Mining fine-grained opinions on closed captions of YouTube videos with an attention-RNN

Edison Marrese-Taylor, Jorge A. Balazs, Yutaka Matsuo
Graduate School of Engineering
The University of Tokyo
Tokyo, Japan
emarrese,jorge,matsuo@weblab.t.u-tokyo.ac.jp

Abstract

Video reviews are the natural evolution of written product reviews. In this paper we target this phenomenon and introduce the first dataset created from closed captions of YouTube product review videos as well as a new attention-RNN model for aspect extraction and joint aspect extraction and sentiment classification. Our model provides state-of-the-art performance on aspect extraction without requiring the usage of hand-crafted features on the SemEval ABSA corpus, while it outperforms the baseline on the joint task. In our dataset, the attention-RNN model outperforms the baseline for both tasks, but we observe important performance drops for all models in comparison to SemEval. These results, as well as further experiments on domain adaptation for aspect extraction, suggest that differences between speech and written text, which have been discussed extensively in the literature, also extend to the domain of product reviews, where they are relevant for fine-grained opinion mining.

1 Introduction

On-line videos have become indispensable to people's daily lives, as traffic statistics showed that by 2010 it accounted for 56.6% of the total global consumer traffic (Siersdorfer et al., 2010). Studies support the notion that on-line reviews can have a strong influence in the decision-making of potential Internet buyers (Chevalier and Mayzlin, 2006), thus becoming a major factor for both consumers and marketers (Hu et al., 2008).

Video reviews are the natural evolution of written product reviews. In fact, people are increasingly turning to platforms such as YouTube to help them shop, looking for product reviews (Lawson, 2015). YouTube unboxing videos have become a growing phenomenon (Lawson, 2015; Insights, 2014). In 2015 alone, people in the U.S. watched 60M hours of them on YouTube, totaling 1.1 B views. The same year, views of product review videos increased by 40% compared to 2014, and more than 1 million channels related to product reviews were counted (Baysinger, 2015). Despite all of this, the most widely used approaches in opinion mining focus only on tweets or written product reviews available on websites like Amazon.

Therefore, in this paper we present the first opinion mining study focusing on *video product reviews*. We take the fine-grained approach, which aims to detect the subjective expressions in text and to characterize their sentiment orientation, and analyze the closed captions of *video product reviews* extracted from YouTube. Fine-grained opinion mining is important for a variety of NLP problems, including opinion-oriented question answering and opinion summarization, having been studied extensively in recent years. In practical terms, this approach defines the tasks of aspect extraction (*AE*), sentiment classification (*SC*) and a joint setting (*AESC*).

While *AE* and *AESC* have often been tackled as sequence labeling problem, where the sentence is a stream of tokens to be labeled using IOB and collapsed or sentiment-bearing IOB labels (Zhang et al., 2015) respectively, *SC* can be regarded as a semantic compositional problem, where the obtained representation is used to predict the sentiment.

Accounting for the patent differences between speech and written text, which have also led linguists to consider them as different domains (Biber, 1991) exhibiting different syntactic (O'Donnell, 1974) and distributional properties, we created the first annotated dataset using closed

Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 102–111
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

captions of YouTube product review videos, which we named the *Youtubean* dataset.

Motivated by the success of attention-based approaches in multiple NLP problems such as machine translation (Bahdanau et al., 2015), parsing (Vinyals et al., 2015), slot-filling (Liu and Lane, 2016) and others (Luong et al., 2015), we also introduce an attention-augmented RNN model for *AE* and *AESC*. Compared to previous work, the attentional component makes our model specially suitable for *AESC*, since it directly addresses the compositional nature of the sentiment classification task as it allows the model to represent the input sentence as a convex combination of word representations. This is confirmed by our results on the SemEval ABSA dataset (Pontiki et al., 2014), given that our model offers state-of-the-art performance for *AESC* while also performing equivalently to the state-of-the-art for aspect extraction without the need for manually-crafted features.

We also show that our attention-RNN model outperforms the baseline for both *AE* and *AESC* on our dataset. However, we observed that compared to the SemEval corpora, all the tested models decreased their performance on it. As indicated by a descriptive analysis of our corpus and by additional experiments using domain adaptation techniques for *AE*, which did not offer considerable gains, our results seem to support the existence of the aforementioned differences between speech and written text in the context of product reviews and their importance for fine-trained opinion mining. Our code and data are available for download on GitHub[1].

2 Related Work

Our work is related to aspect extraction using deep learning, a task that is often tackled as a sequence labeling problem. In particular, our work is related to Irsoy and Cardie (2014), who pioneered in the field by using multi-layered RNNs on a subset of the MPQA 1.2 dataset (Wiebe et al., 2005). Later, Liu et al. (2015) successfully adapted the architectures by Mesnil et al. (2013), experimenting on the SemEval 2014 dataset (Pontiki et al., 2014). Compared to these, our model is novel since it introduces the usage of attention for *AE*. In this sense, our work is also related to Liu and Lane (2016), who introduced an attention RNN for slot-filling in Natural Language Understanding.

We also find related work on the usage of RNNs for open domain targeted sentiment (Mitchell et al., 2013), where Zhang et al. (2015) experimented with neural CRF models using various RNN architectures on a dataset of informal language from Twitter. In our case, the domain is different since we focus on product reviews.

Regarding target-based sentiment analysis, we find several ad-hoc models that account for the sentence structure and the position of the aspect on it, such as Tang et al. (2016b) and Tang et al. (2016a), who use attention-augmented RNNs for the task. However, these models require the location of the aspect to be known in advance and therefore are only useful in pipeline models. Our work is similar to these since it also makes use of an attentional component to model compositionally in sentiment classification, but we model aspect extraction and sentiment classification as a joint task instead of using a pipeline approach.

AESC has also often been tackled as a sequence labeling problem, mainly using CRFs (Mitchell et al., 2013). To model the problem in this fashion, collapsed or sentiment-bearing IOB labels (Zhang et al., 2015) are used. Pipeline models (i.e. task-independent model ensembles) have also been extensively studied by the same authors. We also find Xu et al. (2014) who performed *AESC* by modeling the linking relation between aspects and the sentiment-bearing phrases.

When it comes to the video review domain, we find related work on YouTube mining, mainly focused on exploiting user comments. For example, Wu et al. (2014) exploited crowdsourced texual data from time-synced commented videos, proposing a temporal topic model based on LDA. However, Schultes et al. (2013) showed that comments with references to video content[2] represent only 2% to 4% of comments in YouTube. Therefore, we think this kind of analysis might be limited. The work of Tahara et al. (2010) introduced a similar approach for *Nico Nico* using time-indexed social annotations to search for desirable scenes inside videos.

On the other hand, Severyn et al. (2014) proposed a systematic approach to mine user comments that relies on tree kernel models. Additionally, Krishna et al. (2013) performed sentiment analysis on YouTube comments related to popular topics using machine learning techniques, show-

[1] github.com/epochx/opinatt

[2] Class C7 in the paper

Video title	Video id	Length	# of sentences
Sprint Samsung Galaxy S5 Full Review!	jdzbw68mpZE	10:23	97
Samsung Galaxy S5 Review	zV0u2UFwv6E	12:07	147
Samsung Galaxy S5 Review - Phones 4u	1lxAO_YgZ98	5:07	41
Samsung Galaxy S5 Review	_Ihe7jm63kU	3:49	45
Samsung Galaxy S5 “Special ”Review & Camera Samples	nayKYv_7b6M	12:00	52
Samsung Galaxy S5 vs Apple iPhone 5s: Which Is Better?	1dvzHyHID0k	3:34	32
Samsung Galaxy S5 review	bRv5JrKnp3M	24:15	164

Table 1: Detail of the reviews used to create the *Youtubean* dataset.

ing that the trends in users' sentiments is well correlated to the corresponding real-world events. Siersdorfer et al. (2010) presented an analysis of dependencies between comments and comment ratings, proving that community feedback in combination with term features in comments can be used for automatically determining the community acceptance of comments.

Finally, we find some papers that have successfully attempted to use closed caption mining for video activity recognition (Gupta and Mooney, 2010) and scene segmentation (Gupta and Mooney, 2009). Similar work has been done using closed captions to classify movies by genre (Brezeale and Cook, 2006) and summarize video programs (Brezeale and Cook, 2006).

3 Dataset

In YouTube, video authors con provide their own closed captions, or they can be generated automatically by the engine. In both cases, these captions can be interpreted as a time-indexed transcript of the speech in the video. Therefore, to minimize the amount of noise in the data, we utilized the user-provided closed captions of seven of the most popular reviews of the Samsung Galaxy S5 and creatd an annotated dataset for fine-grained opinion mining. We obtained, cleaned and processed the data, and annotated the aspects following the guidelines by Pontiki et al. (2014) using *brat*[3] (Stenetorp et al., 2012). We divided the annotation process into two steps.

First, two different annotators tagged aspects independently, obtaining an exact inter-annotation agreement of 0.705 F1-score. This value rose to 0.823 when allowing for partial matches, which we defined as any overlap between the annotated terms. Discrepancies were discussed until a final setting was reached.

With these annotations fixed, we asked the same annotators to tag the sentiment of each extracted aspect. On this task, the annotators obtained an average agreement of 0.942 F1-score. This time, discrepancies were discussed with a third person who acted as an arbiter, until an agreement was reached. Both aspect extraction and sentiment classification inter-annotator agreements are comparable to the values obtained in similar tasks (Jimenez-Zafra et al., 2015) (Wiebe et al., 2005).

Corpus	R	L	Y
# Sentences	3041	3045	578
# Aspects	1288	1042	525
Mean word/sentence	15.47	16.76	20.71
Mean const. tree depth	9.10	10.16	11.40
Mean word/aspect	1.97	1.83	2.14
Mean aspects/sentence	1.20	0.76	1.38
Sentences with aspects	66.46%	48.87%	66.96%

Table 2: Descriptive corpora comparison.

Table 1 provides some key information about the the source video reviews we have used to build our dataset, which we named the *Youtubean* dataset. Table 2 compares it to the SemEval Laptops and Restaurants corpora, regarded as the de facto datasets for written review mining. Several differences can be observed. A big distinction lies in mean sentence and aspect lengths, both of which are considerably longer in *Youtubean*. We also analyzed sentence syntax complexity in terms of the constituency tree depth, observing that our sentence trees are deeper on average. Furthermore, *Youtubean* exhibits both longer and more frequent aspect mentions.

4 Proposed Model

Our proposed model is a two-pass bidirectional RNN architecture that includes an attentional component. Formally, given an embedded input sequence $x = [x_1, ..., x_n]$ with one-hot encoded labels $y = [y_1, ..., y_n]$, we define the first pass as follows.

$$\bar{x_i} = [x_{i-d}; ...; x_i; ...; x_{i+d}] \quad (1)$$

$$\vec{h}_i = \sigma(\bar{x_i}, \vec{h}_{i-1}) \quad (2)$$

[3] http://brat.nlplab.org/

$$\overleftarrow{h}_i = \sigma(\bar{x}_i, \overleftarrow{h}_{i+1}) \quad (3)$$

$$h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i] \quad (4)$$

Where σ denotes the sigmoid nonlinearity, $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ are the forward and backward hidden states of the RNN, which are concatenated, and $\bar{x}_i$ is a context window of ordered word embedding vectors around position i, with a total size of $2d + 1$. This context window is intended to improve the model capabilities to capture short-term temporal dependencies (Mesnil et al., 2013).

The second pass goes through the hidden states h_i and performs sequence labeling token by token. We use the attentional decoder from (Vinyals et al., 2015).

$$u_{i,j} = v^{\top} \tanh(W_{\alpha}[h_i; h_j]) \quad (5)$$

$$\alpha_{i,j} = softmax(u_{i,j}) \quad (6)$$

$$t_i = \sum_{j=1}^{n} \alpha_{i,j} \cdot h_j \quad (7)$$

$$\hat{y}_i = softmax(W_s[h_i; t_i; y_{i-1}]) \quad (8)$$

Where $\hat{y}_i$ is a probability distribution over the label vocabulary for input i. As shown, this is obtained using both the corresponding *aligned* input h_i and the attention distribution over all hidden states t_i, i.e. using a global attention scheme (Luong et al., 2015). While generating the output $\hat{y}_i$, we explicitly model the dependency on the previous label by adding y_{i-1} to the computation. These two components are combined using a feed forward neural network, whose output dimension is the size of the tag label vocabulary for *AE* or *AESC*. To initialize the attention matrix h_n is used so the model does not bypass it. As a loss function we use the mini-batch average cross-entropy.

The addition of the attentional component to our model is motivated by two factors. In the first place, in contrast to Mesnil et al. (2013) who directly make use of a window of previous hidden states for AE, the attentional components allows us to access contextual information in a more natural and selective way. For AESC, the attention directly models sentiment compositionality.

5 Experimental setup

For our experiments, in addition to *Youtubean*, we also worked with the SemEval ABSA 2014 Laptops and Restaurants corpora (Pontiki et al., 2014), which can be regarded as the de facto datasets for fine-grained review mining. For *AE* we use the train/test splits provided for Phase B. For *AESC*, since the test data does not have sentiment labels, we worked only with the training data. On the other hand, since the size of *Youtubean* is smaller than the SemEval corpora, we used 5-fold cross validation to make results more robust. For each fold, we used 10% of the development data as a validation set and compare our results using two-sided t-tests.

For evaluation, we used the CoNLL *conlleval* script for evaluation based on F1-score. To perform joint aspect extraction and sentiment classification, we only considered *positive* (+), *negative* (−) and *neutral* (0) as sentiment classes, and the additional *conflict* class is mapped to *neutral*. To gain insights on the output of the models for *AESC*, we decoupled the IOB collapsed tags using simple heuristics to recover the *simple* aspect extraction F1-score as well as classification performances for each sentiment class, but we used the *joint* tagging *conlleval* F1-score to evaluate the models.

As a baseline, we implemented the RNN architectures by Liu et al. (2015), which are the state-of-the-art in fine-grained aspect extraction. We experimented with Jordan-style RNNs (JRNN), Elman-style RNNs (RNN), LSTMs and the bidirectional versions of these last two. We followed Irsoy and Cardie (2014) to merge the forward and backward hidden states, setting $y_t = \sigma(\overrightarrow{U}\overrightarrow{h}_t + \overleftarrow{U}\overleftarrow{h}_t)$, where $\overrightarrow{U}$, $\overleftarrow{U}$ are output matrices for the forward and backward hidden states $\overrightarrow{h}_t$, $\overleftarrow{h}_t$, respectively. This gives the models more flexibility to capture complex relations in a sentence, making them able to learn how to weight future and past information.

For both our attention-RNN model and the baseline RNNs, we experimented with Senna embeddings (Collobert et al., 2011), GoogleNews embeddings (Mikolov et al., 2013) and WikiDeps (Levy and Goldberg, 2014). The usefulness of working with pre-trained embeddings for the baseline RNNs was already shown by (Liu et al., 2015). However, for comparison when experimenting with our model, we also used randomly initialized embeddings of sizes 50 and 300 to test this hypothesis.

To make our results more transparent, we explicitly experimented with two different preprocessing pipelines. We used Senna (Collobert et al., 2011), which provides both a POS-tagger and a chunker, and CoreNLP (Manning et al., 2014). The latter lacks a chunker so we combined it with the CoNLL *chunklink* script[4]. As Liu et al. (2015), we also experimented adding the same 14 linguistic binary features they used, which are based on POS-tags and chunk IOB-tags. These are concatenated to the hidden layer of the RNN before the final output non-linearity.

To train our baseline models we set a learning rate of 0.01 with decay and early stopping on the validation set. We set a fixed window size of 1 for bi-directional and 3 for unidirectional models, and always train word embeddings. Exploratory experiments showed that most models stop learning after a few epochs —3 or 4— so we only trained for a maximum of 5 epochs.

In the case of our attention-RNN model (ARNN), here we only report results using LSTMs, which outperformed all others cells we tried on preliminary experiments. We explored different hyper-parameter configurations, including context window sizes of 1, 3 and 5 as well as hidden state sizes of 100, 200 and 300, and dropout keep probabilities of 0.5 and 0.8. We also experimented concatenating the RNN hidden states after the first pass with the binary features used by (Liu et al., 2015). Finally, we also experimented with unidirectional versions of the RNNs. For training, we used mini-batch stochastic gradient descent with a mini-batch size of 16 and padded sequences to a maximum size of 200 tokens. We used exponential decay of ratio 0.9 and early stopping on the validation when there was no improvement in the F1-score after 1000 training steps.

6 Results

6.1 Aspect Extraction (*AE*)

6.1.1 Laptops

Table 3 summarizes our best baseline results on the Laptops datasets. For contrast we include the best F1-scores obtained by Liu et al. (2015) (cf. F1* columns). We observed the CoreNLP pipeline outperformed the Senna pipeline, with an average absolute gain of 2.105%, significant at $p = 1.29 \times 10^{-5}$, and binary features proved useful offering average absolute gains of 1.538% ($p = 1.29 \times 10^{-5}$). Finally, note that the best configurations always use SennaEmbeddings, which outperformed others significantly for each case.

Model	Emb.	$\|h\|$	feat	F1	F1*
JRNN	Senna	50	No	70.81	73.42
LSTM	Senna	100	Yes	70.92	**75.00**
BiLSTM	Senna	50	Yes	69.03	74.03
RNN	Senna	50	No	**71.87**	74.43
BiRNN	Senna	50	Yes	69.45	74.57

Table 3: Results of our implemented baseline RNN models on the Laptops dataset.

Table 4 summarizes the best results of our ARNN model on the Laptops dataset, where we obtained a maximum F1-score of 74.74. Again, the CoreNLP pipeline significantly outperformed Senna, with an average absolute gain of 1.39 ($p = 3.4 \times 10^{-33}$) F1-score. Bidirectionality provided an absolute average gain of 1.15 F1-score ($p = 4.61 \times 10^{-}20$).

Both SennaEmbeddings and GoogleNews provided statistically equivalent results ($p = 0.65$), which were also significantly superior to WikiDeps with p-values 9.54×10^{17} and 2.6×10^{-13} respectively. Pre-trained embeddings outperformed random embeddings on average, comparing across same-sized cases. Linguistic binary features did not statistically contribute to the performance.

Embeddings	|d|	|cw|	|h|	F1
SennaEmbeddings	50	1	100	**74.74**
Random	50	3	300	70.19
WikiDeps	300	3	200	69.53
GoogleNews	300	3	100	71.17
Random	300	3	200	70.03

Table 4: Best results for our ARNN on *AE* for the Laptops dataset.

6.1.2 Restaurants

Table 5 summarizes our best baseline results for the Restaurants dataset, again for contrast we include the best F1-scores obtained by Liu et al. (2015) (cf. F1* columns).

Regarding the usage of the linguistic features, we found that they contributed to increasing performance with an average absolute gain of 1.083% ($p = 1.65 \times 10^{-6}$). This is consistent with previous findings by Liu et al. (2015). The Senna pipeline outperformed CoreNLP with an average absolute gain of 1.161% ($p = 1.02 \times 10^{-8}$). Embeddings caused statistically significant differences, where WikiDeps outperformed both other embeddings on average.

[4] http://ilk.uvt.nl/team/sabine/homepage/software.html

Model	Emb.	$\|h\|$	feat	F1	F1*
JRNN	WDeps	100	Yes	78.20	79.89
LSTM	WDeps	100	Yes	**78.97**	81.37
BiLSTM	WDeps	200	Yes	74.73	81.06
RNN	Senna	200	Yes	77.13	81.66
BiRNN	WDeps	100	No	74.33	**82.06**

Table 5: Results of our implemented baseline RNN models on the Restaurants dataset.

Table 6 summarizes the best results by our ARNN model on the Restaurants dataset, where we obtained a maximum F1-score of 81.83. All of our best performing models use a bidirectional architecture. In fact, bidirectionality provided an average significant absolute gain of 0.89 F1-score ($p = 1.25 \times 10^{-17}$). Additionally, using CoreNLP as preprocessing pipeline provided an average gain of 0.585 F1-score ($p = 2.98 \times 10^{-21}$) over Senna.

Embeddings	|d|	|cw|	|h|	F1
SennaEmbeddings	50	1	100	**81.83**
Random	50	3	100	78.79
WikiDeps	300	3	100	78.68
GoogleNews	300	3	300	78.73
Random	300	1	100	78.38

Table 6: Best results for our attention-RNNs on *AE* on the Restaurants dataset.

Context windows proved beneficial as confirmed by the significantly different average F1-scores of 76.55, 77.59 and 77.28 for window sizes 1, 3 and 5 respectively. We also observed significant performance differences using SennaEmbeddings, which outperformed all others with an average F1-score of 77.94. GoogleNews and WikiDeps exhibited average F1-scores of 76.93 and 76.55, which are statistically different ($p = 4.08 \times 10^{-6}$) and although they also outperformed random embeddings for $d = 300$, they performed statistically worse than random embeddings for $d = 50$. Linguistic binary features did not statistically contribute to the performance.

6.1.3 *Youtubean*

Table 7 summarizes our results for baseline RNNs on *Youtubean*. Again, we observed that adding linguistic features had a positive effect on the performance, with an average absolute gain of 1.30% ($p = 0.01$). SennaEmbeddings and WikiDeps provided better performance compared to GoogleNews, with average F1-scores of 49.11, 49.64 and 45.37 respectively. The first two values were statistically indistinguishable. We could not observe significant differences in the performance for different pipelines.

RNN	Pipeline	Emb.	Feat.	|h|	F1
RNN	Senna	WDeps	Yes	100	55.82*
RNN	CoreNLP	WDeps	No	200	55.69*
LSTM	CoreNLP	Senna	No	100	**56.13**
BiRNN	CoreNLP	WDeps	No	200	50.15
BiLSTM	Senna	Senna	Yes	100	50.09

Table 7: Results of our implemented baseline RNN models on *AE* for the *Youtubean* dataset.

To further study the relation between written and video product reviews for aspect extraction, a task that has been broadly studied by our community, we complemented our RNNs baseline with two classic domain adaptation methods. Despite their simplicity, they are surprisingly difficult to beat (Daume III and Marcu, 2006). These techniques basically mean using each of the SemEval corpora as a source (SRC) dataset for transfer learning, where *Youtubean* is set as the target (TGT).

Our first domain-adaptation technique was WEIGHTED, a method that trains a model on the union of the SRC and TGT datasets, reweighting examples from SRC (Daume III and Marcu, 2006). We did so by multiplying the input embedding matrix by the given weight w, which we set to 0.2 based on the corpus size ratio. For training, we used 10-fold cross validation, adding all the examples of the SRC dataset to the training part of each fold-based arrangement. Since these model took longer to train we only experimented with the Senna pipeline. We omitted our bidirectional architectures given their poor performance and always included linguistic features, which generally contributed to an improved F1-score in our in-domain models.

RNN	SRC	Emb.	|h|	F1
LSTM	L	Google	50	57.17
RNN	L	Google	100	55.12
JRNN	L	WDeps	200	**58.30**

Table 8: Results for the WEIGHTED technique.

As Table 8 shows, using the Laptops dataset as SRC gives the best results in each case. Using this corpus led to an average absolute improvement over Restaurants of 3.79% ($p = 7.76 \times 10^{-11}$.) When it comes to embeddings, GoogleNews provided the best average performance with 53.44 F1-score. However, this value was statistically indistinguishable at $p < 0.08$ from WikiDeps, with an average 52.8 F1-score.

Our second domain adaptation method was PRED, which uses the output of a SRC-trained classifier as a feature in the TGT model. Concretely, we first trained a model using all the examples on SRC. We then fed that model with all the TGT examples, adding its outputs as additional features to the TGT dataset, thus creating a new augmented version of it. Since these features are IOB-tags, we concatenate them with the linguistic features. We trained our models on the augmented TGT dataset, choosing the best performing settings from our previous experiments on *AE*.

RNN	**SRC**	**Emb.**	**|h|**	**F1**
LSTM	L	Senna	100	56.83
BiLSTM	R	WDeps	100	52.81
BiRNN	R	WDeps	100	52.99*
BiRNN	R	WDeps	200	52.90*
RNN	R	WDeps	100	57.70
JRNN	R	WDeps	200	**59.69**

Table 9: Results for the PRED technique.

Table 9 summarizes our results for PRED. We found that using Senna as the pre-processor provided better results in average, with an 0.89% absolute gain significant at $p = 0.01$. The Restaurants dataset provided better results than Laptops in average, with an absolute gain of 3.23%, significant at $p = 8.78 \times 10^{-6}$.

Finally, Table 10 shows our best results for our introduced ARNN in the *Youtubean* dataset. For this case, we omitted random embeddings and binary features as previous experiments showed they did not contribute to increase the performance.

Embeddings	**|cw|**	**|h|**	**F1**
SennaEmbeddings	3	100	56.28
WikiDeps	3	100	57.21
GoogleNews	3	100	**57.67**

Table 10: Best results for our ARNNs for *AE* on *Youtubean*.

6.2 Joint aspect extraction and sentiment classification *(AESC)*

On our experiments for this task we based our parameter settings on the results for *AE*, so we only used bidirectional ARNN models, and skipped binary features and random embeddings.

6.2.1 Laptops

Table 11 summarizes our best results for the Laptops corpus. Based on the results for *AE*, we only used CoreNLP as a pre-processing pipeline. For the RNN baseline, embeddings also reported significant differences, with SennaEmbeddings offering average absolute gains of 5.78 F1-score ($p = 10^{-4}$) over GoogleNews and 2.47 F1-score ($p = 8 \times 10^{-3}$) over WikiDeps.

For training our ARNN we only used the CoreNLP pipeline, since it significantly outperformed Senna in our experiments for *AE*. All the values in the table were significantly different, although we observed different embeddings provided statistically equivalent results for certain lower performing parameter settings.

		Tagging F1		**Classification F1**		
Model	**Emb.**	single	joint	+	−	0
LSTM	Senna	74.30	**47.19**	77.40	12.63	80.00
RNN	Senna	74.08	46.52	77.13	17.70	**80.52**
JRNN	Senna	**76.00**	46.62	**77.97**	**22.86**	80.39
ARNN	Google	68.22	46.69	69.23	62.69	**86.83**
ARNN	Senna	**72.85**	**52.46**	**73.23**	**69.29**	85.59
ARNN	Wiki	71.46	50.85	63.94	61.07	83.23

Table 11: Results for *AESC* on Laptops

6.2.2 Restaurants

Regarding the Restaurants dataset, Table 12 shows a summary of our best results. For this case, we only used the Senna pipeline, as it provided better results for *AE*. We found that in the baseline RNNs SennaEmbeddings outperformed both other embeddings with average absolute gains of 2.37 ($p = 7.2 \times 10^{-4}$) and 3.36 ($p = 1.19 \times 10^{-6}$) F1-score WikiDeps and GoogleNews, respectively.

For our ARNN, as in the previous case, we only used CoreNLP as preprocessing pipeline given that it provided better results for *AE*. All the values in the table were significantly different.

		Tagging F1		**Classification F1**		
Model	**Emb.**	single	joint	+	−	0
LSTM	Senna	**69.24**	**44.75**	67.81	**62.40**	87.22
RNN	Senna	67.08	40.64	**70.73**	58.47	**87.39**
JRNN	Senna	66.74	40.65	67.04	49.29	86.47
ARNN	Google	73.80	50.63	78.90	**53.25**	81.08
ARNN	Senna	**79.57**	**54.75**	79.78	46.45	82.70
ARNN	Wiki	74.90	52.74	**81.47**	51.39	**83.22**

Table 12: Results for *AESC* on Restaurants

6.2.3 *Youtubean*

On *Youtubean*, as Table 13 shows, we see important performance drops compared to SemEval. In particular, the baseline models seem to be unable to correctly classify negative aspects. For this dataset, we found out that Senna provides better results than CoreNLP with an average absolute gain of 3.94 F1-score, which was significant at $p = 2.5 \times 10^{-4}$. Embeddings did not provide statistically significant differences. Similarly, binary features did not statistically contribute to the performance either.

Model	Emb.	Tagging F1		Classification F1		
		single	joint	$+$	$-$	0
LSTM	Senna	41.32	25.38	**35.83**	**9.59**	72.53
RNN	Senna	**47.59**	30.12	0	0	**76.64**
JRNN	Senna	42.86	**30.45**	23.33	0	62.32
ARNN	Google	52.84	40.58	45.39	**22.07**	79.94
ARNN	Senna	52.43	41.17	48.05	15.58	80.28
ARNN	Wiki	**55.50**	**41.49**	**52.32**	14.85	**81.07**

Table 13: Results for *AESC* on *Youtubean*.

7 Discussion

Results for aspect extraction showed that our implemented RNN baseline performs similarly to the original models by (Liu et al., 2015), although we remained unable to replicate their exact numbers. Despite that, our attention-RNN is able to provide results that are better than our implementation and comparable to the original values for both Laptops and Restaurants datasets. Moreover, we achieved these results without the need to add the linguistic features, which did not offer significant performance differences in our experiments. We think the variable sentence representation introduced by the attentional component is able to model some of the semantics encoded in these binary features.

For aspect extraction in our dataset, we see our model is able to perform better than the baseline, again without the need to add manually-crafted features. However, simple domain adaptation techniques applied to the baseline RNNs managed to obtain the best results, adding a maximum of 3.56 F1-score over the baseline. We think this shows that video reviews and written reviews share some regularities, which could be exploited further to obtain better results. In this sense, it would be interesting to apply these domain adaptations techniques to our attention-RNN model and compare the results. However, regularities among these domains seem to be limited, given that our obtained gains were small and that no domain consistently delivered better performance.

Regarding *AESC*, as shown by our decoupled results, we see all models slowly decreased their performance for aspect extraction, compared with results for *AE*. This seems reasonable given the additional challenges of performing both tasks at the same time.

When it comes to sentiment classification, we see our attention-RNN outperforms the baseline RNNs by a solid margin. However, all models tend to perform poorly for the negative ($-$) class. We believe this may be related to the imbalanced nature of the datasets, or due to the additional composition challenges negation involves, which seem to be critical in our dataset. Compared to the baseline RNNs, which in some cases seemed basically unable to detect negative sentiment, our attention-RNN model offers increased, although yet limited capabilities to deal with the negative class.

For *AESC*, we also observed that SennaEmbeddings did not always provide top performances, being outperformed by other embeddings, even though the former were previously shown to offer the best performance for aspect extraction in all cases. We think this is related to the nature of the embeddings, since SennaEmbeddings were designed for the tasks in (Collobert et al., 2011) which do not include sentiment, while other embeddings can be regarded as general-purpose.

8 Conclusions

In this paper we presented the first fine-grained opinion mining study focusing on *product video reviews*. We introduced the first annotated dataset for the domain, *Youtubean*, and aspect extraction and *AESC* with a novel attention-RNN. Our model offered state-of-the art performance for *AESC* and results comparable to a strong RNN baseline for aspect extraction. Our descriptive corpus analysis as well as the performance obtained by all the models in our dataset suggest that differences between speech and written text, discussed extensively in the literature, also extend to the domain of product reviews, where they are relevant for fine-grained opinion mining. These findings introduce relevant research challenges and concrete paths for future researchers.

For future work, we plan to increase the size of our dataset and include reviews extracted from different product categories. By doing this, we intend to make our results more robust and to further study the differences between written and video review, ultimately deriving new ways to overcome them. Finally, we also want to exploit the additional data from YouTube, such as the audio, video or specific frames extracted from it, and user comments, to improve our results.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations*. San Diego, California. http://arxiv.org/abs/1409.0473.

Tim Baysinger. 2015. YouTube wants viewers to buy directly from product review videos.

Douglas Biber. 1991. *Variation across speech and writing*. Cambridge University Press.

Darin Brezeale and Diane Cook. 2006. Using closed captions and visual features to classify movies by genre. In *Proceedings of the 7th International Workshop on Multimedia Data Mining (MDM/KDD06): Poster Session*. ACM, Washington, DC, USA.

Judith Chevalier and Dina Mayzlin. 2006. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research* 43(3):345–354. https://doi.org/10.1509/jmkr.43.3.345.

Ronan Collobert, Jason Weston, Lon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* 12:2493–2537. http://dl.acm.org/citation.cfm?id=1953048.2078186.

Hal Daume III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* pages 101–126. http://www.jair.org/papers/paper1872.html.

S. Gupta and R.J. Mooney. 2009. Using closed captions to train activity recognizers that improve video retrieval. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. pages 30–37. https://doi.org/10.1109/CVPRW.2009.5204202.

Sonal Gupta and Raymond J. Mooney. 2010. Using closed captions as supervision for video activity recognition. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2010)*. Atlanta, GA, pages 1083–1088. http://www.cs.utexas.edu/users/ai-lab/?gupta:aaai10.

Nan Hu, Ling Liu, and Jie Jennifer Zhang. 2008. Do online reviews affect product sales? the role of reviewer characteristics and temporal effects. *Inf. Technol. and Management* 9(3):201–214. https://doi.org/10.1007/s10799-008-0041-2.

YouTube Insights. 2014. The magic behind unboxing on YouTube.

Ozan Irsoy and Claire Cardie. 2014. Opinion Mining with Deep Recurrent Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 720–728. http://www.aclweb.org/anthology/D14-1080.

Salud M. Jimenez-Zafra, Giacomo Berardi, Andrea Esuli, Diego Marcheggiani, Mara Teresa Martn-Valdivia, and Alejandro Moreo Fernandez. 2015. A Multi-lingual Annotated Dataset for Aspect-Oriented Opinion Mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2533–2538. http://aclweb.org/anthology/D15-1302.

Amar Krishna, Joseph Zambreno, and Sandeep Krishnan. 2013. Polarity trend analysis of public sentiment on youtube. In *Proceedings of the 19th International Conference on Management of Data*. Computer Society of India, Mumbai, India, India, COMAD '13, pages 125–128. http://dl.acm.org/citation.cfm?id=2694476.2694505.

Matt Lawson. 2015. 2015 holiday trends - shopping moments are replacing shopping marathons. *https://adwords.googleblog.com/2015/10/2015-holiday-trends-shopping-moments.html*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. volume 2, pages 302–308. http://www.aclweb.org/anthology/P14-2050.pdf.

Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Interspeech 2016*. pages 685–689. https://doi.org/10.21437/Interspeech.2016-1352.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1433–1443. http://aclweb.org/anthology/D15-1168.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1412–1421. http://aclweb.org/anthology/D15-1166.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Grgoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*. pages 3771–3775.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 3111–3119.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open Domain Targeted Sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1643–1654. http://www.aclweb.org/anthology/D13-1171.

Roy C O'Donnell. 1974. Syntactic differences between speech and writing. *American Speech* 49(1/2):102–110.

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)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 27–35. http://www.aclweb.org/anthology/S14-2004.

Peter Schultes, Verena Dorner, and Verena Lehner. 2013. Leave a comment! an in-depth analysis of user comments on youtube. In *11. Internationale Tagung Wirtschaftsinformatik, Leipzig, Germany*. page 42.

Aliaksei Severyn, Alessandro Moschitti, Olga Uryupina, Barbara Plank, and Katja Filippova. 2014. Opinion Mining on YouTube. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1252–1261. http://www.aclweb.org/anthology/P14-1118.

Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. 2010. How useful are your comments?: Analyzing and predicting youtube comments and comment ratings. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '10, pages 891–900. https://doi.org/10.1145/1772690.1772781.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*. Association for Computational Linguistics, Avignon, France.

Yasuyuki Tahara, Atsushi Tago, Hiroyuki Nakagawa, and Akihiko Ohsuga. 2010. Nicoscene: Video scene search by keywords based on social annotation. In Aijun An, Pawan Lingras, Sheila Petty, and Runhe Huang, editors, *Active Media Technology*, Springer Berlin Heidelberg, volume 6335 of *Lecture Notes in Computer Science*, pages 461–474.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective LSTMs for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 3298–3307. http://aclweb.org/anthology/C16-1311.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 214–224. https://aclweb.org/anthology/D16-1021.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*. pages 2773–2781. http://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation* 39(2):165–210. https://doi.org/10.1007/s10579-005-7880-9.

Bin Wu, Erheng Zhong, Ben Tan, Andrew Horner, and Qiang Yang. 2014. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '14, pages 721–730. https://doi.org/10.1145/2623330.2623625.

Liheng Xu, Kang Liu, and Jun Zhao. 2014. Joint Opinion Relation Detection Using One-Class Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 677–687. http://www.aclweb.org/anthology/C14-1064.

Meishan Zhang, Yue Zhang, and Duy Tin Vo. 2015. Neural Networks for Open Domain Targeted Sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 612–621. http://aclweb.org/anthology/D15-1073.