# Bootstrap Domain-Specific Sentiment Classifiers from Unlabeled Corpora

**Andrius Mudinas, Dell Zhang, and Mark Levene**
Department of Computer Science and Information Systems
Birkbeck, University of London
London WC1E 7HX, UK
andrius@dcs.bbk.ac.uk, dell.z@ieee.org, mark@dcs.bbk.ac.uk

## Abstract

There is often the need to perform sentiment classification in a particular domain where no labeled document is available. Although we could make use of a general-purpose off-the-shelf sentiment classifier or a pre-built one for a different domain, the effectiveness would be inferior. In this paper, we explore the possibility of building domain-specific sentiment classifiers w ith u nlabeled d ocuments o nly. Our investigation indicates that in the word embeddings learned from the unlabeled corpus of a given domain, the distributed word representations (vectors) for opposite sentiments form distinct clusters, though those clusters are not transferable across domains. Exploiting such a clustering structure, we are able to utilize machine learning algorithms to induce a quality domain-specific sentiment lexicon from just a few typical sentiment words ("seeds"). An important finding is that simple linear model based supervised learning algorithms (such as linear SVM) can actually work better than more sophisticated semi-supervised/transductive learning algorithms which represent the state-of-the-art technique for sentiment lexicon induction. The induced lexicon could be applied directly in a lexicon-based method for sentiment classification, but a higher performance could be achieved through a two-phase bootstrapping method which uses the induced lexicon to assign positive/negative sentiment scores to unlabeled documents first, a nd t hen u ses those documents found to have clear sentiment signals as pseudo-labeled examples to train a document sentiment classifier v ia supervised learning algorithms (such as LSTM). On several benchmark datasets for document sentiment classification, our end-to-end pipelined approach which is overall unsupervised (except for a tiny set of seed words) outperforms existing unsupervised approaches and achieves an accuracy comparable to that of fully supervised approaches.

## 1 Introduction

Sentiment analysis (Liu, 2015) is a popular research topic which has a wide range of applications, such as summarizing customer reviews, monitoring social media, and predicting stock market trends (Bollen et al., 2011). A basic task in sentiment analysis is to classify the sentiment polarity of a given piece of text (document), i.e., whether the opinion expressed in the text is positive or negative (Pang et al., 2002), which is the focus of this paper.

There are many different approaches to sentiment classification in the Natural Language Processing (NLP) literature — from simple lexicon-based methods (Ding et al., 2008; Thelwall et al., 2010; Thelwall et al., 2012) to learning-based approaches (Pang and Lee, 2004; Turney, 2002; Jo and Oh, 2011; Argamon et al., 2007; Lin and He, 2009), and also hybrid methods in between (Mudinas et al., 2012; Zhang et al., 2011). No matter which approach is taken, a sentiment classifier built for its target domain would work well only within that specific domain, but suffer a serious performance loss once the domain boundary is crossed. The same word could drastically change its sentiment polarity (and/or strength) if it is used in a different domain. For example, being "small" is likely to be negative



Transactions of the Association for Computational Linguistics, vol. 6, pp. 269–285, 2018. Action Editor: Diana McCarthy.
Submission batch: 11/2017; Revision batch: 2/2018; Published 5/2018.

for a hotel room but positive for a digital camcorder, being "unexpected" may be a good thing for the ending of a movie but not for the engine of a car, and we will probably enjoy "interesting" books but not necessarily "interesting" food. Here, the domain could be defined not by the topic of the documents but by the style of writing. For example, the meanings of words like "gay" and "terrific" would depend on whether the text was written in a historical era or modern times.

When we need to perform sentiment classification in a new domain unseen before, there are usually neither labeled dictionary available to employ lexicon-based sentiment classifiers nor labeled corpus available to train learning-based sentiment classifiers. It is, of course, possible to resort to a general-purpose off-the-shelf sentiment classifier, or a pre-built one for a different domain. However, the effectiveness would often be unsatisfactory because of the reasons mentioned above. There have been some studies on domain adaptation or transfer learning for sentiment classification (Blitzer et al., 2007; Tan et al., 2009; Pan et al., 2010; Glorot et al., 2011; Yoshida et al., 2011; Bollegala et al., 2013; Xia et al., 2013; Yang and Eisenstein, 2015), but they still require a large amount of labeled training data from a fairly *similar* source domain, which is not always feasible. Those algorithms also tend to be computational-expensive and time-consuming (Mohammad and Turney, 2010; Fast et al., 2016).

In this paper, we propose an end-to-end pipelined *nearly-unsupervised* approach to *domain-specific* sentiment classification of documents for a new domain based on distributed word representations (vectors). As shown in Fig. 1, the proposed approach consists of three main stages (components):

(1) domain-specific sentiment word embedding,
(2) domain-specific sentiment lexicon induction,
(3) domain-specific sentiment classification of documents.

Briefly speaking, given a large unlabeled corpus for a new domain, we would first set up the vector space for that domain via word embedding, then induce a sentiment lexicon in the discovered vector space from a very small set of seed words as well as a general-purpose lexicon, and finally exploit the induced lexicon in a lexicon-based document sentiment classifier to bootstrap a more effective learning-based document sentiment classifier for that domain. The second stage of our approach outperforms the state-of-the-art unsupervised method for sentiment lexicon induction (Hamilton et al., 2016), which is the most closely related work (see Section 2). The key to the superior performance of our method compared with theirs is the insight gained from our first stage that positive and negative sentiment words are largely clustered in the domain-specific vector space but these two clusters have a non-negligible overlap, therefore semi-supervised/transductive learning algorithms could be easily misled by the examples in the overlap and would actually not work as well as simple supervised classification algorithms. Overall, the document sentiment classifier resulting from our nearly-unsupervised approach does not require any labeled document to be trained, and it can outperform the state-of-the-art unsupervised method for document sentiment classification (Eisenstein, 2017). The source code for our implemented system and the datasets for our experiments are open to the research community[1].

The rest of this paper is organized as follows. In Section 2, we review previous studies on this topic. In Sections 3 to 5, we describe the three main stages of our approach respectively. In Section 6, we draw conclusions and discuss future work.

## 2 Related Work

Most of the early sentiment analysis systems took lexicon-based approaches to document sentiment classification which rely on pre-compiled sentiment lexicons (Owsley et al., 2006). Various methods have been proposed to automatically produce such sentiment lexicons (Hu and Liu, 2004; Ding et al., 2008). Later, the focus of research shifted to learning-based approaches (Pang et al., 2002; Pang and Lee, 2004), as supervised learning algorithms usually deliver a much higher accuracy in sentiment classification than pure lexicon-based methods. However, lexicons have not completely lost their attractiveness: they are usually easier to understand and to maintain by non-experts, and they can also be integrated into learning-based sentiment classifiers (Mudinas et al., 2012; Eisenstein, 2017).

[1] https://goo.gl/8K9PbE

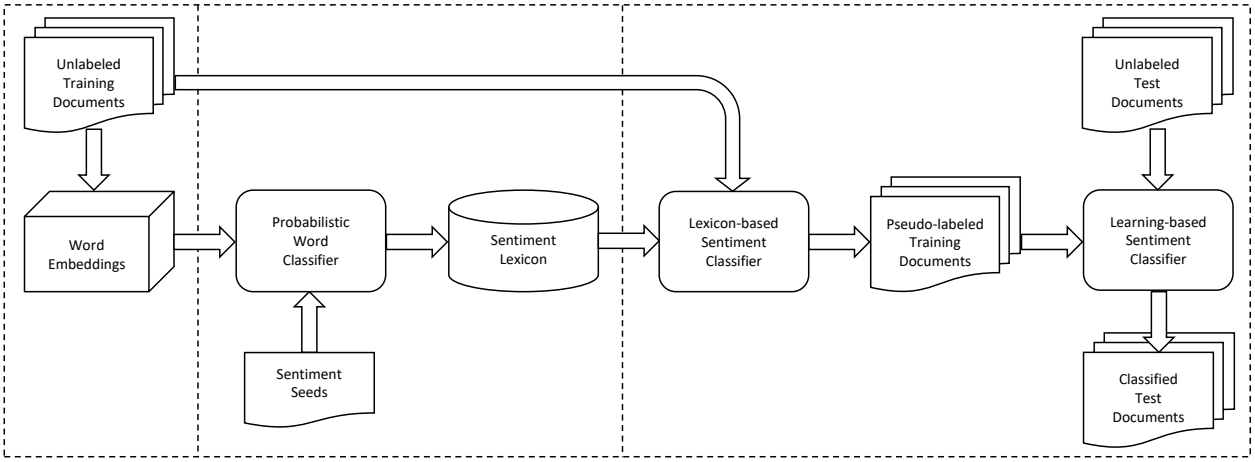



Figure 1: Our *nearly-unsupervised* approach to *domain-specific* sentiment classification.

The lexicon-based sentiment classifier used in our experiments is a publicly-available system called pSenti[2] (Mudinas et al., 2012). In addition to a customizable sentiment lexicon, it also uses shallow NLP techniques like part-of-speech (POS) tagging and the detection of sentiment inverters and other modifiers (intensifying and diminishing adverbs).

The introduction of modern word embedding techniques like word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) have opened the possibility of new sentiment analysis methods. Given a large unlabeled corpus, such techniques can learn from word co-occurrence information and produce a vector space of hundreds of dimensions, with each word being assigned a corresponding vector. The resulting vector space helps in understanding the semantic relationships between words and allows grouping of words based on their linguistic similarities. Recently Rothe et al. (2016) proposed the DENSIFIER method that can reduce the dimensionality of word embeddings without losing semantic information and explored its application in various domains. For the SemEval-2015 task (Rosenthal et al., 2015), DENSIFIER performed slightly worse compared to word2vec, though its training time was shorter by a factor of 21. In fact, previous studies such as (Rothe et al., 2016; Cliche, 2017) suggest that word2vec usually provides the best word embeddings for sentiment analysis tasks.

In their recent work, Hamilton et al. (2016) demonstrated that by starting from a small set of seed words and conducting label propagation over the lexical graph derived from the pairwise proximities of word embeddings, they could induce a domain-specific sentiment lexicon comparable to a hand-curated one. Intuitively, the success of their method named SentProp requires a relatively clear separation between sentiment words of opposite polarity in the vector space which, as we will show later, is not very realistic. Moreover, they have focused on the induction of sentiment lexicons alone, while we are trying to design an end-to-end pipeline that can turn unlabeled documents in a new domain directly to their sentiment classifications, with domain-specific sentiment lexicon induction as a key component.

Recent advances in *deep learning* (LeCun et al., 2015) has elevated sentiment analysis to new performance levels (Kim, 2014; Dai and Le, 2015; Hong and Fang, 2015). As reported by Dai and Le (2015), the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) Recurrent Neural Network (RNN) can reach or surpass the performance levels of all previous baselines for sentiment classification of documents. One of the many appeals of LTSM is that it can connect previous information to the current context and allow seamless integration of pre-trained word embeddings as the first (projection) layer of the neural network. Moreover, Radford et al. (2017) discovered the "sentiment unit", the single unit which can learn the perfect representation of sentiment, in a multiplicative LSTM with

[2] https://goo.gl/pj4XAQ

4096 units, despite the fact that the LSTM was only trained for a completely different purpose — to predict the next character in the text of Amazon reviews. Our results are in line with those findings and confirmed the superiority of LSTM in building document-level sentiment classifiers.

Zhang et al. (2011) tried to address the low recall problem of lexicon-based methods for Twitter sentiment classification via training a learning-based sentiment classifier using the noisy labels generated by a lexicon-based sentiment classifier (Ding et al., 2008). Although the basic idea of their work is similar to what we do in the third stage of our approach (see Section 5), there exist several notable differences. First, they adopted a single general-purpose sentiment lexicon provided by Ding et al. (2008) and used it for all domains, while we would induce a different lexicon for each different domain. Consequently, their method could have a relatively large variance in the document sentiment classification performance because of the domain mismatch (e.g., $F_1 = 0.874$ for the "Tangled" tweets and $F_1 = 0.647$ for the "Obama" tweets), whereas our approach would perform quite consistently over different domains. Second, they would need to strip out all the previously-known opinion words in their single general-purpose sentiment lexicon from the training documents in order to prevent the *training bias* and force their document sentiment classifier to exploit domain-specific features, but doing this would obviously lose the very valuable sentiment signals carried by those opinion words. In contrast, we would be able to utilize all terms in the training documents, including those opinion words that appeared in our automatically induced domain-specific lexicons, as features, when building our document sentiment classifiers. Third, they designed their method specifically for Twitter sentiment classification, while our approach would work for not only short texts such as tweets (see Section 5.2) but also long texts such as customer reviews (see Section 5.1). Fourth, they had to use an intermediate step to identify additional opinionated tweets (according to the opinion indicators extracted through the $\chi^2$ test on the results of their lexicon-based sentiment classifier) in order to handle the neutral class, but we would not require that time-consuming step as we would use the calibrated probabilistic outputs of our document sentiment classifier to detect the neutral class (see Section 5.3).

## 3 Domain-Specific Sentiment Word Embedding

Our approach to domain-specific document-level sentiment classification is built on top of word embeddings — distributed word representations (vectors) that could be learned from an unlabeled corpus to encode the semantic similarities between words (Goldberg, 2017).

In this section, we investigate how the embeddings of sentiment words for a particular domain would look like in the domain-specific vector space. To ensure a fair comparison with the state-of-the-art sentiment lexicon induction technique SentProp[3] (Hamilton et al., 2016) later in Section 4, we adopt the same publicly-available *pre-trained* word embeddings for the following three domains together with the corresponding sets of sentiment words (i.e., sentiment lexicons).

- Standard-English. We use the the Google News word embeddings[4] and the 'General Inquirer' lexicon (Stone et al., 1966) with the sentiment polarity scores collected by Warriner et al. (2013).
- Twitter. We use the word embeddings constructed by Rothe et al. (2016) and the sentiment lexicon from the SemEval-2015 Task 10E (Rosenthal et al., 2015).
- Finance. We use the word embeddings learned using an SVD-based method (Manning et al., 2008) from a collection of "8-K" financial reports[5] (Lee et al., 2014) and the finance sentiment lexicon hand-crafted by Hamilton et al. (2016).

Note that the above three sentiment lexicons would be used for both the inspection of sentiment word distributions in this section and the evaluation of sentiment lexicon induction later in the next section. Furthermore, to facilitate a fair comparison with the state-of-the-art unsupervised document sentiment classification technique ProbLex-DCM[6] (Eisenstein, 2017) later in Section 5, we also adopt the following two document collections which they have used.

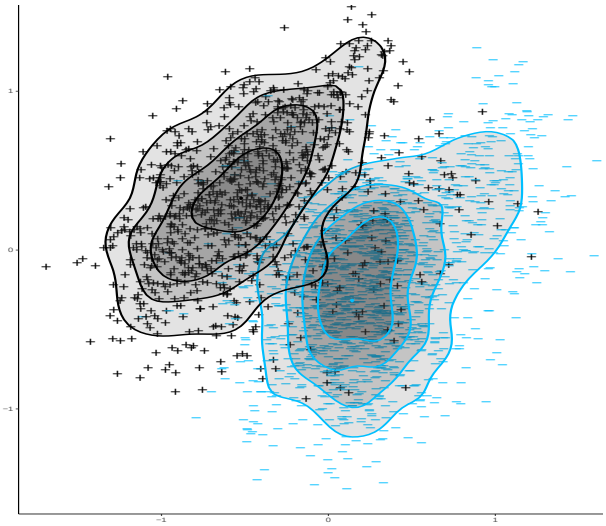
[3] https://goo.gl/BFkY8N
[4] https://goo.gl/5r79l6
[5] https://goo.gl/7ntr2V
[6] https://goo.gl/Qr993F

- IMDB. We use 50k movie reviews in English from IMDB (Maas et al., 2011) with 25k labeled training documents.
- Amazon. We use about 28k product reviews in English across four product categories from Amazon (Blitzer et al., 2007; McAuley and Leskovec, 2013) with 8k labeled training documents.

The word embeddings for the above two domains were *trained by us* on the respective corpora using word2vec (Mikolov et al., 2013) which employs a two-layer neural network and is by far the most widely used word embedding technique. Specifically, we ran word2vec with skip-gram of a five-word window to construct word vectors of 500 dimensions, as recommended by previous studies[7]. The sentiment lexicon made by Liu (2015) is consistently one of the best for analyzing reviews (Ribeiro et al., 2016), so it is used for both of those domains.

Drawing an analogy to the well-known *cluster hypothesis* in Information Retrieval (IR) (Manning et al., 2008), here we put forward the cluster hypothesis for sentiment analysis: words in the same cluster behave similarly with respect to sentiment polarity in a specific domain. That is to say, we expect positive and negative sentiment words to form distinct clusters, given that they have been represented in an appropriate vector space. To verify this hypothesis, it would be useful to visualize the high-dimensional sentiment word vectors in a 2D plane. We have tried a number of dimensionality reduction techniques including the $t$-distributed Stochastic Neighbor Embedding ($t$-SNE) (van der Maaten and Hinton, 2008), but found that simply using the classic Principle Component Analysis (PCA) (Bishop, 2006) works very well for this purpose.

We have found that in general, the above cluster hypothesis holds for word embeddings within a specific domain. Fig. 2a shows that in the Standard-English domain, the sentiment words with opposite polarities would form two distinct clusters. However, it can also be seen that those two clusters would overlap with each other. That is because each word carries not only a sentiment value but also its linguistic and semantic information. Zooming into one of the word vector space regions (Fig. 2b) can help us understand why sentiment words with different polarities could be grouped together: 'hail', 'stormy' and 'sunny' are linguistically similar as they all describe weather conditions, yet they convey very different sentiment values. Moreover, as described by (Plutchik, 1984), sentiment could be grouped into multiple dimensions such as joy–sadness, anger–fear, trust–disgust and anticipation–surprise. Putting that aside, certain sentiment words can be classified sometimes as positive and sometimes as negative, depending on the context. These reasons lead to the phenomenon that many sentiment words are located in the overlapping noisy region between two clusters in the domain-specific vector space.



(a) The global vector space showing two clusters.

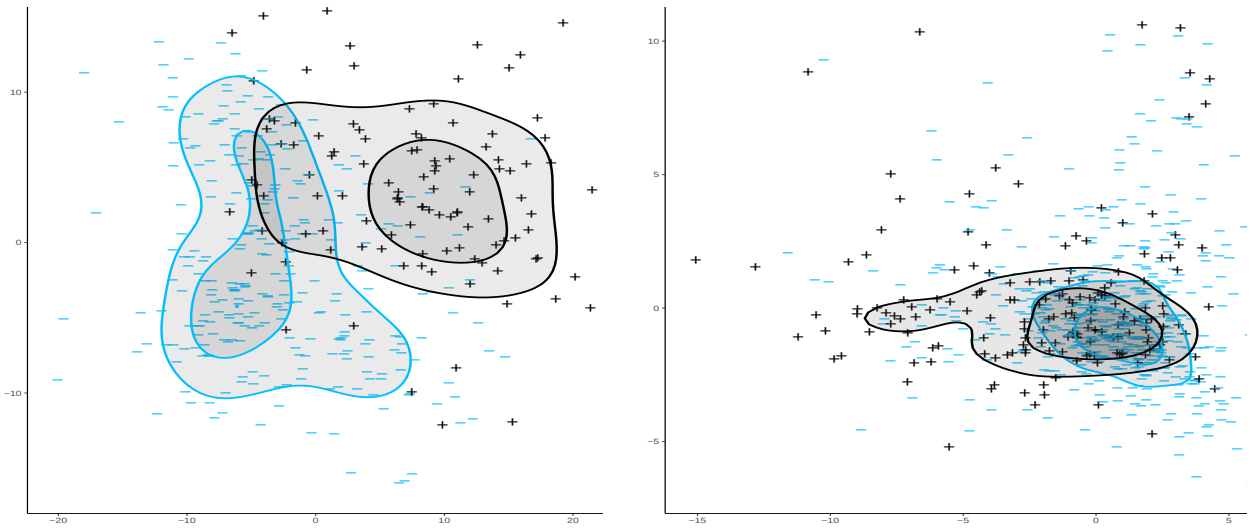

(b) A local region of the vector space zoomed in.

Figure 2: Visualisation of the sentiment words in the Standard-English domain.

On visual inspection of the Finance (Fig. 3a) sentiment words and IMDB (Fig. 4a) sentiment words in their respective vector spaces, we can see that positive and negative words form distinct clusters which are largely separable. However, if we consider Finance sentiment words in the IMDB vector space (see Fig. 3b), positive and negative words would be mixed together and could not be separated easily.

One may be surprised that positive and negative sentiment words form their respective clusters, because most of the time they could be used in ex-

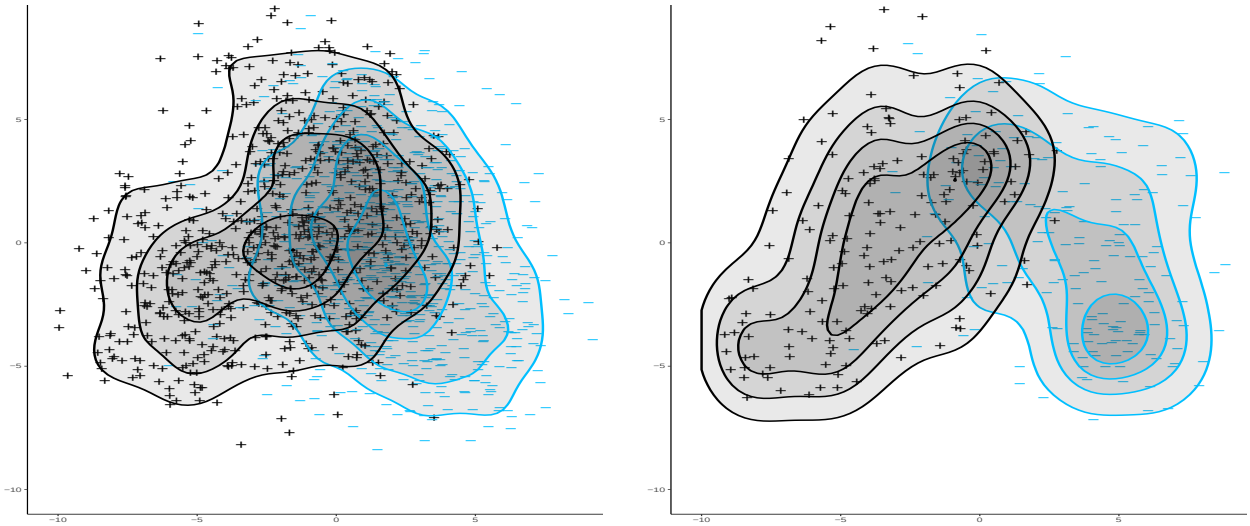
[7] https://goo.gl/SyAdej



(a) In the Finance (same domain) vector space.

(b) In the IMDB (different domain) vector space.

Figure 3: Sentiment words of Finance in the same/different domain vector space.



(a) Original/Full.

(b) Filtered.

Figure 4: Sentiment words about movies in the IMDB vector space before/after filtering.

actly the same context which might suggest that they would result in similar word embeddings. For example, we could say "the room is good" and also "the room is bad": both are legitimate sentences. The probable reason for the cluster hypothesis to be true is that in reality people tend to use positive sentiment words together much more often than to mix them with negative sentiment words, and vice versa. For example, it would be much more often for us to see sentences like "the room is clean and tidy" than "the room is clean but messy". It is a long established fact in computational linguistics that words with similar meanings tend to occur nearby each other (Miller and Charles, 1991); sentiment words are no exception (Turney, 2002). Moreover, it has been widely observed that online customer reviews are affected by the so-called love-hate self-selection bias: users tend to rate only products which they either like or hate, leading to a lot more 1-star and 5-star ratings than other (moderate) ratings; if the product is just

average or so-so, they probably will not bother to leave reviews. The polarization of online customer reviews would also encourage the clustering of sentiment words into opposite polarities.

## 4 Domain-Specific Sentiment Lexicon Induction

Given the word embeddings for a specific domain, we can induce a customized sentiment lexicon from a few typical sentiment words ("seeds") frequently used in that particular domain. Such an induced domain-specific sentiment lexicon plays a crucial role in the pipeline towards domain-specific document-level sentiment classification.

Table 1 shows the seed words for five different domains which are *identical* to those used by Hamilton et al. (2016) except for the two additional domains IMDB and Amazon. The induction of a sentiment lexicon could then be formulated as a simple *word sentiment classification* problem with two classes (positive vs. negative). Each word is represented as a vector via domain-specific word embedding; the seed words are labeled with their corresponding classes while all the other words (i.e., "candidates") are unlabeled; the task here is to learn a classifier from the labeled examples first and then apply it to predict the sentiment polarity of each unlabeled candidate word. The probabilistic outputs of such a word sentiment classifier could be regarded as the measure of confidence about the predicted sentiment polarity. In the end, those candidate words with a high probability of being either positive or negative would be added to the sentiment lexicon. The final induced sentiment lexicon would include both the seed words and the selected candidate words.

As pointed out by Mudinas et al. (2012), if we simply consider all words from the given corpus as candidate words, the above described word sentiment classifier tends to assign sentiment values not only to the actual sentiment words but also to their associated product features or more generally the *aspects* of the expressed view. For example, if a lot of customers do not like the weight of a product, the word sentiment classifier may assign strong negative sentiment to "weight", yet this is not stable — the sentiment polarity of "weight" may be different when a new version of the product is released or the customer population has changed, and furthermore it probably does not apply to other products. To avoid this potential issue, it would be necessary to consider only a high-quality list of candidate words which are likely to be genuine sentiment words. Such a list of candidate words could be obtained directly from general-purpose sentiment lexicons. It is also possible to perform NLP on the target domain corpus and extract frequently-occurring adjectives or other typical sentiment indicators like emoticons as candidate words, which is beyond the scope of this paper.

To examine the effectiveness of different machine learning algorithms for building such domain-specific word sentiment classifiers, we attempt to recreate known sentiment lexicons in three domains: Standard-English, Twitter, and Finance (see Section 3), in the same way as Hamilton et al. (2016) did. Put differently, for the purpose of evaluation, we would just use a known sentiment lexicon in the corresponding domain as the list of candidate words and see how different machine learning algorithms would classify those candidate words based on their domain-specific word embeddings. For those lexicons with ternary sentiment classification (positive vs. neutral vs. negative), the class-mass normalization method (Zhu et al., 2003) used by Hamilton et al. (2016) has been applied here to identify the neutral category. The quality of each induced lexicon for a specific domain is evaluated by comparing it with its corresponding known lexicon as the ground-truth, according to the performance metrics which are the same as in (Hamilton et al., 2016): Area Under the Receiver-Operating-Characteristic (ROC) Curve ($AUC$) for the binary classifications (ignoring the neutral class, as is common in previous work) and Kendall's $\tau$ rank correlation coefficient with continuous human-annotated polarity scores. Note that Kendall's $\tau$ is not suitable for the Finance domain, as its known sentiment lexicon is only binary. Therefore, our experimental setting and performance measures are all identical to those of Hamilton et al. (2016), which ensures the validity of the empirical comparison between our approach and theirs.

In Table 2, we compare a number of typical supervised and semi-supervised/transductive learning algorithms for *word sentiment classification* in the context of domain-specific sentiment lexicon induc-

| Corpus | Positive | Negative |
|---|---|---|
| Standard-English | good, lovely, excellent, fortunate, pleasant, delightful, perfect, loved, love, happy | bad, horrible, poor, unfortunate, unpleasant, disgusting, evil, hated, hate, unhappy |
| Twitter | love, loved, loves, awesome, nice, amazing, best, fantastic, correct, happy | hate, hated, hates, terrible, nasty, awful, worst, horrible, wrong, sad |
| Finance | successful, excellent, profit, beneficial, improving, improved, success, gains, positive | negligent, loss, volatile, wrong, losses, damages, bad, litigation, failure, down, negative |
| IMDB | good, excellent, perfect, happy, interesting, amazing, unforgettable, genius, gifted, incredible | bad, bland, horrible, disgusting, poor, banal, shallow, disappointed, disappointing, lifeless, simplistic, bore |
| Amazon | IMDB domain seeds (as above) plus positive, fortunate, correct, nice | IMDB domain seeds (as above) plus negative, unfortunate, wrong, terrible, inferior |

Table 1: The "seeds" for domain-specific sentiment lexicon induction.

tion:

- $k$NN — $k$ Nearest Neighbors (Hastie et al., 2009),
- LR — Logistic Regression (Hastie et al., 2009),
- SVM$_{lin}$ — Support Vector Machine with the linear kernel (Joachims, 1998),
- SVM$_{rbf}$ — Support Vector Machine with the non-linear RBF kernel (Joachims, 1998),
- TSVM — Transductive Support Vector Machine (Joachims, 1999),
- S3VM — Semi-Supervised Support Vector Machine (Gieseke et al., 2012),
- CPLE — Contrastive Pessimistic Likelihood Estimation (Loog, 2016),
- SGT — Spectral Graph Transducer (Joachims, 2003),
- SentProp — a label propagation based classification method proposed for the SocialSent system (Hamilton et al., 2016).

The suitable parameter values of the above learning algorithms (such as the $C$ for SVM) are found via grid search with cross-validation, and the probabilistic outputs are given by Platt scaling (Platt, 2000) if they are not provided by the original learning algorithm.

The experimental results shown in Table 2 demonstrate that in almost every single domain, simple linear model based supervised learning algorithms (LR and SVM$_{lin}$) can achieve the optimal or near-optimal accuracy for the sentiment lexicon induction task, and they outperform the state-of-the-art sentiment lexicon induction method SentProp (Hamilton et al., 2016) by a large margin. The performance improvements are statistically significant ($p$-value $< 0.05$) according to the sign test. There does not seem to be any benefit of utilizing non-linear models ($k$NN and SVM$_{rbf}$) or semi-supervised/transductive learning algorithms (TSVM, S3VM, CPLE, SGT, and SentProp). The qualitative analysis of the sentiment lexicons induced by different methods shows that they differ only on those borderline, ambiguous words (such as "soft") residing in the noisy overlapping region between two clusters in the vector space (see Section 3). In particular, SentProp is based on label propagation over the lexical graph of words, so it could be easily misled by noisy borderline words when sentiment clusters have considerable overlap with each other, kind of "over-fitting" (Bishop, 2006). Furthermore, according to our experiments on the same machine, those simple linear models are 70+ times faster than SentProp. The speed difference is mainly due to the fact that supervised learning algorithms only need to train on a small number of labeled words ("seeds" in our context) while semi-supervised/transductive learning algorithms need to train on not only a small number of labeled words but also a large number of unlabeled words.

It has also been observed in our experiments that there is a typical *precision/recall trade-off* (Manning et al., 2008) for the automatic induction of semantic lexicons. Assuming that the classified candidate words are added to the lexicon in the descending order of their probabilities (of being either positive or negative), the induced lexicon will be noisier and noisier when it becomes bigger and bigger.

| | Corpus | Supervised | | | | Semi-Supervised/Transductive | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $k$NN | LR | SVM$_{lin}$ | SVM$_{rbf}$ | TSVM | S3VM | CPLE | SGT | SentProp |
| $AUC$ | Standard-English | 0.892 | 0.931 | 0.939 | **0.941** | 0.901 | 0.540 | 0.680 | 0.852 | 0.906 |
| | Twitter | 0.849 | **0.900** | 0.895 | 0.895 | 0.770 | 0.521 | 0.651 | 0.725 | 0.860 |
| | Finance | 0.711 | **0.944** | 0.942 | 0.932 | 0.665 | 0.561 | 0.836 | 0.725 | 0.916 |
| $\tau$ | Standard-English | 0.469 | 0.495 | **0.498** | 0.495 | 0.487 | 0.038 | 0.162 | 0.409 | 0.440 |
| | Twitter | 0.490 | **0.569** | 0.548 | 0.547 | 0.522 | 0.001 | 0.211 | 0.437 | 0.500 |

Table 2: Comparing the induced lexicons with their corresponding known lexicons (ground-truth) according to the ranking of sentiment words measured by $AUC$ and Kendall's $\tau$.

Fig. 5 shows that imposing a higher cut-off probability threshold (for candidate words to enter the induced lexicon) would decrease the size of the induced lexicon but increase its quality (accuracy). On one hand, the induced lexicon needs to contain a sufficient number of sentiment words, especially when detecting sentiment from short texts, as a lexicon-based method cannot reasonably classify documents with none or too few sentiment words. On the other hand, the noise (misclassified sentiment words) in the induced lexicon would obviously have a detrimental impact on the accuracy of the document sentiment classifier built on top of it. Contrary to most previous work like that from Qiu et al. (2011) which tries to expand the sentiment lexicon as much as possible and thus maintain a high recall, we would put more emphasis on the precision and keep a tight control of the lexicon size. For us, having a small sentiment lexicon is affordable, because our proposed approach to document sentiment classification will be able to mitigate the low recall problem of lexicon-based methods by combining them with learning-based methods, which we shall talk about next.

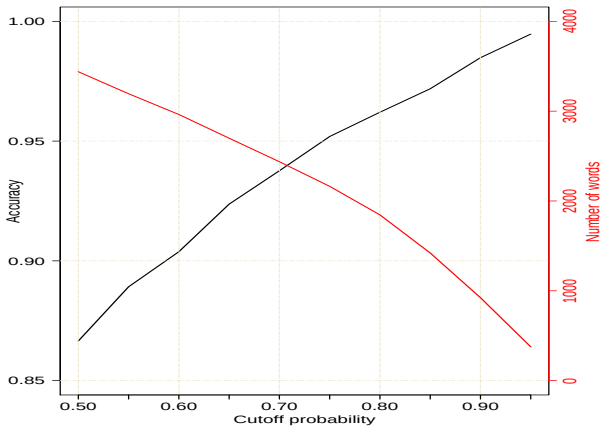



Figure 5: How the accuracy and size of an induced lexicon are influenced by the cut-off probability threshold.

## 5 Domain-Specific Sentiment Classification of Documents

A domain-specific sentiment lexicon, automatically induced using the above technique, provides a solid basis for building domain-specific document sentiment classifiers. For the experiments here, we would use a list of 7866 candidate words constructed by merging two well-known general-purpose sentiment lexicons that are both publicly available — the 'General Inquirer' (Stone et al., 1966) and the sentiment lexicon from Liu (2012). This set of candidate words is itself a combined, general-purpose sentiment lexicon, so we name it the GI+BL lexicon. Moreover, we would set the cut-off probability threshold to a generally good value 0.7 in our sentiment lexicon induction algorithm. Comparing the IMDB vector space including all the candidate words (Fig. 4a) with that including only the high-probability candidate words (Fig. 4b), it is obvious that the positive and negative sentiment clusters become more clearly separated in the latter.

The induced sentiment lexicon on its own could be applied directly in a lexicon-based method for sentiment classification of documents, and a reasonably good performance could be achieved as we will show later in Table 4. However, most of the time, lexicon-based sentiment classifiers are not as effective as learning-based sentiment classifiers. One reason is that the former tends to suffer from a poor recall. For example, with a limited size sentiment lexicon, lexicon-based methods would often fail to

detect the sentiment present in short texts, e.g., from Twitter, due to the lexical gap.

Given the induced sentiment lexicon, we propose to use a lexicon-based sentiment classifier to classify unlabeled documents, and then use those classified documents containing at least three sentiment words as *pseudo-labeled* documents to be used later for the training of a learning-based sentiment classifier. The condition of "at least three sentiment words" is to ensure that only reliably classified documents would be further utilised as training examples.

### 5.1 Sentiment Classification of Long Texts

First, we try the induced sentiment lexicons in the lexicon-based sentiment classifier pSenti (Mudinas et al., 2012) to see how good they are. Given a sentiment lexicon, pSenti is able to perform not only binary sentiment classification but also ordinal sentiment classification on a five-point scale. To measure the binary classification performance, we use both micro-averaged $F_1$ (mi$F_1$) and macro-averaged $F_1$ (ma$F_1$) which are commonly used in text categorization (Yang and Liu, 1999). To measure the five-point scale classification performance, we use both Cohen's $\kappa$ coefficient (Manning et al., 2008) and also Root-Mean-Square Error ($RMSE$) (Bishop, 2006). As the baseline, we use a combined general-purpose sentiment lexicon, GI+BL, mentioned previously in Section 4. As we can see from the results shown in Table 3, using the induced sentiment lexicon for the target domain would make the lexicon-based sentiment classifier pSenti perform better than simply employing an existing general-purpose sentiment lexicon. Moreover, using the sentiment lexicons induced from the same domain would lead a much better performance than using the sentiment lexicons induced from a different domain.

Second, to evaluate the proposed two-phase bootstrapping method, we make empirical comparisons on the IMDB and Amazon datasets using a number of representative methods for *document sentiment classification*:

- pSenti — a concept-level lexicon-based sentiment classifier (Mudinas et al., 2012),
- ProbLex-DCM — a probabilistic lexicon-based classification using the Dirichlet Compound Multinomial (DCM) likelihood to reduce effective counts for repeated words (Eisenstein, 2017),
- SVM$_{lin}$ — Support Vector Machine with linear kernel (Joachims, 1998),
- CNN — Convolutional Neural Network (Kim, 2014),
- LSTM — Long Short-Term Memory, a Recurrent Neural Network (RNN) that can remember values over arbitrary time intervals (Hochreiter and Schmidhuber, 1997; Dai and Le, 2015).

To apply the deep learning algorithms CNN and LSTM that have a word embedding projection layer, we fix the review size to 500 words, truncating reviews longer than that and padding reviews shorter than that with null values. As pointed out by Greff et al. (2017), the hidden layer size is an important hyperparameter of LSTM: usually the larger the network, the better the performance but the longer the training time. In our experiments, we have used an LSTM network with 400 units on the hidden layer which is the capacity that a PC with one Nvidia GTX 1080 Ti GPU can afford and a dropout (Wager et al., 2013) rate of 0.5 which is the most common setting in research literature (Srivastava et al., 2014; Hong and Fang, 2015; Cliche, 2017).

As shown in Table 4, the above described two-phase bootstrapping method has been demonstrated to be beneficial: the learning-based sentiment classifiers trained on pseudo-labeled data are superior to lexicon-based sentiment classifiers, including the state-of-the-art unsupervised sentiment classifier ProbLex-DCM (Eisenstein, 2017). Furthermore, the two-phase bootstrapping method is a general framework which can utilize any lexicon-based sentiment classifier to produce pseudo-labeled data. Therefore the more sophisticated ProbLex-DCM could also be used instead of pSenti in this framework, which is likely to deliver an even higher performance. Among the three learning-based sentiment classifiers, LSTM achieved the best performance on both datasets, which is consistent with the observations in other studies like Dai and Le (2015).

Comparing the LSTM-based sentiment classifiers trained on pseudo-labeled and real labeled data, we can also see that using a large number of pseudo-labeled examples could achieve a similar effect as using $25/4 \approx$ 6k and $8/2 =$ 4k real labeled examples for IMDB and Amazon respectively. This suggests that the unsupervised approach is actually preferable to the supervised approach if there are

| Lexicon | | binary | | | | 5-point scale | |
|---|---|---|---|---|---|---|---|
| | | mi$F_1$ | ma$F_1$ | $F_1^{\text{pos}}$ | $F_1^{\text{neg}}$ | Cohen's $\kappa$ | $RMSE$ |
| general-purpose | GI+BL | 0.745 | 0.744 | 0.764 | 0.722 | 0.235 | 1.325 |
| domain-specific | same domain (Kitchen) | **0.761** | **0.761** | **0.772** | **0.750** | **0.236** | **1.310** |
| | different domain (Electronics) | 0.749 | 0.749 | 0.750 | 0.749 | 0.215 | 1.373 |
| | different domain (Video) | 0.736 | 0.735 | 0.752 | 0.717 | 0.206 | 1.372 |

Table 3: Lexicon-based sentiment classification of Amazon Kitchen product reviews.

| Method | | | IMDB | | Amazon | |
|---|---|---|---|---|---|---|
| | | | $AUC$ | $F_1$ | $AUC$ | $F_1$ |
| Unsupervised | Lexicon-based | pSenti with existing general-purpose lexicon | 0.808 | 0.705 | 0.818 | 0.747 |
| | | pSenti with induced domain-specific lexicon | 0.841 | 0.768 | 0.839 | 0.771 |
| | | ProbLex-DCM (Eisenstein, 2017) | 0.884 | 0.806 | 0.836 | 0.756 |
| | Learning-based | SVM$_{lin}$ trained on pseudo-labeled data | 0.863 | 0.771 | 0.845 | 0.763 |
| | | CNN trained on pseudo-labeled data | 0.879 | 0.781 | 0.849 | 0.773 |
| | | LSTM trained on pseudo-labeled data | **0.890** | **0.810** | **0.850** | **0.776** |
| Supervised | Learning-based | LSTM trained on real labeled data (full size) | 0.971 | 0.912 | 0.878 | 0.802 |
| | | " ($1/2$ size) | 0.934 | 0.862 | 0.852 | 0.752 |
| | | " ($1/4$ size) | 0.892 | 0.821 | 0.841 | 0.744 |
| | | " ($1/8$ size) | 0.850 | 0.746 | 0.831 | 0.735 |

Table 4: Sentiment classification of long texts.

only a few thousand (or less) labeled examples.

### 5.2 Sentiment Classification of Short Texts

To evaluate our proposed approach to sentiment classification of short texts, we have carried out experiments on the Twitter sentiment classification benchmark dataset from SemEval-2017 Task 4B (Rosenthal et al., 2017) which is to classify $6185$ tweets as either positive or negative. Other than the training set of $20,508$ tweets, we also collected unlabeled tweets using the Twitter API. All the tweets would be pre-processed by replacing emoticons with their corresponding text representations and encoding URLs by tokens. In addition to the Twitter-domain seed words listed in Table 1, we have also made use of common positive/negative emoticons which are ubiquitous on Twitter as additional seeds for the task of sentiment lexicon induction. Note that in all our experiments, we do not use the sentiment labels and the topic information provided in the training data.

Making use of the provided training data and our own unlabeled data collected from Twitter, we have constructed the domain-specific word embeddings, induced the sentiment lexicon, and bootstrapped the pseudo-labeled tweet data to train the binary tweet sentiment classifier. As the learning algorithm we have chosen LSTM with a hidden layer of 150 units which would be enough for tweets as they are quite short (with an average length of only 20 words).

The official performance measures for this short text sentiment classification task (Rosenthal et al., 2017) include Accuracy ($Acc$) and $F_1$. Although our approach is nearly-unsupervised (without any reliance on labeled documents), its performance on this benchmark dataset is comparable to that of supervised methods: it would be placed roughly in the middle of all the participating systems in this competition (see Table 5).

### 5.3 Detecting Neutral Sentiment

Many real-world applications of sentiment classification (e.g., on social media) are not simply a binary classification task, but involve a neutral category as well. Although many lexicon-based sentiment classifiers including pSenti can detect neutral sentiment, extending the above learning-based sentiment classifier (trained on pseudo-labeled data)

| System | | $Acc$ | $F_1$ |
|---|---|---|---|
| Unsupervised | Baseline $_{\text{all positive}}$ | 0.398 | 0.285 |
| | Baseline $_{\text{all negative}}$ | 0.602 | 0.376 |
| | **Ours**$_{\text{LSTM}}$ | **0.804** | **0.795** |
| Supervised | Worst system | 0.412 | 0.372 |
| | Median system | 0.802 | 0.801 |
| | Best system | 0.897 | 0.890 |

Table 5: Sentiment classification of short texts into two categories — SemEval-2017 Task 4B.

to recognize neutral sentiment is challenging. To investigate this issue, we have done experiments on the Twitter sentiment classification benchmark dataset from SemEval-2017 Task 4C (Rosenthal et al., 2017) which is to classify 12379 tweets into an ordinal five-point scale ($-2$, $-1$, $0$, $+1$, $+2$) where $0$ represents the neutral class.

One common way to handle neutral sentiment is to treat the set of neutral documents as a separate class for the classification algorithm, which is the method advocated by Koppel and Schler (2006). With the pseudo-labeled training examples of three classes ($-1$: negative, $0$: neutral, and $+1$: positive), we tried both standard multi-class classification (Hsu and Lin, 2002) and ordinal classification (Frank and Hall, 2001). However, neither of them could deliver a reasonable performance. After carefully inspecting the classification results, we realised that it is very difficult to have a set of representative training examples with good coverage for the neutral class. This is because the neutral class is not homogeneous: a document could be neutral because it is equally positive and negative, or because it does not contain any sentiment. In practice, the latter case is more often seen than the former case, and it implies that the neutral class is more often defined by the absence of sentiment word features rather than their presence, which would be problematic to most supervised learning algorithms.

What we discovered is that the simple method of identifying neutral documents from the binary sentiment classifier's decision boundary works surprisingly well, as long as the right thresholds are found. Specifically, we take the probabilistic outputs of a binary sentiment classifier trained as before, and then put all the documents whose probability of being positive lies not close to 0, not close to 1, but in the middle range into the neutral class. It turns out that *probability calibration* (Niculescu-Mizil and Caruana, 2005) is crucially important for this simple method to work. Some supervised learning algorithms for classification can give poor estimates of the class probabilities, and some even do not support probability prediction. For instance, maximum-margin learning algorithms such as SVM focus on hard samples that are close to the decision boundary (the support vectors), which makes their probability prediction biased. The technique of probability calibration allows us to better calibrate the probabilities of a given classifier, or to add support for probability prediction. If a classifier is well calibrated, its probabilistic output should be able to be directly interpreted as a confidence level on the prediction. For example, among the documents to which such a calibrated binary classifier gives a probabilistic output close to $0.8$, approximately $80\%$ of the documents would actually belong to the positive class.

Using the sigmoid model of Platt (2000) with cross-validation on the pseudo-labeled training data, we carry out probability calibration for our LSTM based binary sentiment classifier. Fig. 6 shows that the calibrated probability prediction aligns with the true confidence of prediction much better than the raw probability prediction. In this case, the Brier loss (Brier, 1950) that measures the mean squared difference between the predicted probability and the actual outcome could be reduced from 0.182 to 0.153 by probability calibration.

If we rank the estimated probabilities of being positive from low to high, the curve of probabilities would be in an "S"-shape with a distinct middle range where the slope is steeper than the two ends, as shown in Fig. 7. The documents with their probabilities of being positive in such a middle range should be neutral. Therefore the two elbow points in the probability curve would make appropriate thresholds for the identification of neutral sentiment, and they could be found automatically by a simple algorithm using the central difference to approximate the second derivative. Let $p_{\text{L}}$ and $p_{\text{U}}$ denote the identified thresholds ($p_{\text{L}} < p_{\text{U}}$), then we assign class label "$-1$" to all those documents with the probability below $p_{\text{L}}$, "$+1$" to all those documents with the proba-

| System | | $MAE^{\mu}$ | $MAE^{M}$ | mi$F_1$ | ma$F_1$ |
|---|---|---|---|---|---|
| Unsupervised | Baseline $_{\text{all -2}}$ | 1.895 | 2.000 | 0.006 | 0.014 |
| | Baseline $_{\text{all -1}}$ | 0.923 | 1.400 | 0.089 | 0.286 |
| | Baseline $_{\text{all 0}}$ | 0.525 | 1.200 | 0.133 | 0.500 |
| | Baseline $_{\text{all +1}}$ | 1.127 | 1.400 | 0.063 | 0.188 |
| | Baseline $_{\text{all +2}}$ | 2.105 | 2.000 | 0.004 | 0.011 |
| | Lexicon-based | 0.939 | 1.135 | 0.253 | 0.189 |
| | **Ours**$_{\text{LSTM}}$ | **0.536** | **0.815** | **0.537** | **0.326** |
| Supervised | Worst system | 0.985 | 1.325 | 0.250 | 0.121 |
| | Median system | 0.509 | 0.823 | 0.545 | 0.299 |
| | Best system | 0.554 | 0.481 | 0.504 | 0.405 |

Table 6: Sentiment classification of short texts on a five-point scale — SemEval-2017 Task 4C.

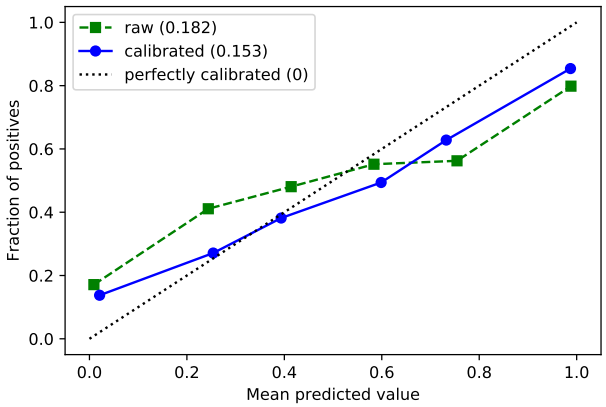



Figure 6: The probability calibration plot of our LSTM-based sentiment classifier on the SemEval-2017 Task 4C dataset.

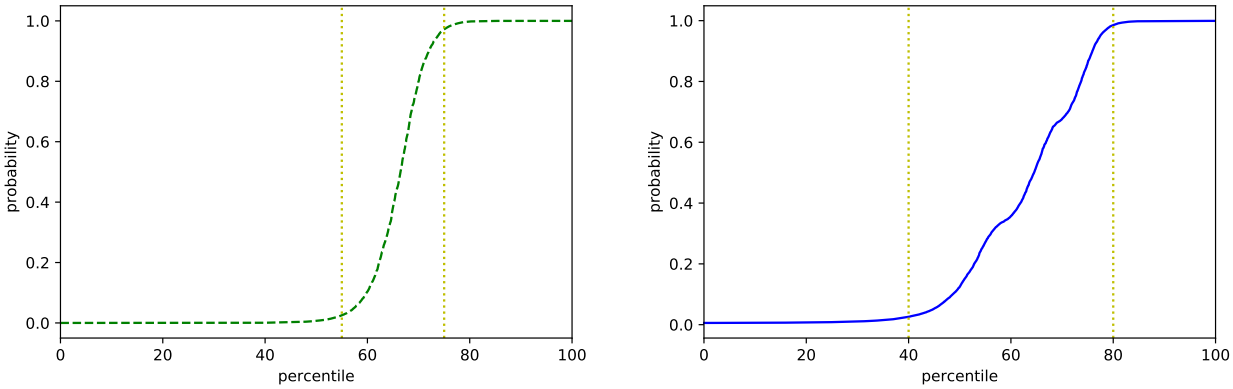
bility above $p_{\text{U}}$, and "0" to all those documents with the probability within $[p_{\text{L}}, p_{\text{U}}]$.

The official performance measures for this sentiment classification task (Rosenthal et al., 2017) are $MAE^{\mu}$ and $MAE^{M}$ which stand for micro-averaged and macro-averaged Mean Absolute Error (MAE), respectively. We would also like to report the micro-averaged and macro-averaged $F_1$ scores which are denoted as mi$F_1$ and ma$F_1$ respectively. As shown in Fig. 7, the thresholds identified from the raw probability curve are roughly at 55 percentile and 75 percentile, which would yield $MAE^{\mu} = 0.632$ and $MAE^{M} = 0.832$; the thresholds identified from the calibrated probability curve are roughly at 40 percentile and 80 percentile, which would yield much better scores $MAE^{\mu} = 0.536$ and $MAE^{M} = 0.815$. So with the help of probability calibration, our proposed approach would be able to comfortably beat all the baselines including the lexicon-based method pSenti (Mudinas et al., 2012) and compete with the average (median) participating systems (see Table 6). Please note that this is not a fair comparison: our approach is at a great disadvantage because (i) it is nearly-unsupervised, without any reliance on labeled documents while all the other systems are supervised; and (ii) it performs only ternary classification while all the other systems make classification on the full five-point scale.

## 6 Conclusions

How far can we go in sentiment classification for a new domain, given only unlabeled data? This paper presents our exploration towards answering the above research question. Specifically, the main contributions of this paper are as follows.

- We have formulated the cluster hypothesis for sentiment analysis (i.e., words with different sentiment polarities form distinct clusters) and verified that in general it holds for word embeddings within a specific domain but not across domains.
- We have demonstrated that a quality domain-specific sentiment lexicon can be induced from the word embeddings of that domain together with just a few seed words. Surprisingly, simple linear model based supervised learning algorithms (such as linear SVM) are good enough for this purpose; there is no benefit of utilizing non-linear models or semi-supervised/transductive learning algorithms due to the noise at the borders of sentiment word clusters. Using such linear models




(a) raw (b) calibrated

Figure 7: The probability curve with a region of intermediate probabilities representing the neutral class.

our system clearly outperforms the state-of-the-art sentiment lexicon induction method — SentProp (Hamilton et al., 2016).

- We have shown that a lexicon-based sentiment classifier could be enhanced by using its outputs as pseudo-labels and employing supervised learning algorithms such as LSTM to train a learning-based sentiment classifier on pseudo-labeled documents. Our end-to-end pipelined approach which, overall, is unsupervised (except for a very small set of seed words), works better than the state-of-the-art unsupervised technique for document sentiment classification — ProbLex-DCM (Eisenstein, 2017), and its performance is at least on par with an average fully supervised sentiment classifier trained on real labeled data (Rosenthal et al., 2017).
- We have revealed the crucial importance of probability calibration to the detection of neutral sentiment which was overlooked in previous studies (Koppel and Schler, 2006). With the right thresholds found, neutral documents can be simply identified at the binary sentiment classifier's decision boundary.

One promising way to further enhance the LSTM-based sentiment classifier in the proposed approach with the induced sentiment lexicon would be to concatenate word embeddings with an indicator feature which tells whether a current word is positive, neutral, or negative (Ebert et al., 2015). We leave this for future work.

## Acknowledgements

The Titan X Pascal GPU used for this research was kindly donated by the NVIDIA Corporation. We thank the reviewers for their constructive and helpful comments. We also gratefully acknowledge the support of Geek.AI for this work.

## References

Shlomo Argamon, Casey Whitelaw, Paul J. Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(6):802–822.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440—447, Prague, Czech Republic.

Danushka Bollegala, David J. Weir, and John A. Carroll. 2013. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 25(8):1719–1731.

Johan Bollen, Huina Mao, and Xiao-Jun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Glenn W. Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

Mathieu Cliche. 2017. BB_twtr at SemEval-2017 Task 4: Twitter sentiment analysis with CNNs and LSTMs. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval@ACL 2017)*, pages 573–580, Vancouver, Canada.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3079–3087, Montreal, Canada.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, pages 231–240, Palo Alto, CA, USA.

Sebastian Ebert, Ngoc Thang Vu, and Hinrich Schütze. 2015. A linguistically informed convolutional neural network. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA@EMNLP)*, pages 109–114, Lisbon, Portugal.

Jacob Eisenstein. 2017. Unsupervised learning for lexicon-based classification. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 3188–3194, San Francisco, CA, USA.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI)*, pages 4647–4657, San Jose, CA, USA.

Eibe Frank and Mark A. Hall. 2001. A simple approach to ordinal classification. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*, pages 145–156, Freiburg, Germany.

Fabian Gieseke, Antti Airola, Tapio Pahikkala, and Oliver Kramer. 2012. Sparse quasi-Newton optimization for semi-supervised support vector machines. In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 45–54, Vilamoura, Algarve, Portugal.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 513–520, Bellevue, WA, USA.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2017. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 28(10):2222–2232.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 595–605, Austin, TX, USA.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

James Hong and Michael Fang. 2015. Sentiment analysis with deeply learned distributed representations of variable length texts. Technical report, Stanford University.

Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks (TNN)*, 13(2):415–425.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177, Seattle, WA, USA.

Yohan Jo and Alice H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the 4th International Conference on Web Search and Web Data Mining (WSDM)*, pages 815–824, Hong Kong, China.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, pages 137–142, Chemnitz, Germany.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, pages 200–209, Bled, Slovenia.

Thorsten Joachims. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 290–297, Washington, DC, USA.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar.

Moshe Koppel and Jonathan Schler. 2006. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Heeyoung Lee, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky. 2014. On the importance of text analysis for stock price prediction. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 1170–1175, Reykjavik, Iceland.

Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 375–384, Hong Kong, China.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Bing Liu. 2015. *Sentiment Analysis — Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Marco Loog. 2016. Contrastive pessimistic likelihood estimation for semi-supervised classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(3):462–475.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–150, Portland, OR, USA.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, pages 165–172, Hong Kong, China.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119, Lake Tahoe, NV, USA.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text (CAAGET)*, pages 26–34, Los Angeles, CA, USA.

Andrius Mudinas, Dell Zhang, and Mark Levene. 2012. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the 1st International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM@KDD)*, pages 5:1–5:8, Beijing, China.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 625–632, Bonn, Germany.

Sara Owsley, Sanjay Sood, and Kristian J. Hammond. 2006. Domain specific affective classification of documents. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 181–183, Stanford, CA, USA.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 751–760, Raleigh, NC, USA.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–278, Barcelona, Spain.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, Stroudsburg, PA, USA.

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

John Platt, 2000. *Advances in Large Margin Classifiers*, chapter Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. MIT Press.

Robert Plutchik, 1984. *Approaches To Emotion*, chapter Emotions: A General Psychoevolutionary Theory, pages 197–219. Psychology Press.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

Filipe N Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto.

2016. Sentibench — A benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval@NAACL-HLT)*, pages 451–463, Denver, CO, USA.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval@ACL)*, pages 502–518, Vancouver, Canada.

Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. 2016. Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, pages 767–777, San Diego, CA, USA.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. 2009. Adapting naïve Bayes to domain adaptation for sentiment analysis. In *Proceedings of the 31th European Conference on IR Research (ECIR)*, pages 337–349, Toulouse, France.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology (JASIST)*, 61(12):2544–2558.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology (JASIST)*, 63(1):163–173.

Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 417–424, Philadelphia, PA, USA.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605.

Stefan Wager, Sida I. Wang, and Percy Liang. 2013. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26: Annual Conference on Neural Information Processing Systems (NIPS)*, pages 351–359, Lake Tahoe, NV, USA.

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

Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria. 2013. Feature ensemble plus sample selection: Domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3):10–18.

Yi Yang and Jacob Eisenstein. 2015. Putting things in context: Community-specific embedding projections for sentiment analysis. *CoRR*, abs/1511.06052.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 42–49, Berkeley, CA, USA.

Yasuhisa Yoshida, Tsutomu Hirao, Tomoharu Iwata, Masaaki Nagata, and Yuji Matsumoto. 2011. Transfer learning for multiple-domain sentiment analysis - identifying domain dependent/independent word polarity. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, San Francisco, CA, USA.

Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. 2011. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. Technical Report HPL-2011-89, HP Laboratories.

Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 912–919, Washington, DC, USA.