance after cross-validating classifier parameters on the training set, again using a linear SVM in all cases. Table 2 shows classification performance on our subset of IMDB reviews. Our model showed superior performance to other approaches, and performed best when concatenated with bag of words representation. Again the variant of our model which utilized extra unlabeled data during training performed best.

Differences in accuracy are small, but, because our test set contains 25,000 examples, the variance of the performance estimate is quite low. For example, an accuracy increase of 0.1% corresponds to correctly classifying an additional 25 reviews.

### 4.4 Subjectivity Detection

As a second evaluation task, we performed sentence-level subjectivity classification. In this task, a classifier is trained to decide whether a given sentence is subjective, expressing the writer's opinions, or objective, expressing purely facts. We used the dataset of Pang and Lee (2004), which contains subjective sentences from movie review summaries and objective sentences from movie plot summaries. This task is substantially different from the review classification task because it uses sentences as opposed to entire documents and the target concept is subjectivity instead of opinion polarity. We randomly split the 10,000 examples into 10 folds and report 10-fold cross validation accuracy using the SVM training protocol of Pang and Lee (2004).

Table 2 shows classification accuracies from the sentence subjectivity experiment. Our model again provided superior features when compared against other VSMs. Improvement over the bag-of-words baseline is obtained by concatenating the two feature vectors.

## 5 Discussion

We presented a vector space model that learns word representations captuing semantic and sentiment information. The model's probabilistic foundation gives a theoretically justified technique for word vector induction as an alternative to the overwhelming number of matrix factorization-based techniques commonly used. Our model is parametrized as a log-bilinear model following recent success in using similar techniques for language models (Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2007), and it is related to probabilistic latent topic models (Blei et al., 2003; Steyvers and Griffiths, 2006). We parametrize the topical component of our model in a manner that aims to capture word representations instead of latent topics. In our experiments, our method performed better than LDA, which models latent topics directly.

We extended the unsupervised model to incorporate sentiment information and showed how this extended model can leverage the abundance of sentiment-labeled texts available online to yield word representations that capture both sentiment and semantic relations. We demonstrated the utility of such representations on two tasks of sentiment classification, using existing datasets as well as a larger one that we release for future research. These tasks involve relatively simple sentiment information, but the model is highly flexible in this regard; it can be used to characterize a wide variety of annotations, and thus is broadly applicable in the growing areas of sentiment analysis and retrieval.

[2] Dataset and further details are available online at: `http://www.andrew-maas.net/data/sentiment`

## References


C. O. Alm, D. Roth, and R. Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of HLT/EMNLP*, pages 579–586.

A. Andreevskaia and S. Bergler. 2006. Mining WordNet for fuzzy sentiment: sentiment tag extraction from WordNet glosses. In *Proceedings of the European ACL*, pages 209–216.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. a neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, August.

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, May.

J. Boyd-Graber and P. Resnik. 2010. Holistic sentiment analysis across languages: multilingual supervised latent Dirichlet allocation. In *Proceedings of EMNLP*, pages 45–55.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing. In *Proceedings of the ICML*, pages 160–167.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, September.

R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. 2008. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, August.

J. R. Finkel and C. D. Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of NAACL*, pages 326–334.

A. B. Goldberg and J. Zhu. 2006. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *TextGraphs: HLT/NAACL Workshop on Graph-based Algorithms for Natural Language Processing*, pages 45–52.

T. Jay. 2000. *Why We Curse: A Neuro-Psycho-Social Theory of Speech*. John Benjamins, Philadelphia/Amsterdam.

D. Kaplan. 1999. What is meaning? Explorations in the theory of *Meaning as Use*. Brief version — draft 1. Ms., UCLA.

A. Kennedy and D. Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22:110–125, May.

F. Li, M. Huang, and X. Zhu. 2010. Sentiment analysis with global topics and local dependency. In *Proceedings of AAAI*, pages 1371–1376.

C. Lin and Y. He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pages 375–384.

J. Martineau and T. Finin. 2009. Delta tfidf: an improved feature space for sentiment analysis. In *Proceedings of the 3rd AAAI International Conference on Weblogs and Social Media*, pages 258–261.

A. Mnih and G. E. Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the ICML*, pages 641–648.

G. Paltoglou and M. Thelwall. 2010. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the ACL*, pages 1386–1395.

B. Pang and L. Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278.

B. Pang and L. Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

C. Potts. 2007. The expressive dimension. *Theoretical Linguistics*, 33:165–197.

B. Snyder and R. Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL)*, pages 300–307.

M. Steyvers and T. L. Griffiths. 2006. Probabilistic topic models. In T. Landauer, D McNamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the ACL*, page 384394.

P. D. Turney and P. Pantel. 2010. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

H. Wallach, D. Mimno, and A. McCallum. 2009. Rethinking LDA: why priors matter. In *Proceedings of NIPS*, pages 1973–1981.

C. Whitelaw, N. Garg, and S. Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of CIKM*, pages 625–631.

T. Wilson, J. Wiebe, and R. Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of AAAI*, pages 761–769.