



# Aspect Term Extraction Using Deep Learning Model with Minimal Feature Engineering

Felipe Zschornack Rodrigues Saraiva, Ticiana Linhares Coelho da Silva<sup>(✉)</sup>,  
and José Antônio Fernandes de Macêdo

Insight Data Science Lab, Federal University of Ceará, Fortaleza, Brazil  
{felipezrs,ticianalc}@insightlab.ufc.br, jose.macedo@dc.ufc.br

**Abstract.** With the explosive growth of social media on the Web, opinion mining has been extensively investigated and consists of the automatic identification and extraction of opinions, emotions, and sentiments from text and multimedia data. One of the tasks involved in opinion mining is Aspect Term Extraction (ATE) which aims at identifying aspects (attributes or characteristics) that have been explicitly evaluated in a sentence or a document. For example, in the sentence “*The picture quality of this camera is amazing*”, the aspect term is “*picture quality*”. This work proposes POS-AttWD-BLSTM-CRF, a neural network architecture using a deep learning model, and minimal feature engineering, to solve the problem of ATE in opinionated documents. The proposed architecture consists of a BLSTM-CRF classifier that uses the part-of-speech tag (POS tags) as an additional feature, along with a BLSTM encoder with an attention mechanism to allow the incorporation of another relevant feature: the grammatical relations between words. The experiments show that the proposed architecture achieves promising results with minimal feature engineering comparing to the state-of-the-art solutions.

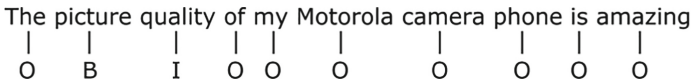
**Keywords:** Aspect term extraction · Encoder · Attention mechanism · Word dependencies · Bidirectional long short-term memory · Conditional random fields

## 1 Introduction

With the explosive growth of social media on the Web, opinion mining has been extensively investigated and consists of the automatic identification and extraction of opinions, emotions, and sentiments from text and multimedia data. [15] defines opinion mining as five different tasks. The first one is the extraction of all entity expression. The second task is the extraction of all aspect expressions of the entities, and the third consists of extracting the opinion holder and time. The fourth comprises the aspect sentiment classification that determines whether each opinion on an aspect is positive, negative or neutral. And the last task consists of generating the tuples based on the results of the previous tasks.

For example, in the sentence from a blog poster “*The picture quality of my Motorola camera phone is amazing*”. Task 1 should extract the entity ORG expression, “Motorola”. Task 2 should extract aspect expression, “picture quality”. Task 3 should find the holder of the opinions in the sentence, the blog author. Task 4 should find that the sentence gives positive opinion on the picture quality.

In this work, we focus on the second task and we tackle the aspect term extraction task as a classification problem, where each word in the sentence is tagged using the IOB2 format (short for Inside, Out and Begin). The words that are aspect terms are labeled with “B”. In case an aspect term consists of multiple words, the first word receives the “B” label and the remaining aspect words receive the “I” label. The words that are not aspect terms are labeled with “O”. The Fig. 1 illustrates the IOB2 tagging format described.



**Fig. 1.** Aspect extraction example using the IOB2 tagging format.

Companies of products and services usually spend increasing amounts of money in knowledge-based systems or expert systems to track consumer complaints online. For instance, the e-commerce applications as Amazon, Booking, among others encourage buyers to review the products or services they like and dislike to let the customers make informed decisions about the products or services they purchase. These applications can benefit from the Tasks 2 and 4 to extract all aspect expressions (products or services) and if each aspect is positive, negative or neutral. Considering the sentence “*The picture quality of my Motorola camera phone is amazing, however, the battery drains in just a few hours.*” from a buyer complaint, an information system that integrates our proposed solution (aspect extraction classifier) and an aspect sentiment classifier can provide knowledge to help Motorola to improve the phone’s battery life.

Different approaches, either supervised, unsupervised or semi-supervised, have been proposed to perform the task of aspect extraction [21] and, more recently, deep neural networks achieved promising results [20]. However, state-of-the-art techniques present some drawbacks. Usually, these proposals require a huge set of training examples [22], compelling some authors to compensate for the low accuracy obtained from models trained using a few examples with post-processing tasks, auxiliary lexicons, and language rules.

This work proposes a neural network architecture POS-AttWD-BLSTM-CRF model using a deep learning model, however with minimal feature engineering, to solve the problem of ATE in opinionated documents, like reviews of products or restaurants. We propose an encoder structure, similar to the one presented in [24], with an attention mechanism [1] that allows the use of a new additional feature: grammatical relation between words (word dependencies).

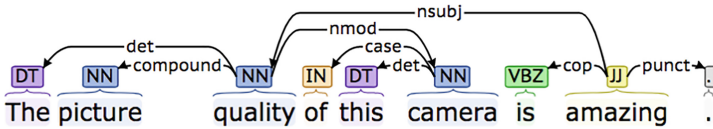
The word dependencies feature is important to the problem of aspect extraction because aspect terms are usually nominal subjects (syntactic subjects) with an adjective associated. It also helps to identify multiple word aspect terms, using the compound dependency between nouns in a noun phrase. We also use the Part-of-speech tag (POS tags) as another feature, because most of the aspect terms present in the sentences are nouns associated with one or more adjectives [20], thus the POS tagging information can help to identify which words are aspect terms. Another reason to use these two additional features is to mitigate the problem of out of vocabulary (OOV) words when there is no pre-trained word embedding to use as input to the model.

The experiments show that the proposed architecture achieves promising results with minimal feature engineering comparing to the state-of-the-art solutions. This is also the first work, to the best of our knowledge, to use an attention mechanism to harness the information of word dependencies to address the problem of aspect term extraction.

The remaining of this paper is organized as follows. Section 2 describes our proposed deep neural network architecture. Section 3 discusses the related works. Section 4 presents the conducted experiments and the results. Section 5 draws the final conclusions and the future works.

## 2 POS-AttWD-BLSTM-CRF Model

In this section, we describe some important word features used to train the proposed model. Then, we explain the proposed model architecture to perform aspect term extraction.



**Fig. 2.** Example of the word dependencies generated by the Stanford Dependencies [6].

### 2.1 Feature Selection

We sought two main goals while determining which features to use to train our model. First, to retain original information of each word. And second, to obtain a structural representation of the sentence that shows the importance of each word and the role it plays in the sentence. Following, we discuss the input features to our model: the word itself and the word dependency.

The word itself is the main input feature of the model. It contains all relevant information that has not been manually extracted from other features, and, of course, it is still useful for classifying whether or not it is an aspect term. It is represented by a pre-trained embedding vector.

Another feature is the word dependency that is a position related feature, in which all words in the sentence can be associated with each other. Figure 2 shows an example of word dependencies for the sentence “*The picture quality of this camera is amazing.*”. However, it is not practical to incorporate this feature on the data by just concatenating it to the input words (as usually performed by other approaches when considered other features, like the POS tag [20]), because the word dependency feature vector has the same size of the sentence and, as the input sentences might have different lengths, then the input vectors would have different lengths. However, the neural network should receive vectors with a fixed length as input.

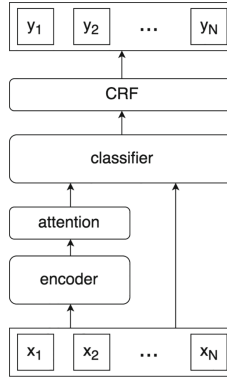
To solve this problem, we could limit the size of the sentences to a maximum number of words  $\gamma$ , and pad the input vectors with less than  $\gamma$  words. But this limits the model to not accept input sentences with size greater than  $\gamma$  without missing information. Another solution is the attention mechanism used in the encoder structure that will be explained in the next subsection. It allows the incorporation of word dependency, so the model can take advantage of the grammatical dependency between the words to regard to certain parts of the sentence, i.e. to the most relevant ones, improving the identification of aspect terms.

## 2.2 Model Architecture

As mentioned before, the trained model is capable of, given a sentence,  $W = (w_0, w_1, \dots, w_N)$  where each  $w_i$  represents a word of the sentence, classify such word  $w_i$  if it corresponds to an aspect term using IOB2 format. To this end, our proposed model was designed based on the encoder-decoder architecture proposed in [5] and [24]. This architecture has two main modules: an encoder, responsible for learning a vector representation for a sequence of tokens and a decoder, responsible for generating a sequence of tokens from a vector that represents an encoded sequence.

In general, this architecture is used to build models that can learn how to transform a sequence of tokens into another one. Our model is based on encoder-decoder architecture, however, its modules do not necessarily have the same roles. In this paper, the encoder is responsible for generating a vector representation of an entire sentence, and then provides information about the whole sentence as input for the classifier. This is similar to the original encoder purpose. But, instead of a decoder, we have a classifier that receives as input a sequence of words (represented by its features), and the vector that represents the full sentence encoded, as we can see in the Fig. 3.

LSTM (stands for Long Short-Term Memory) is a variant of recurrent neural networks (RNNs), capable to capture time dynamics in series via cycles in the network, designed to deal with the problem of gradient vanishing inherent to RNNs [17]. Because of these characteristics, they proved to be very useful in sequential labeling tasks like the ATE [13]. The encoder of our proposal is implemented using a BLSTM (stands for Bidirectional LSTM) stack architecture [9]. In BLSTMs, one LSTM layer (forward) receives the input sequence as input and



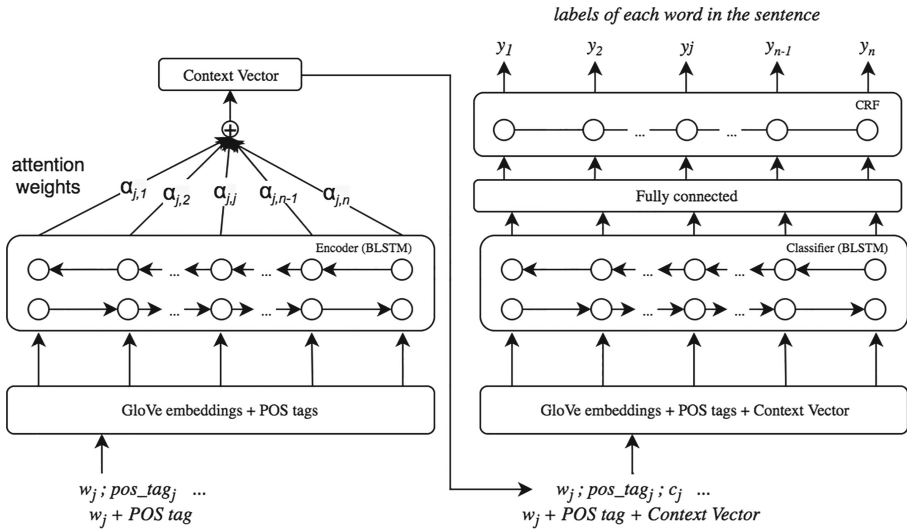
**Fig. 3.** Overview of the proposed model.

another LSTM layer (backward) receives the inverted sentence as input. All in all, BLSTM can capture both past and future information. As you can see in Fig. 4, our proposed architecture has two stacks of LSTM networks.

The layer above the encoder (Fig. 4) is the attention mechanism. It allows a model to automatically captures the parts of a source sentence that are most relevant than the others to predict a target word [1]. Instead of using the same context vector, or sentence representation, generated by the encoder for all words, it computes a different context vector for each word based on the grammatical relation between the words in the sentence. Thus, when performing the classification, every word has its own specific context vector.

The context vector is a weighted sum of the hidden outputs generated by the encoder. The attention weights (denoted by  $\alpha_{i,j}$ ) are calculated, for each word, based on the word dependencies between the word being classified and the word representation of each word in the sentence given by the encoder. Figures 5 and 6 illustrate the word dependencies matrix for the sentence “*The picture quality of this camera is amazing.*” and the calculation of the attention weights  $\alpha_{i,j}$  for the word *quality* respectively.

All in all, the encoder consists of a BLSTM, with an attention mechanism, to map the input sentence, along with its POS tags, to context vectors of fixed dimensionality. The classifier is a BLSTM with a CRF layer on top. The CRF considers the correlations between the neighbors’ labels, making a global choice instead of decoding each label independently. In ATE problem it is specifically useful to correctly label multiple word aspect terms [13]. As shown in Fig. 4, for each word in the input sentence, the classifier uses the word embedding (from GloVe [19]), the POS tag vector and the context vector obtained from the encoder with an attention mechanism to compute representations that are passed to a CRF layer to evaluate the output labels. Both encoder and the classifier are trained jointly to maximize the conditional probability of the sentence labels given an input sentence.



**Fig. 4.** The neural network architecture proposed. The attention mechanism uses the word dependencies information to weight the encoder hidden states and combines them to generate a different context vector for each word in the sentence.

It is worth to mention that we handle the words for which there is no pre-trained embedding vector, our approach sets the word embedding vector to zero. Moreover, in this paper, we use GloVe as a pre-trained embedding model, however, our approach is flexible to use any other model.

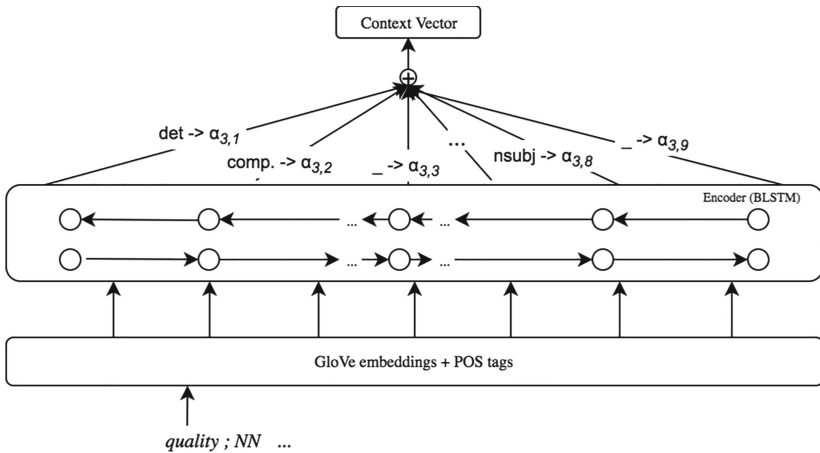
### 3 Related Work

Some deep learning models have been proposed to solve the problem of aspect extraction. The paper [12] proposed a hierarchical deep learning structure to learn representations for words (embeddings) which aim to explain the aspect-sentiment relationship at the phrase level. Their model used the dependency parse of the phrase to compute the embeddings, where each level of the tree was represented by an embedding. The embeddings learned were then used to the joint modeling of aspects and sentiments, for the posterior aspect and sentiment extraction.

The paper [20] proposes PORIA, a 7-layer Convolutional Neural Network along with a set of linguistic rules to tag each word in sentences as being aspects or not. [20] used pre-trained embeddings as input features and a sentiment lexicon, beyond that part of speech vectors as handcrafted features to improve the model performance, filtering 6 basic parts of speech (noun, verb, adjective, adverb, preposition, conjunction) and encoding it as a 6-dimensional binary vector for each input word.

	The	picture	quality	of	this	camera	is	amazing	.
The	-	-	det	-	-	-	-	-	-
picture	-	-	compound	-	-	-	-	-	-
quality	det	compound	-	-	-	nmod	-	nsubj	-
of	-	-	-	-	-	case	-	-	-
this	-	-	-	-	-	det	-	-	-
camera	-	-	nmod	case	det	-	-	-	-
is	-	-	-	-	-	-	-	cop	-
amazing	-	-	nsubj	-	-	-	cop	-	punct
.	-	-	-	-	-	-	-	punct	-

**Fig. 5.** Word dependencies matrix for the sentence “The picture quality of this camera is amazing.”



**Fig. 6.** Illustration of how the attention weights are computed for the word *quality*.

The paper [8] proposed a two-layer BLSTM-CRF model, trained on automatically labeled datasets, to extract aspects. [8] created an unsupervised algorithm to automatically label the datasets used as the training set. [29] uses a dependency-tree RNN (DT-RNN) with CRF and three hand-crafted features: POS tags, name-list, and sentiment lexicon to perform the tasks of aspect term extraction and opinion term extraction at the same time. The motivation of using the DT-RNN to encode grammatical dependency between words for feature learning was because it is infeasible or difficult to incorporate the dependency structure explicitly as input features.

[30] uses an attention mechanism to identify both aspect and opinion terms in a sentence: one attention layer to identify aspect terms and another attention layer to identify opinion terms. The goal is to avoid using engineered features, like word dependencies. In [30], each attention layer learns a prototype vector, a general feature representation for aspect terms and opinion terms. The attention weights measure the extent of correlation between each input token and the prototype using a tensor operator. Tokens with high weight values are labeled as an aspect or opinion terms. The attention layers were coupled to fully exploit the relations between aspect terms and opinion terms.

The framework proposed by [14] for ATE problem and uses truncated history attention and a selective transformation network to incorporate opinion information. *IHS\_RD* [4] is a model that uses the IHS Goldfire linguistic processor and a CRF. Such model won the competition SEMEVAL 2014, which was an ATE subtask on the Laptop domain. The paper [27] proposes a CRF classifier with manually engineered features. [27] was the winner of the SEMEVAL 2014 challenge, ATE subtask on the Restaurant domain. [26] is an RNN-CRF classifier with manually engineered features, the winner of the SEMEVAL 2016 challenge, ATE subtask on the Restaurant domain.

The paper [18] addresses the problem of aspect-based sentiment analysis, that is quite different comparing to our approach since [18] extracts the aspect category. The proposed approach is a hierarchical attention model combined with LSTM. The paper [18] also incorporates affective commonsense knowledge into the deep neural network.

[32] proposes the use of a CNN model with two types of pre-trained embeddings: general-purpose embeddings and domain-specific embeddings, to improve the aspect extraction task. The general purpose embeddings are trained on a corpus of billions tokens e.g., GloVe [19]. The domain-specific embeddings are trained using the fastText [3] library on a review corpus restricted to the same domain of the reviews where the aspect extraction task is being performed, which can be seen as a drawback because in some domains this data may not be available.

[23] improves the DE-CNN model proposed in [32] introducing control layers between the embedding and CNN layers to add noise on each CNN layer's input. The control layers and CNN layers are trained separately, in an asynchronous fashion, to avoid over-fit training data. As the model uses the same double embedding layer of the previous DE-CNN model, it also presents its same limitations.

The paper [7] provides a summary of different approaches for aspect term extraction. Most of them include standard and variants of Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). They also investigated pre-trained and fine-tuned word embeddings and part-of-speech as our approach. However, none of them proposed the same model architecture as ours with minimal feature engineering.



Other works on natural language processing tasks, like Named Entity Recognition (NER), have been using attention mechanisms to obtain state-of-the-art results. As the paper [16] which proposed a BLSTM-CRF with an attention mechanism to perform chemical NER. [2] and [31] use a self-attention mechanism to the neural architecture aiming at solving the NER problem in a cross-lingual setting by transferring knowledge from a source language to a target language with few or no labels. To the best of authors' knowledge, this is the first work that uses an attention mechanism combined with information of word dependencies to tackle the problem of aspect term extraction.

## 4 Experimental Evaluation

In this section, we present the experimental evaluation conducted to assess our proposed POS-AttWD-BLSTM-CRF model.

### 4.1 Datasets

The aspect datasets used for training the model were the Laptop and Restaurant domain training sets from the SemEval 2014 competition<sup>1</sup> and the Restaurant domain training set, Subtask 2, from the SemEval 2016 competition<sup>2</sup>. Both datasets are tagged using the IOB2 tags and their statistics are described in Table 1.

The datasets were pre-processed and annotated with POS tags generated using the Stanford POS tagger [28] and word dependencies generated by the Stanford Dependencies [6]. In total, there were 41 different types of POS tags and 39 different types of word dependencies in the datasets. It is worth to emphasize that we did not manually select a subset of those features, we used all the existent POS tags and word dependencies, and let the model select those that are most relevant during the training phase.

### 4.2 Evaluation Metrics

To evaluate the model, we calculated the precision, recall and F1 score of the sentences from the test set against the ground truth. The precision measures the proportion of words that were correctly classified as an aspect term over all the aspect terms retrieved from the model. The recall is the proportion of detected true aspect terms over the ground truth. The F1 score is a metric derived from the other two metrics, precision, and recall.

---

<sup>1</sup> <http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools>.

<sup>2</sup> <http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools>.

**Table 1.** Number of training sentences, test sentences, and aspect terms present in the SEMEVAL 2014 and 2016 datasets.

Domain	Training	Test	Total	Number of aspect terms
Laptop (2014)	3,041	800	3,841	2,373
Restaurant (2014)	3,045	800	3,845	3,699
Restaurant (2016)	2,000	676	2,676	2,530

### 4.3 Competitors

We assessed three different models based on POS-AttWD-BLSTM-CRF model to study the importance of the POS tag feature and the attention mechanism using word dependencies for the problem of aspect term extraction:

- **Enc-BLSTM-CRF**: the encoder and the BLSTM-CRF classifier using no additional features.
- **Enc-BLSTM-CRF+POS**: the encoder and the BLSTM-CRF classifier with the POS tag feature.
- **POS-AttWD-BLSTM-CRF**: our proposal, the encoder with the attention mechanism on word dependencies, and the BLSTM-CRF classifier, along with the POS tag feature.

We also compare our proposed (AttWD-BLSTM-CRF+POS) with state-of-the-art models for the ATE task:

- **BLSTM-CRF**: a BLSTM-CRF classifier from [8].
- **Poria**: a deep convolutional neural network combined with language rules, that uses filtered POS tags and a lexicon as additional features, from [20].
- **Li**: a framework for ATE that uses truncated history attention and a selective transformation network to incorporate opinion information, from [14].
- **IHS\_RD**: a model that uses the IHS Goldfire linguistic processor and a CRF [4].
- **DLIREC**: a CRF classifier with manually engineered features [27].
- **NLANGP**: a RNN-CRF classifier with manually engineered features [26].
- **DE-CNN**: a CNN model using general-purpose and domain-specific pre-trained word embeddings [32].
- **Ctrl**: the DE-CNN model using control layers between the embedding and CNN layers [23].

### 4.4 Experimental Setup

Table 2 reports the list of parameters used by our model POS-AttWD-BLSTM-CRF and its variations during the evaluation. For what concerns the parameters used by POS-AttWD-BLSTM-CRF, we remind that we need to define the number of LSTM cell units used on the encoder and classifier, the ideal number of

epochs to train the model, and the dropout rate. We randomly sampled 10% of the datasets for validation and we select the values that achieved the best results. The models were trained using Adam algorithm [11] with a learning rate of 0.01 and a dropout rate of 20% [10] on LSTM layers.

To represent each word in a sentence by its embedding, we used the 300d GloVe embeddings [19] trained on 6B tokens. Word embeddings are distributed representations of text, which encode semantic and syntactic properties of words.

For the other competitors discussed in the last section, we avoid studying the best configuration for each of their parameters. Instead, we present in the next section the results reported in their papers and the comparison with our results.

**Table 2.** The hyperparameters used for each model and dataset.

Domain	Model	# Epochs	# Cells (Enc.)	# Cells (Class.)
Laptop (2014)	Enc-BLSTM-CRF	30	128	256
	Enc-BLSTM-CRF+POS	32	128	256
	POS-AttWD-BLSTM-CRF	56	128	256
Restaurant (2014)	Enc-BLSTM-CRF	22	128	256
	Enc-BLSTM-CRF+POS	16	128	256
	POS-AttWD-BLSTM-CRF	46	128	256
Restaurant (2016)	Enc-BLSTM-CRF	58	256	256
	Enc-BLSTM-CRF+POS	37	256	256
	POS-AttWD-BLSTM-CRF	31	256	256

## 4.5 Experimental Results

Table 3 shows the evaluation metric values obtained for each model on the test sets. The results show the average performance after 10 runs.

From the results presented in Table 3, we claim that both the POS tag feature and the attention mechanism with word dependencies are important to improve the recall metric, increasing the model capability of identifying aspect terms in the sentences, reducing the number of false negatives.

Table 4 shows the F1 scores obtained using our proposed architecture and the state-of-the-art methods. The results reported for our competitors were copied from their papers. Our model achieved competitive results when compared with other state-of-the-art models, but using only the POS tag and word dependencies feature, and without manually selecting a subset of them or other features as those approaches. We believe our model is a promising alternative baseline with minimal feature engineering effort.

**Table 3.** Results obtained using the three different models. P stands for precision, R for recall and F1 for F1 score.

Domain	Model	P	R	F1
Laptop (2014)	Enc-BLSTM-CRF	89.31%	73.59%	80.69%
	Enc-BLSTM-CRF+POS	88.50%	<b>74.90%</b>	81.12%
	POS-AttWD-BLSTM-CRF	<b>89.51%</b>	74.78%	<b>81.47%</b>
Restaurant (2014)	Enc-BLSTM-CRF	89.52%	85.50%	87.46%
	Enc-BLSTM-CRF+POS	88.97%	<b>86.61%</b>	87.75%
	POS-AttWD-BLSTM-CRF	<b>89.81%</b>	86.00%	<b>87.86%</b>
Restaurant (2016)	Enc-BLSTM-CRF	72.35%	70.07%	71.15%
	Enc-BLSTM-CRF+POS	72.15%	72.23%	72.13%
	POS-AttWD-BLSTM-CRF	<b>73.05%</b>	<b>73.15%</b>	<b>73.04%</b>

**Table 4.** Comparison between the F1 scores obtained using our architecture and state-of-the-art methods. The symbol ‘-’ indicates the results were not available in the paper.

Model	Laptop (2014)	Restaurant (2014)	Restaurant (2016)
IHS_RD	74.55%	79.62%	-
DLIREC	73.78%	84.01%	-
NLANGP	-	-	72.34%
BLSTM-CRF	77.96%	84.12%	-
Li	79.52%	85.61%	73.61%
Poria	82.32%	87.17%	-
DE-CNN	81.59%	-	74.37%
Ctrl	<b>82.73%</b>	-	<b>75.64%</b>
POS-AttWD-BLSTM-CRF (our)	81.47%	<b>87.86%</b>	73.04%

## 5 Conclusion and Future Work

In this work, we have addressed the problem of aspect term extraction. We used an encoder structure with an attention mechanism that allowed the use of an important feature: grammatical dependencies between words. We also used POS tags as another feature, but unlike other works, we did not manually select a subset of those features, we let the model select those that are most relevant. Our proposed architecture compared to the state-of-the-art models shows very promising results without resorting to manual inputs like dictionaries or linguistic rules, only minimal feature engineering.

Analyzing product reviews increasingly becomes a research practice of great value to e-commerce, with the explosive growth of user-generated content on the Web. As the number of reviews is increasing to thousands or even millions, it is challenging for the potential buyers and the manufacturers to read through

them to make a wise decision. Consider an e-commerce system architecture, in the web interface, the buyer can review the products or services he/she likes and dislikes. As a future research line, we aim at extending our proposed deep learning model (POS-AttWD-BLSTM-CRF) as a component of the e-commerce architecture with a module to continuously consume the product and service reviews as stream data (using the Apache Kafka framework [25], for instance), and another module with a microservice that can consume each fired stream and extract the aspect term and aspect sentiment using our deep learning model. The results outputted by the model can be stored in the e-commerce application database and show when required by the potential buyers or the manufacturers.

## References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2014)
2. Bharadwaj, A., Mortensen, D., Dyer, C., Carbonell, J.: Phonologically aware neural model for named entity recognition in low resource transfer settings. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1462–1472 (2016)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **5**, 135–146 (2017)
4. Chernyshevich, M.: IHS R&D belarus: cross-domain extraction of product features using CRF. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 309–313 (2014)
5. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) (2014)
6. De Marneffe, M.C., Manning, C.D.: Stanford typed dependencies manual. Technical report, Stanford University (2008)
7. Do, H.H., Prasad, P., Maag, A., Alsadoon, A.: Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Syst. Appl.* **118**, 272–299 (2019)
8. Giannakopoulos, A., Musat, C., Hossmann, A., Baeriswyl, M.: Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. arXiv preprint [arXiv:1709.05094](https://arxiv.org/abs/1709.05094) (2017)
9. Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional LSTM networks for improved phoneme classification and recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 799–804. Springer, Heidelberg (2005). [https://doi.org/10.1007/11550907\\_126](https://doi.org/10.1007/11550907_126)
10. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint [arXiv:1207.0580](https://arxiv.org/abs/1207.0580) (2012)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
12. Lakkaraju, H., Socher, R., Manning, C.: Aspect specific sentiment analysis using hierarchical deep learning. In: NIPS Workshop on Deep Learning and Representation Learning (2014)
13. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint [arXiv:1603.01360](https://arxiv.org/abs/1603.01360) (2016)
14. Li, X., Bing, L., Li, P., Lam, W., Yang, Z.: Aspect term extraction with history attention and selective transformation. arXiv preprint [arXiv:1805.00760](https://arxiv.org/abs/1805.00760) (2018)

15. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C., Zhai, C. (eds.) *Mining Text Data*, pp. 415–463. Springer, Boston (2012). [https://doi.org/10.1007/978-1-4614-3223-4\\_13](https://doi.org/10.1007/978-1-4614-3223-4_13)
16. Luo, L., et al.: An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics* **34**(8), 1381–1388 (2017)
17. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint [arXiv:1603.01354](https://arxiv.org/abs/1603.01354) (2016)
18. Ma, Y., Peng, H., Cambria, E.: Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: *Thirty-Second AAAI Conference on Artificial Intelligence* (2018)
19. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014). <http://www.aclweb.org/anthology/D14-1162>
20. Poria, S., Cambria, E., Gelbukh, A.: Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl.-Based Syst.* **108**, 42–49 (2016)
21. Rana, T.A., Cheah, Y.N.: Aspect extraction in sentiment analysis: comparative analysis and survey. *Artif. Intell. Rev.* **46**(4), 459–483 (2016)
22. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning (2016)
23. Shu, L., Xu, H., Liu, B.: Controlled CNN-based sequence labeling for aspect extraction. arXiv preprint [arXiv:1905.06407](https://arxiv.org/abs/1905.06407) (2019)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: *Advances in Neural Information Processing Systems*, pp. 3104–3112 (2014)
25. Thein, K.M.M., Nyunt, T.T.S.: Apache Kafka: a high-throughput distributed messaging system. In: *Thirteenth International Conference On Computer Applications (ICCA 2015)* (2015)
26. Toh, Z., Su, J.: NLANGP at SemEval-2016 task 5: improving aspect based sentiment analysis using neural network features. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, pp. 282–288 (2016)
27. Toh, Z., Wang, W.: DLIREC: aspect term extraction and term polarity classification system. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 235–240 (2014)
28. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, pp. 173–180. Association for Computational Linguistics (2003)
29. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Recursive neural conditional random fields for aspect-based sentiment analysis. arXiv preprint [arXiv:1603.06679](https://arxiv.org/abs/1603.06679) (2016)
30. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: *Thirty-First AAAI Conference on Artificial Intelligence* (2017)
31. Xie, J., Yang, Z., Neubig, G., Smith, N.A., Carbonell, J.: Neural cross-lingual named entity recognition with minimal resources. arXiv preprint [arXiv:1808.09861](https://arxiv.org/abs/1808.09861) (2018)
32. Xu, H., Liu, B., Shu, L., Yu, P.S.: Double embeddings and CNN-based sequence labeling for aspect extraction. arXiv preprint [arXiv:1805.04601](https://arxiv.org/abs/1805.04601) (2018)