A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis

Sebastian Ruder\textsuperscript{1,2}, Parsa Ghaﬀari\textsuperscript{2}, and John G. Breslin\textsuperscript{1}

\textsuperscript{1}Insight Centre for Data Analytics
National University of Ireland, Galway
\{sebastian.ruder,john.breslin\}@insight-centre.org

\textsuperscript{2}Aylien Ltd.
Dublin, Ireland
\{sebastian,parsa\}@aylien.com

Abstract

Opinion mining from customer reviews has become pervasive in recent years. Sentences in reviews, however, are usually classified independently, even though they form part of a review’s argumentative structure. Intuitively, sentences in a review build and elaborate upon each other; knowledge of the review structure and sentential context should thus inform the classification of each sentence. We demonstrate this hypothesis for the task of aspect-based sentiment analysis by modeling the interdependencies of sentences in a review with a hierarchical bidirectional LSTM. We show that the hierarchical model outperforms two non-hierarchical baselines, obtains results competitive with the state-of-the-art, and outperforms the state-of-the-art on five multilingual, multi-domain datasets without any hand-engineered features or external resources.

1 Introduction

Sentiment analysis (\?) is used to gauge public opinion towards products, to analyze customer satisfaction, and to detect trends. With the proliferation of customer reviews, more fine-grained aspect-based sentiment analysis (ABSA) has gained in popularity, as it allows aspects of a product or service to be examined in more detail.

Reviews – just with any coherent text – have an underlying structure. A visualization of the discourse structure according to Rhetorical Structure Theory (RST) (\?) for the example review in Figure figure 1 reveals that sentences and clauses are connected via different rhetorical relations, such as \textit{Elaboration} and \textit{Background}.

Intuitively, knowledge about the relations and the sentiment of surrounding sentences should inform the sentiment of the current sentence. If a reviewer of a restaurant has shown a positive sentiment towards the quality of the food, it is likely that his opinion will not change drastically over the course of the review. Additionally, overwhelmingly positive or negative sentences in the review help to disambiguate sentences whose sentiment is equivocal.

Neural network-based architectures that have recently become popular for sentiment analysis and ABSA, such as convolutional neural networks (\?), LSTMs (\?), and recursive neural networks (\?), however, are only able to consider intra-sentence relations such as \textit{Background} in Figure figure1 and fail to capture inter-sentence relations, e.g. \textit{Elaboration} that rely on discourse structure and provide valuable clues for sentiment prediction.

We introduce a hierarchical bidirectional long short-term memory (H-LSTM) that is able to leverage both intra- and inter-sentence relations. The sole dependence on sentences and their structure within a review renders our model fully language-
independent. We show that the hierarchical model outperforms strong sentence-level baselines for aspect-based sentiment analysis, while achieving results competitive with the state-of-the-art and outperforming it on several datasets without relying on any hand-engineered features or sentiment lexica.

2 Related Work

Aspect-based sentiment analysis. Past approaches use classifiers with expensive hand-crafted features based on n-grams, parts-of-speech, negation words, and sentiment lexica (?. ?). The model by Zhang and Lan (?) is the only approach we are aware of that considers more than one sentence. However, it is less expressive than ours, as it only extracts features from the preceding and subsequent sentence without any notion of structure. Neural network-based approaches include an LSTM that determines sentiment towards a target word based on its position (?. ?) as well as a recursive neural network that requires parse trees (?. In contrast, our model requires no feature engineering, no positional information, and no parser outputs, which are often unavailable for low-resource languages. We are also the first – to our knowledge – to frame sentiment analysis as a sequence tagging task.

Hierarchical models. Hierarchical models have been used predominantly for representation learning and generation of paragraphs and documents: Li et al. (?) use a hierarchical LSTM-based autoencoder to reconstruct reviews and paragraphs of Wikipedia articles. Serban et al. (?) use a hierarchical recurrent encoder-decoder with latent variables for dialogue generation. Denil et al. (?) use a hierarchical ConvNet to extract salient sentences from reviews, while Kotzias et al. (?) use the same architecture to learn sentence-level labels from review-level labels using a novel cost function. The model of Lee and Dernoncourt (?) is perhaps the most similar to ours. While they also use a sentence-level LSTM, their class-level feed-forward neural network is only able to consider a limited number of preceding texts, while our review-level bidirectional LSTM is theoretically able to consider an unlimited number of preceding and successive sentences.

3 Model

In the following, we will introduce the different components of our hierarchical bidirectional LSTM architecture displayed in Figure 2.2.

3.1 Sentence and Aspect Representation

Each review consists of sentences, which are padded to length \( l \) by inserting padding tokens. Each review in turn is padded to length \( h \) by inserting sentences containing only padding tokens. We represent each sentence as a concatenation of its word embeddings \( x_{1:l} \) where \( x_t \in \mathbb{R}^k \) is the \( k \)-dimensional vector of the \( t \)-th word in the sentence.

Every sentence is associated with an aspect. Aspects consist of an entity and an attribute, e.g. \texttt{FOOD#QUALITY}. Similarly to the entity representation of Socher et al. (?), we represent every aspect \( a \) as the average of its entity and attribute embeddings \( \frac{1}{2}(x_e + x_a) \) where \( x_e, x_a \in \mathbb{R}^m \) are the \( m \)-dimensional entity and attribute embeddings respectively\(^1\).

3.2 LSTM

We use a Long Short-Term Memory (LSTM) (?), which adds input, output, and forget gates to a recurrent cell, which allow it to model long-range dependencies that are essential for capturing sentiment.

For the \( t \)-th word in a sentence, the LSTM takes as input the word embedding \( x_t \), the previous output \( h_{t-1} \) and cell state \( c_{t-1} \) and computes the next output \( h_t \) and cell state \( c_t \). Both \( h \) and \( c \) are initialized with zeros.

3.3 Bidirectional LSTM

Both on the review and on the sentence level, sentiment is dependent not only on preceding but also successive words and sentences. A Bidirectional LSTM (Bi-LSTM) (?) allows us to look ahead by employing a forward LSTM, which processes the sequence in chronological order, and a backward LSTM, which processes the sequence in reverse order. The output \( h_t \) at a given time step is then the concatenation of the corresponding states of the forward and backward LSTM.

\(^1\)Averaging embeddings produced slightly better results than using a separate embedding for every aspect.
3.4 Hierarchical Bidirectional LSTM

Stacking a Bi-LSTM on the review level on top of sentence-level Bi-LSTMs yields the hierarchical bidirectional LSTM (H-LSTM) in Figure 2.

The sentence-level forward and backward LSTMs receive the sentence starting with the first and last word embedding \( x_1 \) and \( x_l \) respectively. The final output \( h_l \) of both LSTMs is then concatenated with the aspect vector \( a^2 \) and fed as input into the review-level forward and backward LSTMs. The outputs of both LSTMs are concatenated and fed into a final softmax layer, which outputs a probability distribution over sentiments\(^3\) for each sentence.

4 Experiments

4.1 Datasets

For our experiments, we consider datasets in five domains (restaurants, hotels, laptops, phones, cameras) and eight languages (English, Spanish, French, Russian, Dutch, Turkish, Arabic, Chinese) from the recent SemEval-2016 Aspect-based Sentiment Analysis task (?), using the provided train/test splits. In total, there are 11 domain-language datasets containing 300-400 reviews with 1250-6000 sentences\(^4\). Each sentence is annotated with none, one, or multiple domain-specific aspects and a sentiment value for each aspect.

4.2 Training Details

Our LSTMs have one layer and an output size of 200 dimensions. We use 300-dimensional word embeddings. We use pre-trained GloVe (?) embeddings for English, while we train embeddings on frWaC\(^5\) for French and on the Leipzig Corpora Collection\(^6\) for all other languages.\(^7\) Entity and attribute em-

---

\(^2\)We experimented with other interactions, e.g. rescaling the word embeddings by their aspect similarity, an attention-like mechanism, as well as summing and multiplication, but found that simple concatenation produced the best results.

\(^3\)The sentiment classes are positive, negative, and neutral.

\(^4\)Exact dataset statistics can be seen in (?).

\(^5\)http://wacky.sslmit.unibo.it/doku.php?id=corpora

\(^6\)http://corpora2.informatik.uni-leipzig.de/download.html

\(^7\)Using 64-dimensional Polyglot embeddings (?) yielded generally worse performance.
<table>
<thead>
<tr>
<th>Language</th>
<th>Domain</th>
<th>Best</th>
<th>XRCE</th>
<th>IIT-TUDA</th>
<th>CNN</th>
<th>LSTM</th>
<th>H-LSTM</th>
<th>HP-LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Restaurants</td>
<td>88.1</td>
<td>88.1</td>
<td>86.7</td>
<td>82.1</td>
<td>81.4</td>
<td>83.0</td>
<td>85.3</td>
</tr>
<tr>
<td>Spanish</td>
<td>Restaurants</td>
<td>83.6</td>
<td>-</td>
<td>83.6</td>
<td>79.6</td>
<td>75.7</td>
<td>79.5</td>
<td>81.8</td>
</tr>
<tr>
<td>French</td>
<td>Restaurants</td>
<td>78.8</td>
<td>78.8</td>
<td>72.2</td>
<td>73.2</td>
<td>69.8</td>
<td>73.6</td>
<td>75.4</td>
</tr>
<tr>
<td>Russian</td>
<td>Restaurants</td>
<td>77.9</td>
<td>-</td>
<td>73.6</td>
<td>75.1</td>
<td>73.9</td>
<td><strong>78.1</strong></td>
<td>77.4</td>
</tr>
<tr>
<td>Dutch</td>
<td>Restaurants</td>
<td>77.8</td>
<td>-</td>
<td>77.0</td>
<td>75.0</td>
<td>73.6</td>
<td>82.2</td>
<td><strong>84.8</strong></td>
</tr>
<tr>
<td>Turkish</td>
<td>Restaurants</td>
<td>84.3</td>
<td>-</td>
<td>84.3</td>
<td>74.2</td>
<td>73.6</td>
<td>76.7</td>
<td>79.2</td>
</tr>
<tr>
<td>Arabic</td>
<td>Hotels</td>
<td>82.7</td>
<td>-</td>
<td>81.7</td>
<td>82.7</td>
<td>80.5</td>
<td>82.8</td>
<td><strong>82.9</strong></td>
</tr>
<tr>
<td>English</td>
<td>Laptops</td>
<td>82.8</td>
<td>-</td>
<td>82.8</td>
<td>78.4</td>
<td>76.0</td>
<td>77.4</td>
<td>80.1</td>
</tr>
<tr>
<td>Dutch</td>
<td>Phones</td>
<td>83.3</td>
<td>-</td>
<td>82.6</td>
<td>83.3</td>
<td>81.8</td>
<td>81.3</td>
<td><strong>83.6</strong></td>
</tr>
<tr>
<td>Chinese</td>
<td>Cameras</td>
<td>80.5</td>
<td>-</td>
<td>-</td>
<td>78.2</td>
<td>77.6</td>
<td>78.6</td>
<td>78.8</td>
</tr>
<tr>
<td>Chinese</td>
<td>Phones</td>
<td>73.3</td>
<td>-</td>
<td>-</td>
<td>72.4</td>
<td>70.3</td>
<td><strong>74.1</strong></td>
<td>73.3</td>
</tr>
</tbody>
</table>

Table 1: Results of our system with randomly initialized word embeddings (H-LSTM) and with pre-trained embeddings (HP-LSTM) for ABSA for each language and domain in comparison to the best system for each pair (Best), the best two single systems (XRCE, IIT-TUDA), a sentence-level CNN (CNN), and our sentence-level LSTM (LSTM).

Beddings of aspects have 15 dimensions and are initialized randomly. We use dropout of 0.5 after the embedding layer and after LSTM cells, a gradient clipping norm of 5, and no $l_2$ regularization.

We unroll the aspects of every sentence in the review, e.g. a sentence with two aspects occurs twice in succession, once with each aspect. We remove sentences with no aspect\(^8\) and ignore predictions for all sentences that have been added as padding to a review so as not to force our model to learn meaningless predictions, as is commonly done in sequence-to-sequence learning (\(?)\). We segment Chinese data before tokenization.

We train our model to minimize the cross-entropy loss, using stochastic gradient descent, the Adam update rule (\(?)\), mini-batches of size 10, and early stopping with a patience of 10.

### 4.3 Comparison models

We compare our model using random (H-LSTM) and pre-trained word embeddings (HP-LSTM) against the best model of the SemEval-2016 Aspect-based Sentiment Analysis task (\(?)\) for each domain-language pair (Best) as well as against the two best single models of the competition: IIT-TUDA (\(?)\), which uses large sentiment lexicons for every language, and XRCE (\(?)\), which uses a parser augmented with hand-crafted, domain-specific rules. In order to ascertain that the hierarchical nature of our model is the deciding factor, we additionally compare against the sentence-level convolutional neural network of Ruder et al. (\(?)\) (CNN) and against a sentence-level Bi-LSTM (LSTM), which is identical to the first layer of our model.\(^9\)

### 5 Results and Discussion

We present our results in Table table*1. Our hierarchical model achieves results superior to the sentence-level CNN and the sentence-level Bi-LSTM baselines for almost all domain-language pairs by taking the structure of the review into account. We highlight examples where this improves predictions in Table table2.

In addition, our model shows results competitive with the best single models of the competition, while requiring no expensive hand-crafted features or external resources, thereby demonstrating its language and domain independence. Overall, our model compares favorably to the state-of-the-art, particularly for low-resource languages, where few hand-engineered features are available. It outperforms the state-of-the-art on four and five datasets using randomly initialized and pre-trained embeddings respectively.

\(^8\)Labeling them with a **NONE** aspect and predicting **neutral** slightly decreased performance.

\(^9\)To ensure that the additional parameters do not account for the difference, we increase the number of layers and dimensions of LSTM, which does not impact the results.
Table 2: Example sentences where knowledge of other sentences in the review (not necessarily neighbors) helps to disambiguate the sentiment of the sentence in question. For the aspect in 1.1, the sentence-level LSTM predicts negative, while the context of the service and food quality in 1.2 allows the H-LSTM to predict positive. Similarly, for the aspect in 2.2, knowledge of the quality of the green tea crème brulée helps the H-LSTM to predict the correct sentiment.

5.1 Pre-trained embeddings

In line with past research (?), we observe significant gains when initializing our word vectors with pre-trained embeddings across almost all languages. Pre-trained embeddings improve our model’s performance for all languages except Russian, Arabic, and Chinese and help it achieve state-of-the-art in the Dutch phones domain. We release our pre-trained multilingual embeddings so that they may facilitate future research in multilingual sentiment analysis and text classification.

5.2 Leveraging additional information

As annotation is expensive in many real-world applications, learning from only few examples is important. Our model was designed with this goal in mind and is able to extract additional information inherent in the training data. By leveraging the structure of the review, our model is able to inform and improve its sentiment predictions as evidenced in Table table2.

The large performance differential to the state-of-the-art for the Turkish dataset where only 1104 sentences are available for training and the performance gaps for high-resource languages such as English, Spanish, and French, however, indicate the limits of an approach such as ours that only uses data available at training time.

While using pre-trained word embeddings is an effective way to mitigate this deficit, for high-resource languages, solely leveraging unsupervised language information is not enough to perform on par with approaches that make use of large external resources (?) and meticulously hand-crafted features (?).

Sentiment lexicons are a popular way to inject additional information into models for sentiment analysis. We experimented with using sentiment lexicons by Kumar et al. (?) but were not able to significantly improve upon our results with pre-trained embeddings\(^\text{11}\). In light of the diversity of domains in the context of aspect-based sentiment analysis and many other applications, domain-specific lexicons (?) are often preferred. Finding better ways to incorporate such domain-specific resources into models as well as methods to inject other forms of domain information, e.g. by constraining them with rules (?) is thus an important research avenue, which we leave for future work.

6 Conclusion

In this paper, we have presented a hierarchical model of reviews for aspect-based sentiment analysis. We demonstrate that by allowing the model to take into account the structure of the review and the sentimental context for its predictions, it is able to outperform models that only rely on sentence information and achieves performance competitive with models that leverage large external resources and hand-engineered features. Our model achieves state-of-the-art results on 5 out of 11 datasets for aspect-based sentiment analysis.

Acknowledgments

We thank the anonymous reviewers, Nicolas Pécheux, and Hugo Larochelle for their constructive feedback. This publication has emanated from research conducted with the financial support of the Irish Research Council (IRC) under Grant Number EBPPG/2014/30 and with Aylien Ltd. as Enterprise Partner as well as from research supported by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

\(^{11}\) We tried bucketing and embedding of sentiment scores as well as filtering and pooling as in (?)
References


