Contextual bidirectional long short-term memory recurrent neural network language models: a generative approach to sentiment analysis

A. Mousa, Björn Schuller

Angaben zur Veröffentlichung / Publication details:

Mousa, A., and Björn Schuller. 2017. “Contextual bidirectional long short-term memory recurrent neural network language models: a generative approach to sentiment analysis.” In *15th Conference of the European Chapter of the Association for Computational Linguistics - proceedings of conference : April 3-7, 2017, Valencia, Spain : EACL 2017 ; Volume 1: Long papers*, edited by Mirella Lapata, Phil Blunsom, and Alexander Koller, 1023–32. Stroudsburg, PA: Association for Computational Linguistics (ACL).

Contextual Bidirectional Long Short-Term Memory Recurrent Neural Network Language Models: A Generative Approach to Sentiment Analysis

Amr El-Desoky Mousa[1] and Björn Schuller[1,2]

[1]Chair of Complex & Intelligent Systems, University of Passau, Passau, Germany
[2]Department of Computing, Imperial College London, London, UK
amr.mousa@uni-passau.de
schuller@ieee.org

Abstract

Traditional learning-based approaches to sentiment analysis of written text use the concept of bag-of-words or bag-of-n-grams, where a document is viewed as a set of terms or short combinations of terms disregarding grammar rules or word order. Novel approaches de-emphasize this concept and view the problem as a sequence classification problem. In this context, recurrent neural networks (RNNs) have achieved significant success. The idea is to use RNNs as discriminative binary classifiers to predict a positive or negative sentiment label at every word position then perform a type of pooling to get a sentence-level polarity. Here, we investigate a novel generative approach in which a separate probability distribution is estimated for every sentiment using language models (LMs) based on long short-term memory (LSTM) RNNs. We introduce a novel type of LM using a modified version of bidirectional LSTM (BLSTM) called contextual BLSTM (cBLSTM), where the probability of a word is estimated based on its full left and right contexts. Our approach is compared with a BLSTM binary classifier. Significant improvements are observed in classifying the IMDB movie review dataset. Further improvements are achieved via model combination.

1 Introduction

Sentiment analysis of text (also known as opinion mining) is the process of computationally identifying and categorizing opinions expressed in a piece of text in order to determine the writer's attitude towards a particular topic. Due to the tremendous increase in web content, many organizations became increasingly interested in analyzing this big data in order to monitor the public opinion and assist decision making. Therefore, sentiment analysis attracted the interest of many researchers.

The task of sentiment analysis can be seen as a text classification problem. Depending on the target of the analysis, the classes can be described by continuous primitives such as valence, a polarity state (positive or negative attitude), or a subjectivity state (objective or subjective). In this work, we are interested in the binary classification of documents into a positive or negative attitude. Such detection of polarity is a non-trivial problem due to the existence of noise, comparisons, vocabulary changes, and the use of idioms, irony, and domain specific terminology (Schuller et al., 2015).

Traditional approaches to sentiment analysis rely on the concept of bag-of-words or bag-of-n-grams, where a document is viewed as a set of terms or short combinations of terms disregarding grammar rules or word order. In this case, usually, the analysis involves: tokenization and parsing of text documents, careful selection of important features (terms), dimensionality reduction, and classification of the documents into categories. For example, Pang et al. (2002) have considered different classifiers, such as Naive Bayes (NB), maximum entropy (MaxEnt), and support vector machines (SVM) to detect the polarity of movie reviews. Pang and Lee (2004) have combined polarity and subjectivity analysis and proposed a technique to filter out objective sentences of movie reviews based on finding minimum cuts in graphs. In (Taboada et al., 2011; Ding et al., 2008), lexicon-based techniques are examined, where word-level sentimental orientation scores are used to evaluate the polarity of product reviews. More advanced approaches utilize word or n-gram vectors, like in (Maas et al., 2011; Dahl et al., 2012).

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1023–1032,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

Novel approaches are mainly based on artificial neural networks (ANNs). These approaches de-emphasize the concept of bag-of-words or bag-of-n-grams. A document is viewed as a set of sentences, each sentence is a sequence of words. The sentiment problem is rather considered as a sequence classification problem. For example, in (Dong et al., 2014; Dong et al., 2016), RNN classifiers are used with an adaptive method to select relevant semantic composition functions for obtaining vector representations of sentences. This is found to improve sentiment classification on the Stanford Sentiment Treebank (SST). Rong et al. (2014) have used a RNN model to learn word representation simultaneously with the sentiment distribution. Santos and Gatti (2014) have proposed a convolutional neural network (CNN) that exploits from character- to sentence-level information to perform sentiment analysis on the Stanford Twitter Sentiment (STS) corpus. Kalchbrenner et al. (2014) have used a dynamic convolutional neural network (DCNN) with a dynamic k-max pooling to perform sentiment analysis on the SST and Twitter sentiment datasets. Lai et al. (2015) have utilized a combination of RNNs and CNNs called recurrent convolutional neural network (RCNN) to perform text classification on multiple datasets including sentiment analysis on the SST dataset.

Other novel approaches use tree structured neural models instead of sequential models (like RNNs) in order to capture complex semantic relationships that relate words to phrases. Despite their good performance, these models rely on existing parse trees of the underlying sentences which are, in most cases, not readily available or not trivial to generate. For example, Socher et al. (2013) have introduced a recursive neural tensor network (RNTN) to predict the compositional semantic effects in the SST dataset. In (Tai et al., 2015; Le and Zuidema, 2015), tree-structured LSTMs are used to improve the earlier models.

Another perspective to the sentiment problem is to assume that each sentence with a positive or negative class is drawn from a particular probability distribution related to that class. Then, instead of estimating a discriminative model that learns how to separate sentiment classes in sentence space, we estimate a generative model that tells us how these sentences are generated. This generative approach can be better or complementary in some sense to the discriminative approach.

The probability distributions over word sequences are well known as language models (LMs). They have also been used for sentiment analysis. However, no trial is made to go beyond simple bigram models. For example, Hu et al. (2007b) have estimated two separate positive and negative LMs from training collections. Tests are performed by computing the Kullback-Leibler divergence between the LM estimated from the test document and the sentiment LMs. Therein, uni- and bigram models are shown to outperform SVM models in classifying a movie review dataset. In (Hu et al., 2007a), a batch of terms in a domain are identified. Then, two different unigram LMs representing classifying knowledge for every term are built up from subjective sentences. A classifying function based on the generation of a test document is defined for the sentiment classification. This approach has outperformed SVM on a Chinese digital product review dataset. Liu et al. (2012) have employed an emoticon smoothed unigram LM to perform sentiment classification.

In this paper, we compare the generative LM approach with the discriminative binary classification approach. We estimate a separate probability distribution for each sentiment using long-span LMs based on unidirectional LSTMs (Sundermeyer et al., 2012) trained to predict a word depending on its full left context. The probability scores from positive and negative LMs are used to classify unseen sentences. In addition, we introduce a novel type of LM using a modified version of the standard bidirectional LSTM called contextual bidirectional LSTM (cBLSTM). In contrast to the unidirectional model, this model is trained to predict a word depending on its full left and right contexts. Moreover, we combine the LM approach with the binary classification approach using linear interpolation of probabilities. We observe that the cBLSTM LM outperforms both the LSTM LM and the BLSTM binary classifier. Combining approaches together yields further improvements. Models are evaluated on the IMDB large movie review dataset[1] (Maas et al., 2011).

2 Language Models

A statistical LM is a probability distribution over word sequences that assigns a probability $p(w_1^M)$ to any word sequence w_1^M of length M. Thus, it provides a way to estimate the relative likelihood

[1] http://ai.stanford.edu/~amaas/data/sentiment/

of different phrases. It is a widely used model in many natural language processing tasks, like automatic speech recognition, machine translation, and information retrieval. Usually, to estimate a LM, the assumption of the $(n-1)^{th}$ order Markov process is used (Bahl et al., 1983), in which a current word w_m is assumed conditionally dependent on the preceding $(n-1)$ history words, such that:

$$p(w_1^M) \approx \prod_{m=1}^{M} p(w_m|w_{m-n+1}^{m-1}). \qquad (1)$$

This is called an n-gram LM. A conventional approach to estimate these probabilities is the back-off LM which is based on count statistics collected from the training text. In addition to the initial n-gram approximation, a major drawback of this model is that it backs-off to a shorter history whenever insufficient statistics are observed for a given n-gram. Novel state-of-the-art LMs are based on ANNs like RNNs that provide long-span probabilities conditioned on all predecessor words (Mikolov et al., 2010; Kombrink et al., 2011).

3 Unidirectional RNN Models

3.1 Standard RNN

A RNN maps from a sequence of input observations to a sequence of output labels. The mapping is defined by a set of activation weights and a non-linear activation function. Recurrent connections allow to access activations from past time steps. For an input sequence x_1^T, a RNN computes the hidden sequence h_1^T and the output sequence y_1^T by performing the following operations for time steps $t = 1$ to T (Graves et al., 2013):

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \qquad (2)$$
$$y_t = W_{hy}h_t + b_y, \qquad (3)$$

where $\mathcal{H}$ is the hidden layer activation function, W_{xh} is the weight matrix between the input and hidden layer, W_{hh} is the recurrent weight matrix between the hidden layer and itself, W_{hy} is the weight matrix between the hidden and output layer, b_h and b_y are the hidden and output layer bias vectors respectively. $\mathcal{H}$ is usually an element-wise application of the sigmoid function.

3.2 LSTM RNN

In (Hochreiter and Schmidhuber, 1997), an alternative RNN called Long Short-Term Memory (LSTM) is introduced where the conventional neuron is replaced with a so-called *memory cell* controlled by input, output and forget gates in order to overcome the vanishing gradient problem of traditional RNNs. In this case, $\mathcal{H}$ can be described by the following composite function:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \qquad (4)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \qquad (5)$$
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \qquad (6)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \qquad (7)$$
$$h_t = o_t \tanh(c_t), \qquad (8)$$

where σ is the sigmoid function, i,f,o, and c are respectively the input, forget, output gates, and cell activation vectors (Graves et al., 2013).

3.3 LSTM LM

In a LSTM LM, the time steps correspond to the word positions in a training sentence. At every time step, the network takes as input the word at the current position encoded as a 1-hot binary vector. The input vector is then passed to one or more recurrent hidden layers with self connections that implicitly take into account all the previous history words presented to the network. The output of the final hidden layer is passed to an output layer with a softmax activation function to produce a correctly normalized probability distribution. The target output at each word position is the next word in the sentence. A cross-entropy loss function is used which is equivalent to maximizing the likelihood of the training data. At the end, the network can predict the long-span conditional probability $p(w_m|w_1^{m-1})$ for any word $w_m \in V$ and a given history w_1^{m-1}, where V is the vocabulary. Fig. 1 shows an unfolded example of a LSTM LM over a sentence <s> w_1 w_2 w_3 </s>, where <s> and </s> are the sentence start and end symbols.

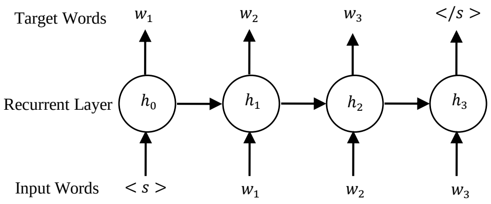

Figure 1: Architecture of a LSTM LM predicting a word given its full previous history.

4 Bidirectional RNN Models

4.1 BLSTM RNN

A BLSTM processes input sequences in both directions with two sub-layers in order to account for the full input context. These two sub-layers compute forward and backward hidden sequences $\overrightarrow{h}$, $\overleftarrow{h}$ respectively, which are then combined to compute the output sequence y (see Fig. 2), thus:

$$\overrightarrow{h}_t = \mathcal{H}(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \quad (9)$$

$$\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \quad (10)$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \quad (11)$$

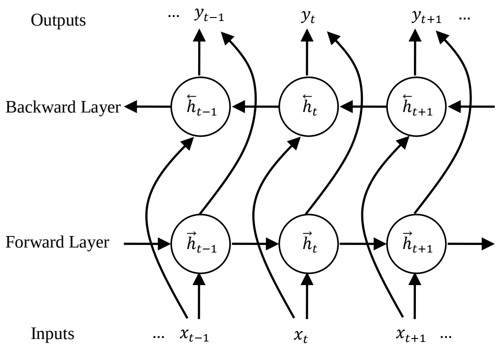

Figure 2: Architecture of a BLSTM, every output depends on the whole input sequence.

4.2 Contextual BLSTM LM

The standard BLSTM described in Section 4.1 is not suitable for estimating LMs. This is because it predicts every output symbol depending on the whole input sequence. Since a LM indeed uses the same word sequence in both input and target sides of the network, it would be incorrect to predict a word given the whole input sentence. Rather, it is required to predict a word given the full left and right context while excluding the predicted word itself from the conditional dependence. To allow for this, we modify the architecture of the standard BLSTM such that it accounts for a contextual dependence rather than a full sequence dependence. The new model is called a contextual BLSTM or cBLSTM in short. The architecture of this model is illustrated in Fig. 3.

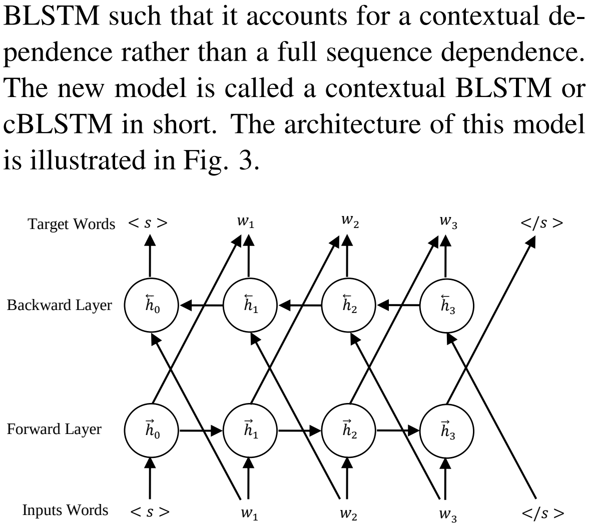

Figure 3: Architecture of a cBLSTM LM predicting a word given its full left and right contexts.

The model consists of a forward and a backward sub-layer. The forward sub-layer receives the encoded input words staring from the sentence start symbol up to the last word before the sentence end symbol (sequence $< s > \;\; w_1\, w_2\, w_3$ in Fig. 3). The forward hidden states are used to predict words starting from the first word after the sentence start symbol up to the sentence end symbol (sequence $w_1\, w_2\, w_3 \;\; </s>$ in Fig. 3). The backward sub-layer does exactly the reverse operation. The two sub-layers are interleaved together in order to adjust the conditional dependence such that the prediction of any target word depends on the full left and right contexts. Note that the hidden state at the first as well as the last time step needs to be padded by zeros so that the size of the hidden vector is consistent across all time steps. At the end, the model can effectively predict the conditional probability $p(w_m | w_1^{m-1}, w_{m+1}^M)$ for any word $w_m \in V$, left context w_1^{m-1} and right context w_{m+1}^M, where V is the vocabulary and M is the length of the sentence. Table 1 shows the predicted probability at each time step of Fig. 3. Note that one direction dependence is maintained at the start and end of sentence (time steps 1 and 5).

time step	predicted conditional prob.
1	$p(<s> \;\mid w_1\, w_2\, w_3 \;\; </s>)$
2	$p(w_1 \mid <s> \;,\; w_2\, w_3 \;\; </s>)$
3	$p(w_2 \mid <s> \;\; w_1 \;,\; w_3 \;\; </s>)$
4	$p(w_3 \mid <s> \;\; w_1\, w_2 \;,\; </s>)$
5	$p(</s> \;\mid <s> \;\; w_1\, w_2\, w_3)$

Table 1: Predicted conditional probabilities at every time step of the cBLSTM shown in Fig. 3.

Our implementation of the novel cBLSTM RNN model is integrated into our publicly available CURRENNT[2] toolkit initially introduced by Weninger et al. (2014). A new version of the toolkit with the novel implementations is planned to be available by the date of publication.

Here, it is worth noting that deep cBLSTM models can not be easily constructed by stacking multiple hidden bidirectional layers together. The reason is that the hidden states obtained after the first bidirectional layer are dependent on the full left and right contexts. If these states are utilized as inputs to a second bidirectional layer that identically repeats the same operation again, then the desired conditional dependence will not be correctly

[2] http://sourceforge.net/p/currennt

maintained. One method to solve this problem is to create deeper models by stacking multiple forward and backward sub-layers independently. The fusion of both sub-layers takes place and the end of the deep stack. The implementation of such a deep cBLSTM model is not yet available.

5 Dataset

Our experiments on sentiment analysis are performed on the IMDB large movie review dataset (v1.0) introduced by Maas et al. (2011). The labeled partition of the dataset consists of 50k balanced full-length movie reviews with 25k positive and 25k negative reviews extracted from the Internet Movie Database (IMDB)[3].

Since the reviews are in a form of long paragraphs which are difficult to handle directly with RNNs, we break down these paragraphs into relatively short sentences based on punctuation clues. After breaking down the paragraphs, the average number of sentences per review is around 13 sentences. We randomly selected 1000 positive and 1000 negative reviews as our test set. A similar number of random reviews are selected as a development set. The remaining reviews are used as a training set. Note that this is not the official dataset division provided by Maas et al. (2011), where 25k balanced reviews are dedicated for training and the other 25k balanced reviews are dedicated for testing. The reasons not to follow the official division are firstly that, it does not provide a development set; secondly, our proposed models need much data to train well as revealed by initial experiments; thirdly, it would be very time consuming to use the whole data as one partition and perform multi-fold cross validation as usually adopted with large sentiment datasets (Schuller et al., 2015). A preprocessed version of the IMDB dataset with the modified partitioning is planned to be available for download by the date of publication.

A word list of the 10k most frequent words is selected as our vocabulary. This covers around 95% of the words in our development and test sets. Any out-of-vocabulary word is mapped to a special *unk* symbol. We use the classification accuracy as our evaluation measure. The unweighted average F1 scores over positive and negative classes are also calculated. However, their values are found almost similar to the classification accuracies. Therefore, only classification accuracies are reported.

6 Related Work

The work of this paper is closely related to several previous publications that report sentiment classification accuracy on the same dataset. For example, in (Maas et al., 2011), the IMDB dataset is introduced and a semi-supervised word vector induction framework is used, where an unsupervised probabilistic model similar to latent Dirichlet allocation (LDA) is proposed to learn word vectors. Another supervised model is utilized to constrain words expressing similar sentiment to have similar representations in vector space. In (Dahl et al., 2012), documents are treated as bags of n-grams. Restricted Boltzmann machines (RBMs) are used to extract vector representations for n-grams. Then, a linear SVM model is utilized to classify documents based on the resulting feature vectors. Wang and Manning (2012) have used a variant of SVM with Naive Bayes log-count ratios as well as word bigrams as features. This modified SVM model is referred to as NBSVM. In our previous publication (Schuller et al., 2015), LSTM LMs trained on 40% of the whole IMDB dataset are used for performing sentiment analysis. However, a carefully tuned MaxEnt classifier is found to perform better. Le and Mikolov (2014) have used a paragraph vector methodology with an unsupervised algorithm based on feed-forward neural networks that learns fixed-length vector representations from variable-length texts. All these publications use the official IMDB dataset division except for (Schuller et al., 2015), where a similar division as in this paper is used. To give a comprehensive idea about the aforementioned techniques, we show in Table 2 the classification results as reported in the related publications. Note that only the results of (Schuller et al., 2015) are directly comparable to our results.

experiment	Accuracy [%]
Maas et al. (2011)	88.89
Dahl et al. (2012)	89.23
Wang and Manning (2012)	91.22
Schuller et al. (2015)*	91.55
Le and Mikolov (2014)	92.58

Table 2: Sentiment classification accuracies from previous publications on the IMDB dataset.

In relation to our novel cBLSTM LM, previous trials have been made to estimate bidirectional

[3] http://www.imdb.com

LMs. For example, in (Frinken et al., 2012), distinct forward and backward LMs are estimated for handwriting recognition. However, no trial is made to go beyond 4-gram models. In (Xiong et al., 2016), standard forward and backward RNN LMs are separately estimated for a conversational speech recognition task. The log probabilities from both models are added. In (Arisoy et al., 2015), bidirectional RNNs and LSTMs are used to estimate LMs for an English speech recognition task. Therein, the standard bidirectional architecture (as in Fig. 2) is used without modifications. This causes circular dependencies to arise when combining probabilities from multiple time steps. Therefore, pseudo-likelihoods are utilized rather than true likelihoods which is not perfect from the mathematical point of view. Not surprisingly, the BLSTM LMs do not yield any gain over the LSTM LMs. In addition, the perplexity of such a model becomes invalid. More importantly, in (Peris and Casacuberta, 2015), bidirectional RNN LMs are used for a statistical machine translation task. However, only standard RNNs but not LSTMs are utilized. Furthermore, no details are provided about how the model is exactly modified and how the left and right dependencies are maintained over time steps.

7 Sentiment Classification

7.1 Generative LM-based classifier

Our first approach to sentiment classification is the generative approach based on LMs. We either use LSTM LMs described in Section 3.3 or cBLSTM LMs described Section 4.2. Two separate LMs are estimated from positive and negative training data. We use networks with a single hidden layer that consists of 300 memory cells followed by a softmax layer with a dimension of $10k + 3$. This is equal to the full vocabulary size in addition to <s>, </s>, and *unk* symbols representing sentence start, sentence end, and unknown word symbols respectively. In case of using cBLSTM networks, a single hidden layer of 600 memory cells is used (300 cells for each forward and backward sub-layer). A cross-entropy loss function is used with a momentum of 0.9. We use sentence-level mini-batches of size 100 sentences computed in parallel. The learning rate is set initially to 10^{-3} and then decreased gradually to 10^{-6}. The training process is controlled by monitoring the cross-entropy error over the development set.

In addition, we use a data sub-sampling methodology during training. For this purpose, a traditional 5-gram backoff LM is created out of the development data, we call this a *ranking LM*. Then, all training sentences are ranked according to their perplexities with the ranking LM. Using these ranks, we divide our training sentences into three partitions that reflect the relative importance of the data, such that the first partition contains the 100k sentences with the lowest perplexities, the second partition contains the 100k sentences with next lowest perplexities. The third partition contains all the other sentences. Instead of using the whole training data in each epoch, we use a random sample with more sentences from the first two partitions than the third one. After a sufficient number of epochs, the whole training data is covered. The sub-sampling approach speeds up the training and makes it feasible with any size of training data. At the same time, the training is focused on the relatively more important examples. In addition, it adds a useful regularization to the training process. Yet, it leads to a less smoother convergence. To show the efficiency of our sentence ranking methodology, Table 3 shows examples of the highest and lowest ranked sentences from positive and negative training data.

most +ve	this is one of the best films ever made.
least +ve	cheap laughs but great value.
most -ve	this is one of the worst movies i have ever seen.
least -ve	life's too short.

Table 3: Examples of the highest/lowest ranked sentences from positive/negative training data.

After training the neural networks, each of the positive and negative sentiment LM estimates a probability distribution for the corresponding sentiment, we call these probability distributions p_+ and p_-. To evaluate the sentiment of some test review, we calculate the perplexity of each model p_+ and p_- with respect to the whole review. Thus, given a probability distribution p, and a review text S composed of K sentences $S = s_1, ..., s_K$, each sentence $s_k : 1 \leq k \leq K$ is composed of a sequence of M_k words $s_k = w_1^k, w_2^k, ..., w_{M_k}^k$; we calculate the perplexity $PP_p(S)$ of a model p with respect to text S. It is a very common measurement of how well a probability distribution predicts a sample. A low perplexity indicates that the probability distribution is good at predicting the sample. Perplexity is defined as the exponentiated negative average log-likelihood, or in other words, the inverse of the geometric average probability assigned by the model to each word in the sample. We calculate the Perplexity using Equation 12 if the model p is based on LSTM, and using Equation 13 if the model is based on cBLSTM:

$$PP_p(S)=\left[\prod_{k=1}^{K}\prod_{m=1}^{M_k} p(w_m^k|w_1^k, w_2^k, .., w_{m-1}^k)\right]^{\frac{-1}{N}} \tag{12}$$

$$PP_p(S) = \left[\prod_{k=1}^{K}\prod_{m=1}^{M_k} p(w_m^k|w_1^k, w_2^k, .., w_{m-1}^k; w_{m+1}^k, w_{m+2}^k, ..., w_{M_k}^k)\right]^{\frac{-1}{N}}, \tag{13}$$

where $N = \sum_{k=1}^{K} M_k$ is the total number of words in text S. Then, a sentiment polarity $\mathcal{P} \in \{-1, +1\}$ is assigned to S according to the following decision rule:

$$\mathcal{P}(S) = \begin{cases} +1 & \text{if} \quad PP_{p_+}(S) < PP_{p_-}(S) \\ -1 & \text{otherwise} \end{cases}. \tag{14}$$

7.2 Discriminative BLSTM-based Binary Classifier

Our second approach to sentiment classification is the discriminative approach based on BLSTM RNNs described in Section 4.1. We use BLSTM networks with a single hidden layer that consists of 600 memory cells (300 cells for each forward and backward sub-layer). Since the BLSTM performs a binary classification task, only a single output neuron is used with a sigmoid activation function. A cross-entropy loss function is used with a momentum of 0.9. The same training settings like the case of LSTM/cBLSTM LMs are used including sub-sampling with the same partitioning of the training data. However, a single training dataset with all positive and negative reviews is used. For a sentence with a positive sentiment, the target outputs are set to *ones* at all time steps. For a sentence with a negative sentiment, the target outputs are set to *zeros* at all time steps. Since the sigmoid function provides output values in the interval [0,1], the network is trained to produce the probability of the positive class at every time step. Although the output of the BLSTM network at a given time step is dependent on the whole input sequence, it is widely known that every output is more affected by the inputs at closer time steps in both directions. Therefore, a sentence-level sentiment can be deduced by comparing the average probability mass assigned to the positive class over all time steps with the average probability mass assigned to the negative class. Thus, similar to Section 7.1, given a review text S composed of K sentences, each sentence is a sequence of M_k words, we calculate two probabilities $p_+(S)$ and $p_-(S)$ that the review S has a positive or negative sentiment using Equations 15 and 16 respectively:

$$p_+(S) = \frac{1}{N}\sum_{k=1}^{K}\sum_{m=1}^{M_k} p_+(w_m^k) \tag{15}$$

$$p_-(S) = \frac{1}{N}\sum_{k=1}^{K}\sum_{m=1}^{M_k}(1 - p_+(w_m^k)), \tag{16}$$

where N is the total number of words in text S, and $p_+(w_m^k)$ is the probability that a positive class is assigned to the word at position m of the k^{th} sentence of the review S. Then, a sentiment polarity $\mathcal{P} \in \{-1, +1\}$ is assigned to S according to the following decision rule:

$$\mathcal{P}(S) = \begin{cases} +1 & \text{if} \quad p_+(S) > p_-(S) \\ -1 & \text{otherwise} \end{cases}. \tag{17}$$

7.3 Model Combination

The probability scores of the generative LM-based classifier and the discriminative BLSTM-based binary classifier discussed in Sections 7.1 and 7.2 can be combined together via linear interpolation. This is achieved by first normalizing the probabilities from the LMs such that the probabilities of positive and negative classes for a given review are summed up to 1.0. Note that this normalization property holds by default for the BLSTM-based binary classifier. Then, the probabilities of both models are linearly interpolated to obtain a single probability score. The interpolation weights are optimized on the development data.

8 Experimental Results

Table 4 shows the results of our experiments. All the neural networks in this work are trained and optimized using our own CURRENNT toolkit (Weninger et al., 2014). Both the LSTM and cBLSTM LMs are linearly interpolated with two additional LMs, namely a 5-gram backoff LM smoothed with modified Kneser-Ney smoothing (Kneser and Ney, 1995), and another 5-gram MaxEnt LM (Alumäe and Kurimo, 2010). These two models are estimated using the SRILM language modeling toolkit (Stolcke, 2002).

classification model	**Acc. [%]**
LSTM LM	89.58
+ 5-grm backoff LM	91.05
+ 5-grm MaxEnt LM	91.23
cBLSTM LM	89.88
+ 5-grm backoff LM	91.38
+ 5-grm MaxEnt LM	91.48
BLSTM binary classifier	90.15
LSTM LM + BLSTM binary classifier	92.35
cBLSTM LM + BLSTM binary classifier	**92.83**
Schuller et al. (2015) LSTM + 5-grm LM	90.50
Schuller et al. (2015) MaxEnt classifier	91.55

Table 4: Sentiment classification accuracies measured on the IMDB dataset.

We observe that the use of cBLSTM LM as a generative sentiment classifier significantly outperforms the use of both LSTM LM and BLSTM discriminative binary classifiers. The statistical significance is verified using a bootstrap method of significance analysis described by Bisani and Ney (2004). The probability of improvement (POI_{boot}) is around 95%. Combining LM-based classifiers with BLSTM-based binary classifiers via linear interpolation of probabilities achieves further improvements. Our best accuracy (92.83%) is obtained by combining the cBLSTM LM classifier with the BLSTM binary classifier. These results reveal that both the generative and discriminative approaches are complementary in solving the sentiment classification problem.

Finally, our best result is better than the best previously published result in (Schuller et al., 2015) on the same IMDB dataset with the same dataset partitioning. Even though they are not directly comparable, our results are better than other previously published results reported in Table 2 where a different dataset partitioning is used.

For illustration, Table 5 shows two examples of positive and negative reviews that could not be correctly classified by the discriminative BLSTM binary classifier, however they are correctly classified by the cBLSTM LM classifier. We can observe the implicit indication of the writer's attitude towards the movie which can not be easily captured by simple approaches. In this case, learning a separate long-span bidirectional probability distribution for each sentiment seems to help.

+ve	low budget mostly no name actors. this is what a campy horror flick is supposed to be all about. these are the types of movies that kept me on the edge of my seat as a kid staying up too late to watch cable. if you liked the eighties horror scene this is the movie for you.
-ve	i and a friend rented this movie. we both found the movie soundtrack and production techniques to be lagging. the movie's plot appeared to drag on throughout with little surprise in the ending. we both agreed that the movie could have been compressed into roughly an hour giving it more suspense and moving plot.

Table 5: Examples of reviews correctly classified by the cBLSTM LM classifier.

9 Conclusions

We have introduced a generative approach to sentiment analysis in which a novel contextual BLSTM (cBLSTM) LM is used as a sentiment classifier. Separate LM probability distributions are estimated for positive and negative sentiment from the training data. Then, probability scores from these LMs are utilized to classify test data. Results have been compared with a discriminative sentiment classification approach that uses a BLSTM-based binary classifier. We have observed that the generative cBLSTM LM approach significantly outperforms other approaches. Models have been evaluated on the IMDB large movie review dataset. The proposed models have achieved better results than the previously published results on the same dataset with the same partitioning. In addition, indicative comparisons have been made with the previously published results on the same dataset with different partitioning. Using model combination, we could achieve further performance improvement indicating that both the generative and discriminative approaches are complementary in solving the sentiment analysis problem. Moreover, we have introduced an efficient methodology based on perplexity calculation to partition the training data according to relative importance to the learning task. This partitioning methodology has been combined with a sub-sampling technique to efficiently train the neural networks on large data. As a future work, we plan to investigate deeper cBLSTM as well as hybrid recurrent and convolutional models. Another direction is to experiment with pre-trained word vectors.

Acknowledgments

The research leading to these results has received funding from the European Unions Horizon 2020 Programme through the Research and Innovation Action #645378 (ARIA-VALUSPA), the Innovation Action #644632 (MixedEmotions), as well as the German Federal Ministry of Education, Science, Research and Technology (BMBF) under grant agreement #16SV7213 (EmotAsS). We further thank the NVIDIA Corporation for their support of this research by Tesla K40 GPU donation.

References

Tanel Alumäe and Mikko Kurimo. 2010. Efficient estimation of maximum entropy language models with N-gram features: an SRILM extension. In *Proc. Interspeech Conference of the International Speech Communication Association*, pages 1820–1823, Makuhari, Chiba, Japan, September.

Ebru Arisoy, Abhinav Sethy, Bhuvana Ramabhadran, and Stanely Chen. 2015. Bidirectional recurrent neural network language models for automatic speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 5421–5425, Brisbane, Australia, April.

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:179 – 190, March.

Maximilian Bisani and Hermann Ney. 2004. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 409 – 412, Montreal, Canada, May.

George E. Dahl, Ryan Prescott Adams, and Hugo Larochelle. 2012. Training restricted boltzmann machines on word observations. In *Proc. International Conference on Machine Learning*, pages 679–686, Edinburgh, Scotland, UK, June.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proc. International Conference on Web Search and Data Mining*, pages 231–240, Palo Alto, California, USA, February.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2014. Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis. In *Proc. AAAI Conference on Artificial Intelligence*, pages 1537–1543, Québec, Québec, Canada, July.

Li Dong, Furu Wei, Ke Xu, Shixia Liu, and Ming Zhou. 2016. Adaptive multi-compositionality for recursive neural network models. *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 24(3):422–431.

Volkmar Frinken, Alicia Fornés, Josep Lladós, and Jean-Marc Ogier, 2012. *Bidirectional Language Model for Handwriting Recognition*, pages 611–619. Springer Berlin Heidelberg, Berlin, Heidelberg.

Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 6645 – 6649, Vancouver, BC, Canada, May.

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

Yi Hu, Ruzhan Lu, Yuquan Chen, and Jianyong Duan. 2007a. Using a generative model for sentiment analysis. *International Journal of Computational Linguistics & Chinese Language Processing*, 12(2):107–126, June.

Yi Hu, Ruzhan Lu, Xuening Li, Yuquan Chen, and Jianyong Duan. 2007b. A language modeling approach to sentiment analysis. In *Proc. International Conference on Computational Science*, pages 1186–1193, Beijing, China, May.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proc. Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 655–665, Baltimore, MD, USA, June.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for M-gram language modeling. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181 – 184, Detroit, Michigan, USA, May.

Stefan Kombrink, Tomáš Mikolov, Martin Karafiát, and Lukáš Burget. 2011. Recurrent neural network based language modeling in meeting recognition. In *Proc. Interspeech Conference of the International Speech Communication Association*, pages 2877 – 2880, Florence, Italy, August.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proc. AAAI Conference on Artificial Intelligence*, pages 2267–2273, Austin, Texas, USA, January.

Quoc V. Le and Tomáš Mikolov. 2014. Distributed representations of sentences and documents. In *Proc. International Conference on Machine Learning*, pages 1188–1196, Beijing, China, June.

Phong Le and Willem Zuidema. 2015. Compositional distributional semantics with long short term memory. In *Proc. Joint Conference on Lexical and Computational Semantics*, pages 10–19, Denver, CO, USA, June.

Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. 2012. Emoticon smoothed language models for twitter sentiment analysis. In *Proc. AAAI Conference on Artificial Intelligence*, pages 1678–1684, Toronto, Ontario, Canada, July.

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 *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 142–150, Portland, Oregon, USA, June.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan H. Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proc. Interspeech Conference of the International Speech Communication Association*, pages 1045 – 1048, Makuhari, Chiba, Japan, September.

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

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proc. Conference on Empirical Methods in NLP*, volume 10, pages 79–86, Philadelphia, PA, USA, July.

Álvaro Peris and Francisco Casacuberta. 2015. A bidirectional recurrent neural language model for machine translation. *Procesamiento del Lenguaje Natural*, 55:109–116, September.

Wenge Rong, Baolin Peng, Yuanxin Ouyang, Chao Li, and Zhang Xiong. 2014. Structural information aware deep semi-supervised recurrent neural network for sentiment analysis. *Frontiers of Computer Science*, 9(2):171–184.

Cícero Nogueira dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proc. International Conference on Computational Linguistics*, pages 69–78, Dublin, Ireland, August.

Björn Schuller, Amr E. Mousa, and Vryniotis Vasileios. 2015. Sentiment analysis and opinion mining: On optimal parameters and performances. *WIREs Data Mining and Knowledge Discovery*, 5:255–263, September/October.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. Conference on Empirical Methods in NLP*, pages 1631–1642, Seattle, WA, USA, October.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. International Conference on Spoken Language Processing*, volume 2, pages 901 – 904, Denver, Colorado, USA, September.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Proc. Interspeech Conference of the International Speech Communication Association*, Portland, OR, USA, September.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, June.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 1556–1566, Beijing, China, July.

Sida I. Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 90 – 94, Jeju Island, Korea, July.

Felix Weninger, Johannes Bergmann, and Björn Schuller. 2014. Introducing CURRENNT – the Munich open-source CUDA RecurREnt Neural Network Toolkit. *Journal of Machine Learning Research*, 15(99), October.

Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. 2016. Achieving human parity in conversational speech recognition. Technical Report MSR-TR-2016-71, Microsoft Research, October.