Sentiment Expression Boundaries in Sentiment Polarity Classification

Rasoul Kaljahi
ADAPT
School of Computing
Dublin City University
rasoul.kaljahi@adaptcentre.ie

Jennifer Foster
ADAPT
School of Computing
Dublin City University
jennifer.foster@adaptcentre.ie

Abstract

We investigate the effect of using sentiment expression boundaries in predicting sentiment polarity in aspect-level sentiment analysis. We manually annotate a freely available English sentiment polarity dataset with these boundaries and carry out a series of experiments which demonstrate that high quality sentiment expressions can boost the performance of polarity classification. Our experiments with neural architectures also show that CNN networks outperform LSTMs on this task and dataset.

1 Introduction

Sentiment analysis is a much studied problem in natural language processing research, yet it is far from solved, especially when a fine-grained analysis is required. Aspect-based sentiment analysis (Pontiki et al., 2014, 2015, 2016) is concerned with multi-faceted opinions. Consider, for example, a restaurant review which states that the food was delicious but the place was too noisy. For anyone using such a review to inform a decision about where to dine, this aspect-based information is more useful than one overall sentiment score. In this work, we aim to improve the performance of sentence-level aspect-based sentiment polarity classification. We compare several neural architectures and we investigate whether identifying and highlighting the parts of the sentence which carry the sentiment can be beneficial for this task.

The data that we use in our experiments is an English dataset used in the SemEval 2016 Task on Aspect-Based Sentiment Analysis (Pontiki et al., 2016), which consists of consumer reviews of restaurants and laptops. Each sentence is annotated with the aspect of the restaurant or laptop that is being discussed in the sentence together with the polarity of the sentiment towards that aspect. Some aspects are quite general to any product or service, e.g. value for money, but many are domain-specific, e.g. battery life and performance for laptops, and ambience and food quality for restaurants.

We first apply two very widely used neural architectures – CNNs (LeCun et al., 1998) and LSTMs (Hochreiter and Schmidhuber, 1997) – to the problem of predicting the polarity towards an aspect. We show that CNNs work better than LSTMs, and experiment with two ways of combining the two networks, neither of which provide a significant improvement over CNNs. We compare to those SemEval 2016 shared task systems which also used a neural architecture, and confirm that our systems are competitive.

Our next step is to provide a further layer of annotation to the data by marking those words in a sentence which are contributing towards the sentiment. Once the data is annotated with sentiment expressions, we use the annotation to augment our baseline models and show that this information increases polarity classification accuracy by approximately six percentage points on average. We then experiment with multi-task learning (Caruana, 1997; Collobert and Weston, 2008; Bingel and Søgaard, 2017; Ruder, 2017) in order to jointly learn sentiment expression boundaries and polarities. However, we do not see an improvement in polarity classification with our joint architecture.

The paper is organised as follows: in Section 2 we describe our data in more detail; in Section 3 we describe the architectures of our baseline systems and compare to other neural systems; in Section 4, we describe the process of enriching the original dataset with sentiment expression annotations; in Section 5, our sentiment polarity classification experiments involving this enriched dataset are presented; we review related work on senti-

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 156–166
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

D	Aspect Category	Sentence	Polarity
R	food#prices	*But the pizza is way to expensive*	Neg
R	ambience#general	*However , go for the ambience, and consider the food just a companion for a trip across the world !*	Pos
R	food#quality	*However , go for the ambience, and consider the food just a companion for a trip across the world !*	Neu
L	laptop#design_features	*Only two USB ports*	Neg
L	laptop#general	*It's a lemon*	Neg
L	laptop#general	*My first Mac computer and as many before me I just fall in love with it*	Pos

Table 1: SemEval 2016 Task 5 Dataset Examples (D: domain, R: restaurant, L: laptop)

	Laptop		Restaurant	
	Train	Test	Train	Test
# sentences	2500	808	2000	676
# aspect categories	2909	801	2507	859
% positive	56%	60%	66%	71%
% negative	37%	34%	30%	24%
% neutral	7%	6%	4%	5%

Table 2: Number of sentences, aspect categories and their polarity distributions in the datasets

ment expression annotation and the use of sentiment expressions in sentiment polarity classification in Section 6 before we summarise our findings and provide some pointers for future work in Section 7.

2 Data

The data used in our experiments consists of English consumer reviews (restaurants and laptops) released as part of the SemEval 2016 Shared Task on Aspect-Based Sentiment Analysis (Task 5, Subtask 1) (Pontiki et al., 2016). Each review sentence has been labelled with a sentiment polarity (*positive, negative, neutral*) towards an aspect of the laptop or restaurant under review (so-called *aspect categories*). Examples from the datasets are shown in Table 1. Each aspect category is of the form E#A, where E is an entity and A is some attribute of that entity. There are 11 distinct aspect categories in the restaurant dataset and 31 in the laptop dataset.

The SemEval 2016 shared task, and its previous iterations in 2014 and 2015, were concerned with several sentence-level subtasks including identifying aspect categories, opinion target expression(s)[1] and sentiment polarities. We focus on the polarity classification subtask, i.e. given a sentence and an aspect category, we attempt to predict the polarity of the sentiment expressed in the sentence towards that aspect category. Table 2 shows the number of sentences and aspect categories together with their polarity distribution in the training and test subsets of each domain. While some sentences contain multiple aspect categories (see the second and third examples in Table 1), most contain only one.

3 Neural Architecture

We build our aspect-based sentiment polarity classification systems using deep neural networks including Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) and Convolutional Neural Networks (CNN) (LeCun et al., 1998). The input layer for these systems is the concatenation of an embedding layer, which uses pre-trained *GloVe* (Pennington et al., 2014) word embeddings[2] (1.9M vocabulary *Common Crawl*), concatenated with a one-hot vector which encodes information about aspect categories. At the output layer, we use a softmax function to perform the classification into positive, negative or neutral. The middle layers are then stacks of LSTM or CNN layers, the depth of which is determined via hyper-parameter tuning. A dropout layer follows each LSTM or CNN layer to prevent

[1] Opinion target expressions are the words in the sentence which refer to the aspect category, e.g. *pizza* in the first example of Table 1. We do not make use of this information since it is only available for those sentences where the aspect category is explicitly expressed in the sentence, and is not available at all for the laptop dataset.

[2] The embedding weights are not updated during training.

	Laptop		Restaurant	
	Dev	Test	Dev	Test
lstm	77.71	75.61	80.42	83.59
cnn	78.47	77.82	79.14	84.90
chcnn+lstm	77.10	74.45	80.74	83.28
lstm+cnn	80.45	76.32	80.68	84.52
lstm+cnn(gse:aux)	86.63	83.73	86.12	89.87
lstm+cnn(gse:filtered)	84.65	83.98	86.69	87.97
lstm+cnn(gse:multitask)	78.47	75.99	81.77	84.71

Table 3: Accuracy of aspect-based sentiment polarity classification models (gse: using gold-standard sentiment expressions, filtered: filtering non-SE tokens)

System	Type	Laptop	Restaurant
Khalil and El-Beltagy (2016)	CNN	77.40	85.44
Yanase et al. (2016)	RNN	70.29	81.02
Chernyshevich (2016)	MLP	77.90	83.90
Ruder et al. (2016b)	CNN	78.40	82.10

Table 4: Test set accuracy of neural systems who participated in the SemEval 2016 polarity classification task for aspect-based sentiment analysis

the models from overfitting.[3]

To tune the hyper-parameters, a development set is randomly sub-sampled from each training set. The list of tuned hyper-parameters and their selected best values for each domain are given in Appendix A. In addition to the tuned parameters, we use the *Adam* (Kingma and Ba, 2014) algorithm for optimization. The models are built using *Keras*[4] with a *TensorFlow*[5] backend. Note that the evaluation of each architecture is performed by three rounds of training and testing, and averaging the resulting accuracy for each dataset. This is done to handle the randomness introduced throughout the model training.

3.1 LSTM vs. CNN

Due to their ability to remember information over sequences of words, LSTMs are a natural choice for many NLP tasks. Our first model uses one or more (bidirectional) LSTM layers as the middle layers between the input and output layers. Figure 1a shows the architecture of this model. The accuracy of the model on the laptop and restaurant datasets can be found in the first row of Table 3.

Convolutional networks have also shown to be useful in text classification tasks (Kim, 2014). We build another model to investigate their effect in aspect-based sentiment polarity classification. The model is displayed in Figure 1b and its performance is reported in the second row of Table 3 . Comparing the accuracy of the LSTM and CNN models, we can see that the CNN models most of the time outperform LSTM models.

3.2 Combining LSTM and CNN

LSTM and CNN networks can be combined to take advantage of the benefits of both. Ma and Hovy (2016), for example, address sequence tagging problems (POS tagging and Named Entity Recognition) using a CNN at the character level, combining the resulting character embeddings with pre-trained word embeddings and feeding it into a bidirectional LSTM network. The output is then fed into a Conditional Random Field (CRF) classifier to jointly classify labels for all words in the sentence. Their combined architecture outperforms networks built using LSTMs alone, especially on the NER task. In a similar vein, Chiu and Nichols (2016) use CNNs to generate character-level features for NER and then concatenate them with word embeddings before finally inputting them into an LSTM network. They report new state-of-the-art performance using this

[3] Note that, to save space, the dropout layers are not shown in the architecture diagrams in Fig. 1 (discussed in subsequent sections).

[4] https://keras.io/

[5] https://www.tensorflow.org/

architecture, outperforming previous models relying on extensive feature engineering and external resources.

Here, we experiment with two different approaches to combining CNN and LSTM networks. In the first approach, we first apply CNN on the character embeddings to extract the character-level representations of each word. These vectors are then concatenated with the word embeddings and sent to a LSTM. This is similar to the approach used by Ma and Hovy (2016) and Chiu and Nichols (2016), except that our problem is a regular classification one rather than sequence labelling. The architecture is depicted in Figure 1c. Note that *TimeDistributed* is a wrapper used in Keras which applies a layer, with the same weights, to every temporal step of the input. As can be seen in Table 3, the resulting model (`chcnn+lstm`) does not perform better than the individual networks, and in fact has the lowest accuracies on the laptop datasets.

In the second approach, the input data is first fed to a LSTM network and then the output representation is passed to a convolutional network. This is different from the work described above in that it does not use characters and it also places the LSTM before the CNN, in order to allow the LSTM to account for the original word order. Figure 1d shows the architecture of this network. The performance of the model trained with this architecture is shown in Table 3 (`lstm+cnn`). This model outperforms the `chcnn+lstm` one. Compared to the best individual model (`CNN`), the accuracy increases on the laptop development set, degrades on the laptop test set and stays about the same on the restaurant datasets.

3.3 Comparison to Other Systems

Table 4 shows the four neural systems who competed in the sentence-level English polarity classification subtask of the SemEval 2016 aspect-based sentiment analysis task. Two of the four systems employed a CNN, one an LSTM and one a MLP. Specifying just the network type is of course a simplification because there are many differences between the systems including the input embeddings, the hyper-parameters and the training data (some systems combined the domains in training) but it does serve to demonstrate that our neural systems achieve competitive performance. Note, however, that the best system on the restaurant domain (Brun et al., 2016), with 88.13% accuracy, and on the laptop domain (Kumar et al., 2016), with 82.77% accuracy, are non-neural systems which employed a range of linguistic information as features. Ruder et al. (2016a) also show that improvements can be achieved by not focusing just on the sentence level and taking the context sentences in the review into account also.

4 Sentiment Expressions

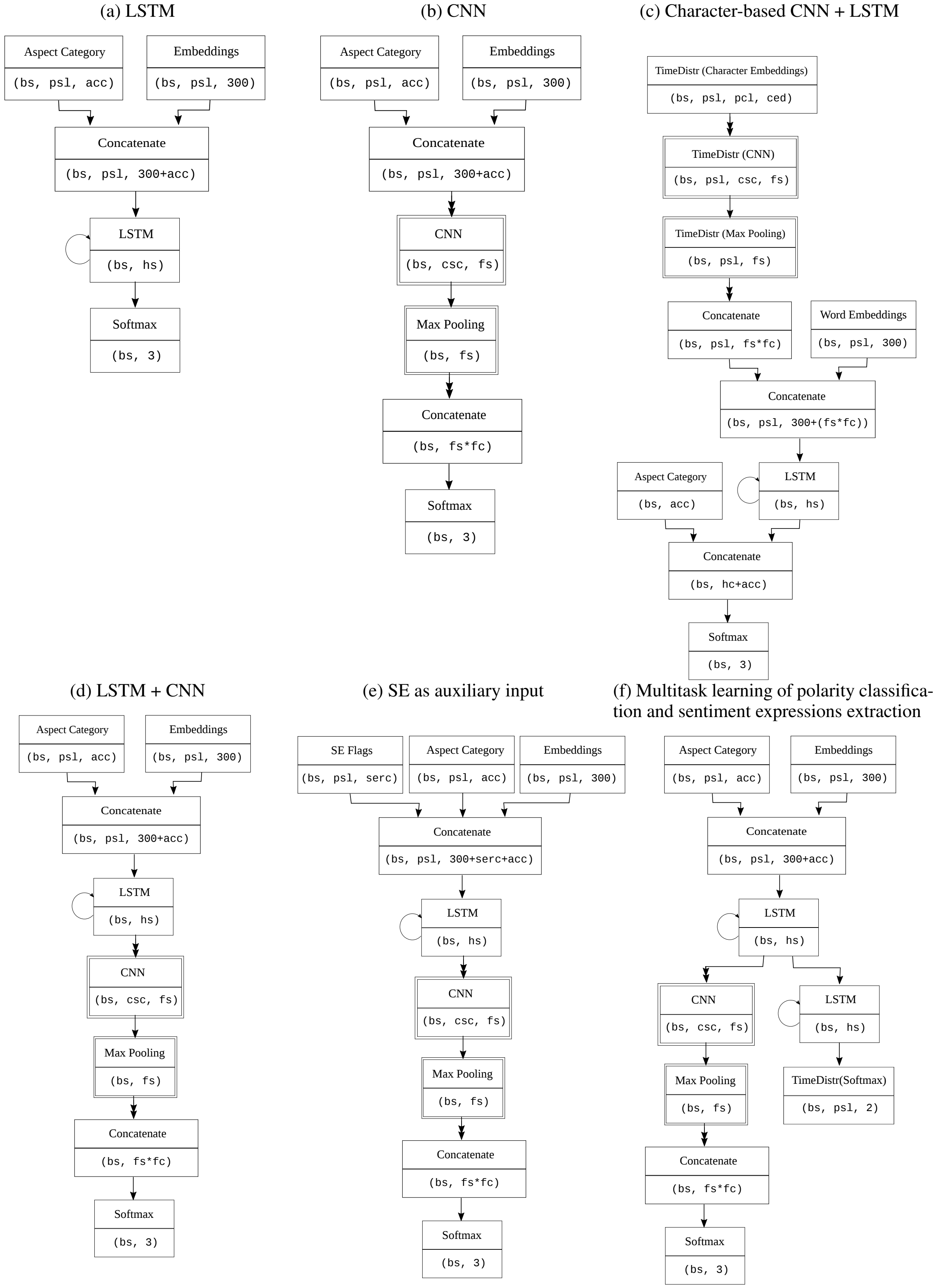
We define a sentiment expression to be the part of the sentence which conveys the sentiment towards a certain aspect of the item under discussion. Thus, annotating a `<sentence,aspect category,polarity>` triple involves highlighting those words in the sentences which express the sentiment towards the aspect category. Table 5 shows the examples from Table 1 with the sentiment expressions marked. We distinguish between neutral polarities where no opinion is expressed (1) and neutral polarities which represent a "neutral" opinion (2). Only the latter type are considered to contain sentiment expressions.

(1) *We had lunch in that restaurant last week.*

(2) *The food was* ***OK****.*

The annotation was carried out by two annotators with a background in computational linguistics, using the *brat*[6] annotation tool. In marking the sentiment expression spans, the annotators followed the general rule of thumb of being concise while at the same time respecting phrase boundaries so that the resulting sentiment expression was a self-contained, semantically coherent phrase. In order to avoid annotator disagreement over sometimes somewhat arbitrary span boundaries, rules about what to include in a span were devised and documented in the annotation guidelines. For example, any preceding articles or auxiliaries are included before sentiment-bearing nouns and verbs, e.g. *the ease of setup* versus *ease of setup*, and *is still not working* versus *still not working*. The guidelines were calibrated at three stages. At each stage, 100 items, 50 per domain, were randomly selected and annotated. The disagreements were then discussed and the guidelines were adjusted. After finalising the guidelines, the entire dataset was divided between the two annotators and annotated.

[6] `http://brat.nlplab.org`



(a) LSTM

(b) CNN

(c) Character-based CNN + LSTM

(d) LSTM + CNN

(e) SE as auxiliary input

(f) Multitask learning of polarity classification and sentiment expressions extraction

Figure 1: Architectures of the ABSA models; tuples are the shapes of the output tensors (bs: batch size, psl: padded sequence length, acc: aspect category count, hs: hidden layer size, csc: convolution step count, fs: filter size, fc: filter count, TimeDistr: distributing the layer over temporal steps of the input with the same weights, CSC: convolution step count, serc: repetition count for the SE flag (0 or 1 per token)). Note that the figures are only illustrative of the models and the middle layers are underrepresented. The double arrows show multiple input and double lines show multiple nodes.

D	Aspect Category	Sentence	Sentiment
R	`food#prices`	*But the pizza is* ***way to expensive***	Neg
R	`ambience#general`	*However ,* ***go for the ambience****, and consider the food just a companion for a trip across the world !*	Pos
R	`food#quality`	*However , go for the ambience, and* ***consider the food just a companion*** *for a trip across the world !*	Neu
L	`laptop#design_features`	***Only two USB ports***	Neg
L	`laptop#general`	*It's* ***a lemon***	Neg
L	`laptop#general`	*My first Mac computer and as many before me* ***I just fall in love with it***	Pos

Table 5: SemEval 2016 Task 5 Dataset Examples with Sentiment Expressions (in bold)

Laptop			Restaurant		
P	R	$F1$	P	R	$F1$
81.63	91.58	86.32	89.67	91.61	90.63

Table 6: Inter-annotator agreement of sentiment expression annotation measured using precision, recall and F1 of sentiment expression span intersections

Inter-annotator agreement was calculated on a subset of 200 items, 100 per domain, where 100 items from each annotator's subset was also annotated by the other annotator. We used precision, recall and F_1 based on the intersection of the sentiment expression spans annotated by the two annotators, assuming the first annotator's annotations as gold-standard and the second annotator's as predicted.

The IAA scores, shown in Table 6, show a high level of agreement between the two annotators. The agreement on the restaurant dataset is particularly high, suggesting that the restaurant reviews use more straightforward language than the laptop reviews (also reflected in the polarity classification results - see Section 3). Examining the doubly annotated data, we see that, most of the time, the disputed annotations overlap and the disagreement is over how long the sentiment expression span should be. In fact, 122 out of the 200 samples in the laptop dataset and 142 out of 200 in the restaurant dataset were annotated in exactly the same way by the two annotators.

An example disagreement can be seen in (3) and (4). While the first annotator has decided that *No more Apple devices* is enough to infer the negative sentiment from the sentence, the second annotator deems *in my household* to be also contributing. With a binary overlap metric (correct for any overlap; wrong for no overlap), as used, for example, by Breck et al. (2007) to evaluate expression extraction, this example would have a perfect precision and recall score.

(3) *No more Apple devices in my household.*

(4) *No more Apple devices in my household.*

Concluding that the sentiment expressions have been marked with a reasonable level of consistency, we now go on to use these expression boundaries in more sentiment polarity classification experiments.

5 Using Sentiment Expressions in Polarity Classification

We conduct experiments to examine the degree to which sentiment expressions can help boost polarity classification performance. We first measure the upper bound of the improved performance using gold-standard sentiment expressions and experiment with two alternative ways of encoding the sentiment expression information. We then attempt to use the sentiment expression annotation in a multi-task setup, with the sentiment expression extraction as an auxiliary task and the polarity classification as the main one. For all our experiments we employ the combined LSTM/CNN architecture described in Section 3.2 – `lstm+cnn` in Table 3.

5.1 Using Gold-standard Sentiment Expressions

To exploit sentiment expressions in polarity classification, we experiment with two approaches. In the first approach, the sentiment expression is fed into the model as an auxiliary input, in concatenation with the embeddings and aspect categories. The architecture of this approach is illustrated in Figure 1e. The sentiment expression annotation of a sentence is encoded in a binary-valued vector of size equal to the sentence length. For every token inside the SE boundary, the binary value is 1 and 0 otherwise. In order to give more weight to this information, the vector is vertically replicated n times (seen as `serc` in Figure 1e) to form a matrix. Table 3 shows the performance of this approach (`lstm+cnn(gse:aux)`). It can be seen that the sentiment expressions consistently boost the performance of polarity classification over all four datasets, with an average improvement of six percentage points.

The second approach works by filtering out the non-sentiment-expression tokens in the sentence. In other words, the input to the model is the sequence of sentiment expression tokens. The architecture of this model is the same as that of the combined LSTM and CNN depicted in Figure 1d, since only the input has changed from the entire sentence to filtered tokens. The results for this approach are also shown in Table 3 (`lstm+cnn(gse:filtered)`). According to the accuracy scores, using sentiment expressions as auxiliary input is preferable to this filtering approach, as the latter obtains significantly lower scores on two of the evaluation subsets, suggesting that the sentiment expressions themselves do not carry all the information relevant to the task.

Overall, both sets of results show that knowing which words (if any) in the sentence are part of the sentiment expression is a valuable source of information for polarity classification, which is what one might expect. The SE-augmented models are better at handling the neutral cases, appearing to learn to associate a lack of sentiment expressions with this category. They are also better at handling negative cases, particularly in the restaurant dataset where the sentiment expression information helps the system to move away from the majority positive class (see the polarity class distribution in Table 2).

5.2 Multitask Learning of Polarity and SE

One way to utilize the sentiment expressions in polarity classification is multitask learning of polarity classification and sentiment expression extraction (Caruana, 1997). The idea is that sharing representations between related tasks can help each of them generalize better. Collobert and Weston (2008) build a unified architecture to simultaneously learn several NLP problems including part-of-speech tagging, chunking, named entity recognition, semantic role labelling (SRL), semantic relatedness detection and language modelling. Their model consists of convolutional networks on top of a shared embedding layer along with individual embedding layers for each task. They try various combinations of these tasks and show that, SRL, for example can benefit from other tasks such as language modelling to achieve state-of-the-art performance without the need for syntactic information as is common in conventional SRL models.

We further investigate this approach by designing a model that learns both tasks at the same time, meaning that the two objective functions are optimized simultaneously, but the main focus is learning the polarity classification and the sentiment expression extraction is an auxiliary task. The architecture of this model is displayed in Figure 1f. The model is built by combining LSTM and CNN, where the input is first fed into a shared LSTM layer between the two tasks. The output of the LSTM layer is then sent to a CNN layer which learns the polarity classification and to another LSTM layer which learns to extract sentiment expressions. The sentiment expressions in the output are represented in a vector of length 2 for each token, where being inside the SE is encoded by $[1, 0]$ and being outside by $[1, 0]$. The prediction for both tasks is achieved using a softmax layer at the end.

The results of this approach are shown in Table 3 (`lstm+cnn(gse:multitask)`). The multitask approach fails to reach the level of performance of the systems which uses gold sentiment expression boundaries as auxiliary input (`lstm+cnn(gse:aux)`) or to filter the original input (`lstm+cnn(gse:filtered)`), especially apparent on the laptop dataset. It should however be noted that multitask learning eliminates the need for gold sentiment expression boundaries at prediction time, so its comparison is more meaningful with systems that use automatically obtained sentiment expression boundaries.

6 Related Work

To the best of our knowledge, the only other dataset containing manual annotations of opinion expressions is MPQA (Wiebe et al., 2005). MPQA annotates *private states* (Quirk et al., 1985), which is a general term covering opinions, evaluations, emotions and speculations. The annotations are categorized into two types: *direct subjective expressions* and *expressive subjective expressions*, the former mentioning the opinions explicitly and the latter implicitly. For example, in (5), *said* is a direct subjective and *full of absurdities* is an expressive subjective expression.

(5) *"The report is full of absurdities," Xirao-Nima said.*

Our annotation scheme does not differentiate between these two types, instead aiming at a simpler guideline for annotating sentiment expressions, where the main rule is to find the part of the sentence which independently carries the sentiment. Therefore, *said* in (5) would be ignored in our annotation as it does not help recognize the sentiment towards *The report*. Instead, *full of absurdities* would be annotated as the sentiment expression towards *The report*, as it is clearly the source negative sentiment expressed by the speaker (*Xirao-Nima*). Therefore, our definition of sentiment expression is closer to the MPQA's expressive subjective expression.

A related work in terms of utilizing opinion expressions for other opinion mining tasks is (Johansson and Moschitti, 2013), who use features extracted from MPQA opinion expressions in product attribute identification (i.e. finding sentiment targets) and also document polarity classification. The features used in the second task – which is more relevant to this work – include the individual opinion expression words combined with the polarity or type of the expressions. Their results show that information extracted from opinion expressions can help improve polarity classification compared to when only bag-of-word features and sentiment polarity lexicons are used. Using a different dataset, a different type of opinion expression and a different way of encoding this knowledge (by marking expression *boundaries*), we provide further evidence that isolating the opinion expression in an utterance helps in polarity classification.

7 Conclusion

A major contribution of the paper is an additional set of manual annotations in the English SemEval 2016 Task 5 dataset in which those words in a sentence which are contributing towards the expression of sentiment towards a particular aspect are explicitly marked. In experiments with this dataset, we demonstrate that knowledge of the boundaries of sentiment expressions can simplify the task of polarity classification. This knowledge seems to have the effect of reducing noise for the learner by de-emphasizing words in the input that are not contributing towards the sentiment and providing clues about how subjective a sentence is.

Although the results of our multitasking experiments were somewhat disappointing, our experiments with gold sentiment expressions motivate us to continue exploring ways of using sentiment expressions in polarity classification. A pipeline approach in which the sentiment expression extraction is carried out before polarity classification is also possible, and indeed several sentiment expression extraction systems have been built with the MPQA dataset (Breck et al., 2007; Choi and Cardie, 2010; Yang and Cardie, 2012; İrsoy and Cardie, 2014). We plan to build a sentiment expression extraction system using our new set of annotations and then investigate the effect of substituting gold sentiment expressions with automatically predicted sentiment expressions.

The dataset reported in this work is available for use by other researchers, as a source of train/test data for sentiment expression extraction or joint polarity classification/sentiment expression extraction, as well as a potential source of linguistic insights about expressions of sentiment in this type of text.[7]

In addition to our sentiment expression data and experiments, we have also compared the use of LSTMs and CNNs and their combination for English aspect-based sentiment polarity classification with the SemEval 2016 Task 5 dataset, concluding that CNNs on their own or in combination with an LSTM are a good choice.

Acknowledgments

This research is supported by Science Foundation Ireland in the ADAPT Centre (Grant 13/RC/2106) (www.adaptcentre.ie) at Dublin City University

[7] https://opengogs.adaptcentre.ie/rszk/sea

References

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv preprint arXiv:1702.08303*.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 2683–2688.

Caroline Brun, Julien Perez, and Claude Roux. 2016. Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 277–281.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Maryna Chernyshevich. 2016. Ihs-rd-belarus at semeval-2016 task 5: Detecting sentiment polarity using the heatmap of sentence. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 296–300. Association for Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 269–274.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

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

Ozan İrsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 720–728.

Richard Johansson and Alessandro Moschitti. 2013. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3):473–509.

Talaat Khalil and Samhaa R. El-Beltagy. 2016. Niletmrg at semeval-2016 task 5: Deep convolutional neural networks for aspect category and sentiment extraction. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 271–276.

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. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ayush Kumar, Sarah Kohail, Amit Kumar, Asif Ekbal, and Chris Biemann. 2016. Iit-tuda at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 1129–1135.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

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

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Randolph Quirk, Charles Ewart Eckersley, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016a. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 999–1005. Association for Computational Linguistics.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016b. Insight-1 at semeval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis. In *SemEval@NAACL-HLT*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 1(2):0.

Toshihiko Yanase, Kohsuke Yanai, Misa Sato, Toshinori Miyoshi, and Yoshiki Niwa. 2016. bunji at semeval-2016 task 5: Neural and syntactic models of entity-attribute relationship for aspect-based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 289–295. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-markov conditional random fields. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1335–1345.

A Model Hyper-parameters

	direction	epochs	batch size	#LSTM layers	LSTM layer size	CNN filter sizes	#CNN filters	learning rate	activation	dropout rate	char. embed. dim.
`lstm`	bi	100	128	1	50	-	-	0.01	tanh	0.3	-
`cnn`	-	200	64	-	-	[2,3,4]	128	0.05	tanh	0.5	-
`lstm+cnn`	bi	100	32	1	50	[3,4,5]	256	0.001	tanh	0.5	-
`chcnn+lstm`	bi	50	32	2	50	[3]	64	0.001	tanh	0.3	50
`lstm+cnn(gse:aux)`	bi	100	32	1	100	[2,3,4]	256	0.001	tanh	0.5	-
`lstm+cnn(gse:filtered)`	bi	100	32	2	100	[3,4,5]	128	0.001	tanh	0.3	-
`lstm+cnn(gse:multitask)`	uni	100	32	2	100	[3,4,5]	64	0.001	tanh	0.3	-

Hyper-parameters of the polarity classification models tuned on the laptop development set

	direction	epochs	batch size	#LSTM layers	LSTM layer size	CNN filter sizes	#CNN filters	learning rate	activation	dropout rate	char. embed. dim.
`lstm`	bi	100	128	1	100	-	-	0.005	tanh	0.3	-
`cnn`	-	200	32	-	-	[2,3,4]	64	0.001	tanh	0.3	-
`lstm+cnn`	bi	100	64	1	100	[2,3,4]	64	0.001	tanh	0.3	-
`chcnn+lstm`	bi	50	64	2	100	[3]	32	0.001	tanh	0.3	20
`lstm+cnn(gse:aux)`	bi	100	32	1	50	[2,3,4]	128	0.001	tanh	0.5	-
`lstm+cnn(gse:filtered)`	bi	100	64	1	100	[3,4,5]	128	0.001	tanh	0.5	-
`lstm+cnn(gse:multitask)`	uni	100	32	2	50	[3,4,5]	128	0.001	tanh	0.3	-

Hyper-parameters of the polarity classification models tuned on the restaurant development set