

ASPECT DETECTION USING WORD AND CHAR EMBEDDINGS WITH (BI)LSTM AND CRF

A PREPRINT

Łukasz Augustyniak, Tomasz Kajdanowicz and Przemysław Kazienko

Wrocław University of Science and Technology, Wrocław, Poland

lukasz.augustyniak@pwr.edu.pl, tomasz.kajdanowicz@pwr.edu.pl, przemyslaw.kazienko@pwr.edu.pl

September 4, 2019

ABSTRACT

We proposed a new accurate aspect extraction method that makes use of both word and character-based embeddings. We have conducted experiments of various models of aspect extraction using LSTM and BiLSTM including CRF enhancement on five different pre-trained word embeddings extended with character embeddings. The results revealed that BiLSTM outperforms regular LSTM, but also word embedding coverage in train and test sets profoundly impacted aspect detection performance. Moreover, the additional CRF layer consistently improves the results across different models and text embeddings. Summing up, we obtained state-of-the-art F-score results for SemEval Restaurants (85%) and Laptops (80%).

1 Introduction

1.1 Aspect Extraction in Sentiment Analysis

Most of current sentiment analysis approaches detect the sentiment orientation for whole documents without information about the target entities (e.g., laptops) and their aspects (e.g., battery, screen, performance). By contrast, the aspect-based sentiment analysis identifies the aspects of given target entities and estimating the sentiment polarity for each of such aspects. Further, in the sentence about a smartphone: *'the screen is very clear but the battery life is crappy'*, the sentiment is positive for one aspect *screen* but negative for another one: *battery life*.

1.2 Motivation

We wonder why not so many approaches from sequence tagging are used in aspect extraction. The sequence tagging techniques are widely used in Named Entity Recognition, Part-of-speech tagging or chunking tasks. Some research [1–3] proved they are a good choice there, but they are also suitable for NLP productization, e.g. in neural networks used in spaCy <https://spacy.io/>, one of the best NLP library. There exist two valuable studies presenting LSTM-based models for aspect extraction, however, they were applied in the limited context. Li and Lam used only one pre-trained word embedding, which was utilized with classic LSTM and their own LSTM extension [4]. Surprisingly, the embeddings trained and published by them (ref. Amazon Reviews in Sec. 4.2.1) performed very poor in our studies, see Sec. 5.

Our goal was to propose comprehensive end-to-end aspect extraction method that uses general language text embeddings combined with advanced neural network architecture: LSTM/BiLSTM with an additional optional CRF layer. We wanted to evaluate our hypothesis that extending pre-trained word embedding with character embedding will improve not good enough aspect extraction models for languages other than English, e.g., Polish. As a result, proposed by us method does not require any feature engineering or data pre-processing. To make an analysis more complete and general, we combined word and character embeddings and provide comprehensive comparison and ablation analysis of neural network model's performance. Our work is most similar to approaches applied in Named Entity Recognition (NER), but according to our best knowledge it has never been applied to aspect extraction. Simultaneously, various word embeddings extracted from different corpora were analyzed to evaluate their impact on final results.

Hence, to test all most important pre-trained word embeddings available and miscellaneous approaches (Sec. 4.2.1), we considered the following issues:

1. How perform the general language word embeddings in the dedicated domains?
2. What is the impact of word coverage between word embedding and the domain on quality of aspect extraction?
3. Does character embedding improves the aspect extraction performance?
4. Does statistical tests show that some models perform better in aspect extraction compared to the other ones?

Our main contributions are: (1) a new method for aspect extraction making use of both word and char embedding, (2) comprehensive analysis of eight LSTM-based approaches to aspect extraction on five large pre-trained word embeddings.

2 Related Work

2.1 Aspect Extraction

Researchers use several approaches for aspect-based sentiment analysis. From still commonly used rule-based methods (POS [5,6] or dependency-based [7,8]), through standard supervised learning (e.g., SVMs [9,10] and CRF [11,12] - all top approaches in SemEval2014 aspect extraction subtask) to deep learning-based approaches with CNN's or LSTM's. The interesting approach proposed by Ruder, Ghaffari, and Breslin [13]. They used a hierarchical, bidirectional LSTM model to leverage both intra- and inter-sentence relations. Word embeddings are fed into a sentence-level bidirectional LSTM which is passed into a bidirectional review-level LSTM. Poria, Cambria, and Gelbukh [6] proposed a seven-layer convolutional neural network to tag each word in opinionated sentences as either aspect or non-aspect word. However, it is worth to mention that Poria also used hand-crafted linguistic patterns to improve their extraction accuracy. Another approach presented by He et al. [14] proposes an attention-based model for unsupervised aspect extraction. The attention mechanism is used to focus more on aspect-related words while de-emphasizing aspect-irrelevant words.

2.2 Sequence Tagging

One of the first approaches using sequence tagging for aspect extraction proposed Jakob and Gurevych [15]. They used features such as token information, POS, short dependency path, word distance and information about opinionated sentences and build CRF model on the top of that. This work was extended with more hand-crafted features by Tohnad Wang [11] in DLIREC system on SemEval 2014. The DLIREC system achieved the 1st place for restaurant and the 2nd for laptops. However, aspect extraction does not use sequence tagging schemes as often as this technique is used in Named Entity Recognition tasks [1,2]. Lample, Ballesteros, Subramanian, Kawakami and Dyer [1] proposed neural architecture based on bidirectional LSTM with a conditional random field. Ma and Hovym [2] introduced a neural network architecture of bidirectional LSTM, CNN, and CRF. Hence, we see many approaches of sequence tagging in NER, but not to many applications of it in aspects extraction.

2.3 Text Embedding methods for Deep Learning

2.3.1 Word Embeddings

Word embedding is a text vectorization technique, and it transforms words in a vocabulary to vectors of continuous real numbers. It is worth to mention that each word dimension in the embedding vector represents a latent feature of this word. These vectors proved to encode linguistic regularities and patterns. The first and the most recognizable word embedding method is called Word2Vec [16]. This neural network-based model covers two approaches: Continuous Bag-of-Words model (CBOW), and Skip-gram model (SG). The second excellent word embedding approach is Global Vector (GloVe) [17], that is trained based on global word-word co-occurrence matrix. The third often used technique is fastText [18]. It is based on the Skip-gram model, where each word is represented as a bag of character ngrams. This approach allows us to compute word representations for words that did not appear in the training data. Recently, researchers started to train and use sentiment oriented word embeddings [6]. It was dictated due to the nature of the text, and they tried to include the opinionated nature of reviews which are not present in the ordinary texts.

2.3.2 Character Embeddings

Beside word embeddings more and more approaches use char-based embeddings. This kind of embeddings has been found useful for morphologically rich languages and to handle the out-of-vocabulary (OOV) problem for tasks, e.g., in part-of-speech (POS) tagging, language modeling [19], dependency parsing [20] or named entity recognition [1]. Zhang, Zhao, and, LeCun [21] presented one of the first approaches to sentiment analysis with char embedding using convolution networks. To best of our knowledge LSTM-based, char embeddings have not been used for sentiment analysis, especially for aspect extraction task.

3 Aspect Extraction Approaches using Word and Character Embedding with LSTM and CRF

Due to best of our knowledge this is the first attempt to evaluate sequence tagging aspect extraction model using so many word embeddings extended with character embeddings. In the next sections we provide a brief description of LSTM-based sequence tagging models with potential CRF layer, explain how to train character embedding models and provide all architectures used in all our experiments.

3.1 Pre-trained Word Embedding

The input layer for all tested architectures are vector representations of individual words. We used several word embeddings as we use pre-trained models in transfer learning. By this we try to mitigate the problem of training models based on the limited aspect training data. Our intuition is that aspect indication words should appear in regular contexts in large corpora. Moreover, there is big problem with the most of word embedding approaches related to the inability to handle unknown or out-of-vocabulary (OOV) words. It can be mitigated and is described in next section.

3.2 Character Embedding

An important distinction of our work from most previous approaches is that we learn character-level embedding while training instead of hand-engineering prefix and suffix information about words or improving not enough large corpus used to train word embedding. Moreover, using character-level embedding can be advantageous for learning domain-specific representations.

One the most common problem with word embeddings is related to out-of-vocabulary (OOV) words. A natural way of avoiding OOV is an extension of the embedding layer with character-level embeddings. The char embedding could represent words skipped in word vector representation.

An architecture that builds word representations from individual characters processes each word separately and maps characters to character embeddings. Then, these are passed through a bidirectional LSTM and the last states from either direction are concatenated (as in Fig. 1). The resulting vector is passed through another feed-forward layer, in order to map it to a suitable space and change the vector size as needed. Finally, we have a word representation, built based on individual characters.

3.3 LSTM-based models

LSTM networks for Long Short-Term Memory networks are a special kind of Recurrent Neural Networks (RNN) that works great on sequential data such as sequences of words. All group of RNNs takes as an input a sequence of vectors (x_1, x_2, \dots, x_n) and an output – another sequence (h_1, h_2, \dots, h_n) that represents the transformed initial sequence. However, often the problem with training on sequences is how to learn the long dependencies. Hence, in the real solutions, RNNs fail in such cases and tend to be biased towards their most recent inputs in the sequence [22]. The Long Short-term Memory Networks [23] have been designed to solve this issue. They are using a memory-cell to capture exactly the long-range dependencies. LSTMs use special gates in neurons to control the proportion of the input to be put to the memory cell, and the proportion from the previous state to be forgotten.

There are some variants of LSTMs. One is called BiLSTM [24] - bidirectional LSTM. The idea behind this architecture is to split the state neurons of a regular RNN into two parts. The former is responsible for the positive time direction (forward states), while the latter learns the negative time direction (backward states). For sequences in the text, we feed the word or character vectors from the beginning to the end of the sentence (forward pass) and in the reverse direction (backward pass). Finally, we have two outputs from the forward and backward pass that could be concatenated into one vector representing each object in the sequence. BiLSTM enables us to train neural network faster; it decodes a representation of a word in the context.

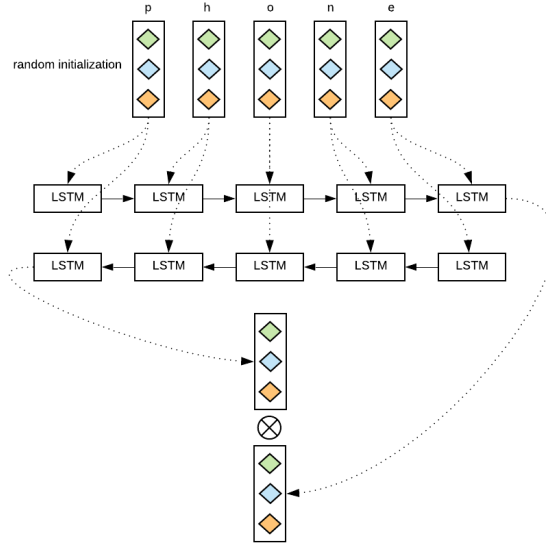


Figure 1: Architecture of character embedding.

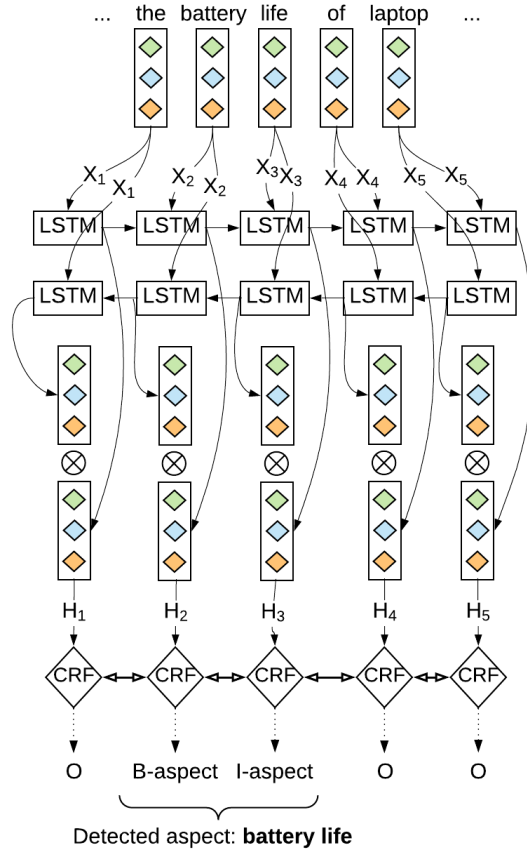


Figure 2: Architecture for word embedding: BiLSTM with the CRF layer.

Table 1: All models used in our experiments. **Word** and **Char** denote the Word Embedding and Char Embedding, respectively.

Method abbreviation	Word	Char	CRF
Wo-LSTM	yes	no	no
Wo-LSTM-CRF	yes	no	yes
WoCh-LSTM	yes	yes	no
WoCh-LSTM-CRF	yes	yes	yes
Wo-BiLSTM	yes	no	no
Wo-BiLSTM-CRF	yes	no	yes
WoCh-BiLSTM	yes	yes	no
WoCh-BiLSTM-CRF	yes	yes	yes

3.4 CRF layer

The CRF layer can learn some constraints related to the final predicted labels and ensure they are valid. What is very important these constraints can be learned by the CRF layer automatically during training process. Hence, we can get constraints related to IOB scheme such as:

- The label of the first word in a sentence starts with *B-aspect* or *O*, but not with *I-aspect*;
- *O I-aspect* is invalid sequence. The first label in aspect chunk in IOB must start with *B-aspect* not *I-aspect*, hence it should be replaced with *O B-aspect*.

Hence, we can decrease the number of invalid predicted label sequences with such constraints. Using the LSTM or BiLSTM, the tagging decision is local, only some information is taken from the context. We don't make use of the neighbouring tagging predictions. For example, in *battery life* with CRF we use information about tagging *battery* as beginning of the aspect term and it should help us to decide that word *life* should be the merge to this aspect as well.

3.4.1 IOB sentence coding

We converted SemEval 2014 datasets (more in Sec. 4.4) into IOB scheme [25]. It is a widely used coding scheme for representing sequences. IOB is short for *inside*, *outside* and *beginning*. The *B-* prefix before a tag (i.e., *B-aspect*) indicates that the tag (*aspect*) is the beginning of an annotated chunk. The *I-* prefix before a tag (i.e., *I-aspect*) indicates that the tag is inside a chunk. *I-tag* could be preceded only by *B-tag* or other *I-tag* for ngram chunks. Finally, the *O* tag (without any tag information) indicates that a token does not belong to any of the annotated chunks.

An exemplary sentence '*I charge it at night and skip taking the cord with me because of the good battery life.*' is encoded with IOB to: *I:O charge:O it:O at:O night:O and:O skip:O taking:O the:O cord:B-aspect with:O me:O because:O of:O the:O good:O battery:B-aspect life:I-aspect :O*

4 Experimental setup

4.1 Experiment workflow

We experimented with various sequential tagging approaches for aspect extraction. All methods are presented in Tab. 1. We tested eight different configurations of features and neural networks. Moreover, we used five different pre-trained word embeddings, see Sec. 4.2.1 for details. In total, we evaluated 40 experimental configurations.

4.2 Text Vectorization

4.2.1 Pre-trained Word Embeddings

As we already mentioned, we used several different word embeddings. All of them are presented in Tab. 2. It is worth to mention that there are trained based on different sizes of corpora and with different coverage of words in language.

Glove 840B - Global Vectors for Word Representation proposed by Stanford NLP Group, trained based on Common Crawl. **fastText** - Distributed Word Representation proposed by Facebook, trained on Common Crawl as well. **word2vec** - protoplast model of any neural word embedding proposed by Mikolov [Google] trained on Google News. **numberbatch** - Numberbatch consists of state-of-the-art semantic vectors derived from ConceptNet with

Table 2: All pre-trained word embeddings.

Word Embedding	# of words	Vocab	Ref
Glove 840B	840B	2.2M	[26]
fastText	600B	2M	[18]
word2vec	100B	3M	[16]
Amazon Reviews	4.7B	42K	[6]
numberbatch	2M	500K	[27]

additions from Glove, Mikolov’s word2vec and parallel text from Open Subtitles 2016 (<http://opus.lingfil.uu.se/OpenSubtitles2016.php>) trained via fastText. **Amazon Reviews** - word2vec model trained on Amazon Reviews [28]. Since it contains opinionated documents, it should be the advantage over common language texts such as Google News or Common Crawl.

4.2.2 Char Embeddings

We treated consecutive chars as sequences and passed it through forward and backward pass of the LSTM to get a vector of each char that were concatenated in the end. We initialized every char vector with a length of 25 with random numbers, hence the size of concatenated char vectors was equal to 50. We used 0.5 dropout.

4.3 Neural Network architecture

For all experiments, we used keras (<https://keras.io/>) with tensorflow (<https://www.tensorflow.org/>) and the following hyperparameters: mini-batch size: 10, max. sentence length: 30 tokens, word embedding size: 300, dropout rate: 0.5. We trained for 25 epochs using cross entropy, the Adam optimizer, and early stopping (max 2 epochs without improvement). We averaged model results according to at least 5 runs. The source code for all experiments is available at GitHub (https://github.com/laugustyniak/aspect_extraction).

4.4 SemEval datasets

We did not use SemEval 2015 or 2016 aspect extraction datasets (in 2017 there was only aspect extraction in tweets) because they were prepared as text classification with predefined aspect categories and entities. SemEval 2014 is the last one that consists of sentences with words manually annotated as aspects.

Table 3: SemEval 2014 datasets profile for Laptops and Restaurants. Multi-aspect means fraction of ngram aspects (two and more words).

		Lap.	Rest.
Train	# of sentences	3,045	3,041
	# of aspects	2,358	3,693
	multi-aspects [%]	37	25
Test	# of sentences	800	800
	# of aspects	654	1,134
	multi-aspects [%]	44	28
All	# of sentences	3,845	3,841
	# of aspects	3,012	4,827

Both datasets are quite different. There were some issues during aspect annotation, i.e., it was unclear if a noun or noun phrase was used as the aspect term or if it referred to the entity being reviewed as the whole [29]. For example in *This place is awesome*, the word *place* most likely refers to the restaurant as the whole. Hence, it should not be tagged as an aspect term. In *Cozy place and good pizza*, it probably refers to the ambience of the restaurant. In such cases, an additional review context would help to disambiguate it. Moreover, the laptop reviews often rate laptops as such without any particular aspects in mind. This domain often contains implicit aspects expressed by adjectives, e.g., *expensive*, *heavy*, rather than using explicit terms, e.g., *cost*, *weight*. We must remember that in both datasets, the annotators were instructed to tag only explicit aspects, thus, adjectives implicitly referring to aspects were discarded. The restaurant dataset contains many more aspect terms in training and in testing subsets (see Tab. 3). The majority of the aspects in both datasets are single-words, Tab. 3. Note that the laptop dataset consists of proportionally more multi-word aspects than the restaurants dataset. It could be one of the reasons why the average accuracy for the laptops is commonly lower than for restaurants.

4.5 Quality measure

4.5.1 F1-measure

The F1-measure (also called F1-score or F-score) is the harmonic mean of precision and recall. It reaches its best value at 1 and worst at 0. We calculated F1-measure only for exact matches of aspects, i.e., *battery life* aspect will be true positive only when both words will be tagged as aspects - no more or fewer words are possible. It is a strong assumption opposed to some other quality measures with weak F1 when any intersection of words between annotation and prediction are treated as correctly tagged.

4.5.2 Nemeneyi

Nemeneyi is a post-hoc test used to find groups of models that differ after a statistical test of multiple comparisons such as the Friedman test [30]. In our case, the Nemeneyi test makes a pair-wise comparison of all model's ranks over the pre-trained word embeddings and all evaluated methods. We used alpha equal to 5%. The Nemeneyi test provides critical distance for compared groups that are not significantly different from each other as presented in Fig. 7b.

4.5.3 Percentage improvement

To compare how much method M_2 improves over method M_1 we calculate the improvement according to Eq. 1. In other words, we show to what extent method M_2 gains within the possible margin left by method M_1 , i.e., to the maximum 100%.

$$improvement(M_1, M_2) = \frac{M_2 - M_1}{100\% - M_1} \quad (1)$$

where M_1 and M_2 denote F1-measures of the first and second method, respectively.

4.6 Baseline Methods

To validate the performance of our proposed models, we compare them against a number of baselines:

- **DLIREC** [11]: Top-ranked CRF-based system in ATE subtask in SemEval 2014 - Restaurants domain.
- **IHS R&D** [12]: Top-ranked system in ATE subtask in SemEval 2014 - Laptops domain.
- **WDEmb**: Enhanced CRF with word embedding, linear context embedding and dependency path embedding [31].
- **RNCRF-O** and **RNCRF-F** [32]: They used tree-structured features and recursive neural network as the CRF input. **RNCRF-O** was trained without opinion labels. **RNCRF-F** was trained with opinion labels and some additional hand-crafted features.
- **DTBCSNN+F**: A convolution stacked neural network using dependency trees to capture syntactic features [33].
- **MIN**: LSTM-based deep multi-task learning framework. It jointly handles the extraction tasks of aspects and opinions using memory interactions [4].
- **CNN**: deep convolutional neural network using *Glove.840B* word embedding as in Poria et al. [6] ¹.

The comparison of presented above models has been done in next section.

5 Results

The best F1-measure were obtained by pre-trained word embeddings (*Glove 840B* and *fastText*) extended with char embedding using BiLSTM and CRF layer and it was 85.69% and 80.13% for Restaurants and Laptops respectively, see Tab. 4 and Tab. 5. Both models achieved performance better than the best SemEval 2014 winners (84% and 74%).

¹This approach was run by us using source code available in <https://github.com/soujanyaoria/aspect-extraction>.

Table 4: All results averaged over 6 runs with standard deviations - Restaurant Dataset.

	fastText	Amazon Reviews	numberbatch	Glove 840B	word2vec
Wo-LSTM	80.8 +/- 1.49	48.78 +/- 1.02	76.26 +/- 0.75	80.91 +/- 1.1	77.73 +/- 0.74
WoCh-LSTM	79.91 +/- 1.85	65.81 +/- 2.3	76.11 +/- 1.9	81.26 +/- 0.42	78.15 +/- 0.54
Wo-LSTM-CRF	85.46 +/- 0.21	52.09 +/- 0.98	82.19 +/- 0.84	85.02 +/- 0.23	82.49 +/- 0.32
WoCh-LSTM-CRF	85.25 +/- 0.46	72.84 +/- 0.62	82.92 +/- 0.33	84.91 +/- 0.38	84.12 +/- 0.3
Wo-BiLSTM	83.17 +/- 0.54	50.49 +/- 0.87	78.57 +/- 1.04	83.56 +/- 0.22	80.16 +/- 0.74
WoCh-BiLSTM	83.27 +/- 0.61	69.53 +/- 1.52	80.89 +/- 0.26	83.55 +/- 0.3	81.39 +/- 1.08
Wo-BiLSTM-CRF	85.28 +/- 0.46	50.63 +/- 0.5	82.31 +/- 0.47	84.96 +/- 0.54	82.94 +/- 0.51
WoCh-BiLSTM-CRF	85.69 +/- 0.64	73.5 +/- 0.91	82.85 +/- 0.41	85.2 +/- 0.28	83.61 +/- 1.35

Table 5: All results averaged over 6 runs with standard deviations - Laptops Dataset.

	fastText	Amazon Reviews	numberbatch	Glove 840B	word2vec
Wo-LSTM	67.75 +/- 4.05	55.18 +/- 1.77	57.88 +/- 2.48	68.38 +/- 3.61	61.59 +/- 2.43
WoCh-LSTM	66.71 +/- 4.88	60.01 +/- 1.18	58.77 +/- 3.86	70.09 +/- 0.61	64.1 +/- 2.67
Wo-LSTM-CRF	77.95 +/- 1.79	65.15 +/- 0.73	69.19 +/- 2.5	77.72 +/- 1.42	72.88 +/- 1.12
WoCh-LSTM-CRF	77.53 +/- 0.93	70.04 +/- 1.3	74.15 +/- 0.39	77.66 +/- 0.46	75.44 +/- 1.57
Wo-BiLSTM	73.32 +/- 1.32	61.22 +/- 1.14	59.02 +/- 7.19	74.25 +/- 0.87	67.96 +/- 2.15
WoCh-BiLSTM	73.44 +/- 2.77	66.06 +/- 1.11	66.69 +/- 2.07	73.38 +/- 2.46	69.77 +/- 2.84
Wo-BiLSTM-CRF	79.34 +/- 1.23	64.89 +/- 0.75	73.03 +/- 1.02	79.99 +/- 0.72	74.93 +/- 1.0
WoCh-BiLSTM-CRF	79.73 +/- 1.36	69.65 +/- 0.97	75.09 +/- 1.75	80.13 +/- 0.34	76.38 +/- 1.37

5.1 Overall Results

We obtained the best F1-measures using *Glove.840B* (80.13% for Laptops) and *fastText* (85.69% for Restaurants) pre-trained word embeddings extended with character embedding using BiLSTM and CRF layer. Table 6 presents a comparison of our models and baselines. It can be seen that all four of our models using either *Glove.840B* or *fastText* word embeddings proved to be better than any other baseline model. It is worth mentioning that our models achieved better performance than the SemEval 2014 winners - *DLIREC* and *IHS R&D*. Summing up, our models' performance was superior in comparison to state-of-the-art models.

Table 6: Comparison of F1 scores on SemEval 2014.

Model	Laptops	Restaurants
DLIREC	73.78	84.01
IHS R&D	74.55	79.62
RNCRF-O	74.52	82.73
RNCRF-F	78.42	84.93
CNN-Glove.840B	77.36	82.76
Wo-BiLSTM-CRF-fastText	79.34	85.28
WoCh-BiLSTM-CRF-fastText	79.73	85.69
Wo-BiLSTM-CRF-Glove.840B	79.99	84.96
WoCh-BiLSTM-CRF-Glove.840B	80.13	85.2

5.2 LSTM vs BiLSTM

We have hypothesized that BiLSTM-based model is consistently better than standard LSTM. What can be observed in Fig. 3 it was verified and proved. The figure shows a comparison of models with LSTM and BiLSTM architecture across all evaluated pre-trained word embeddings. Interestingly, *Amazon Reviews* models prove to be very poor comparing to the other embeddings, even ConceptNet-based *numberbatch* and very news-based *word2vec* are better than *Amazon Reviews* word embedding. It is even more surprising because *Amazon Reviews* embedding was trained based on domain very close to laptops, i.e., Electronics.

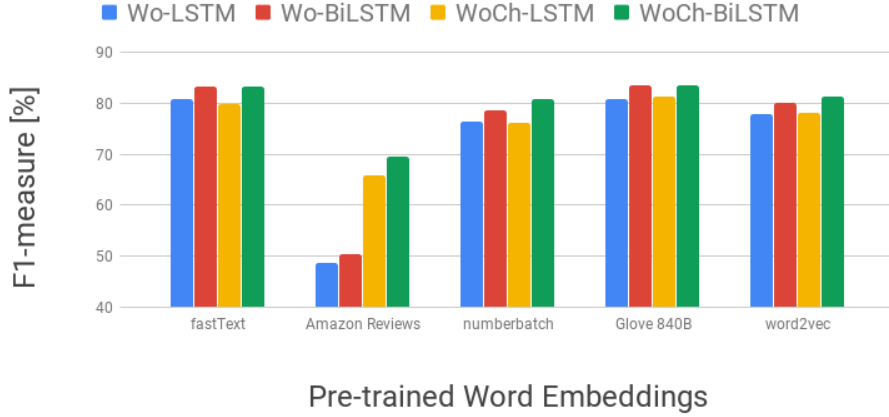


Figure 3: Comparison of LSTM and BiLSTM model’s performance - Restaurant Dataset.

5.3 Influence of the CRF layer

The existence of CRF layer proves to improve the sequence tagger as well. As it can be seen at Fig. 4 and Nemeneyi at Fig. 7b models with CRF layer are most of the time significantly better than non-CRF approaches. The same pattern we saw for Laptops dataset.

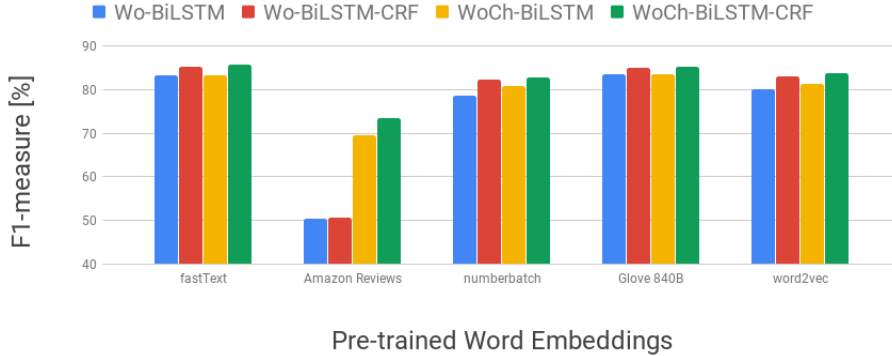


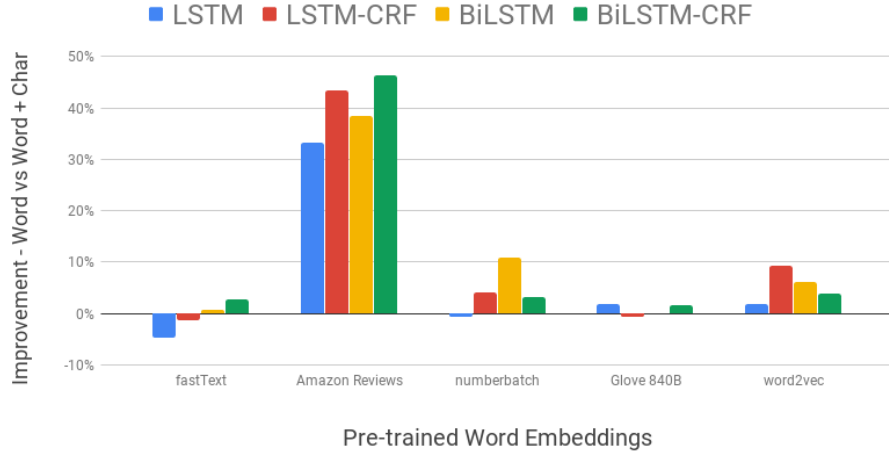
Figure 4: Analysis of CRF layer extension - Restaurant Dataset.

5.4 Character Embedding Extension

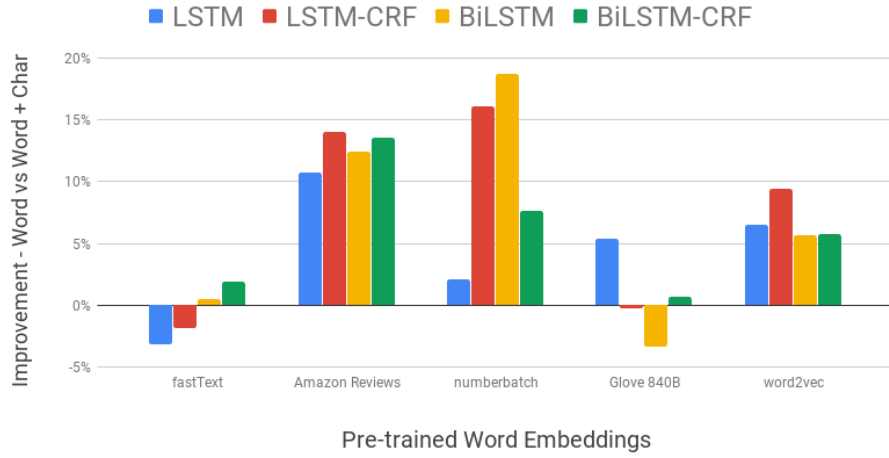
We calculated evaluated the influence of extending all neural network architectures with char embeddings according to Eq. 1. When we analyze the word coverage between dataset (Tab. 2) and used by us pre-trained word embeddings (Fig. 5) we can spot that the character embeddings work very well for low coverage word embedding (*Amazon Reviews* or *ConceptNet numberbatch*), but it could also add noise to the good word embedding and (*fastText* and *Glove 840B*) lower the overall performance.

5.5 Word Embedding Vocabulary Coverage

As we can see most of the word embeddings covers the wording of both datasets quite well. The best coverage of words contains Glove 840B and fastText models with 94% and more coverage. The word2vec and numberbatch models present a little less coverage between 86% and 90%. However, they are still good language representations. Surprisingly, Amazon Reviews model proves to be the worst in case of coverage across all embeddings with about 83% and 67% coverage for laptops and restaurants datasets accordingly. In the Restaurant reviews, there could appear more domain dependent words such as cousins names of ingredients, but only 83% coverage for laptops is a little



(a) Restaurant Dataset.



(b) Laptops Dataset.

Figure 5: Improvements provided by character extensions of Embedding Layer for different network architecture

bit unexpected, hence Amazon Reviews cover also i.e., Electronics and Laptops domains. We will investigate the influence of word embedding coverage to overall model’s accuracy in next subsections.

5.6 Statistical significance analysis

The Nemenyi pair-wise test with Friedman rank test shows the performance compared across all pre-trained word embeddings and across all evaluated methods. As seen in Fig. 7 *Glove 840B* and *fastText* word embeddings are on average the best embedding choice. Fig. 7b shows the significant improvements for models using CRF as the final layer.

6 Conclusions and Future Work

We have introduced a new accurate aspect extraction method that makes use of both word and character-based embedding. Additionally, we performed the first so wide analysis of sequence tagging approaches to aspect extraction using various neural network architectures based on LSTM and several different pre-trained embeddings. Our method outperformed all other approaches, including the best ones from SemEval 2014 competition for both datasets available.

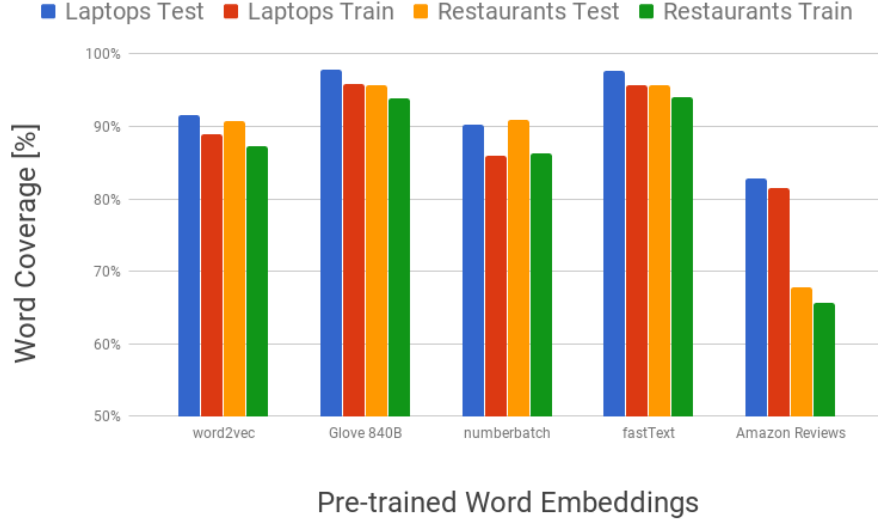
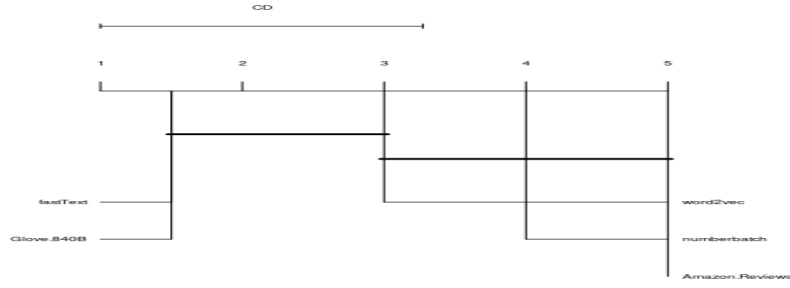
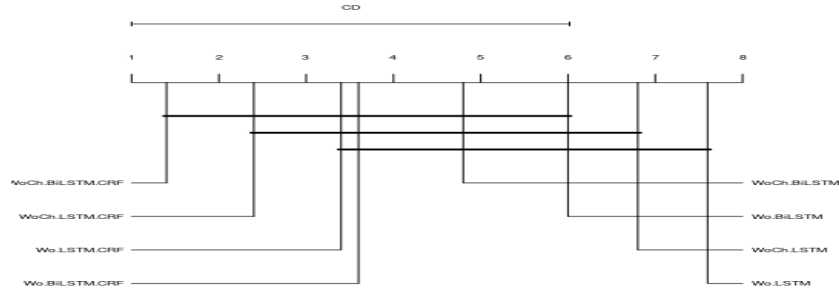


Figure 6: The coverage between words in datasets and different word embeddings.



(a) Different pre-trained embeddings across all evaluated methods.



(b) All evaluated methods across pre-trained word embeddings.

Figure 7: Nemenyi statistical test for Restaurant Dataset.

We also analyzed the influence of several characteristics of word embeddings, especially out-of-vocabulary (OOV) and model settings (neural architecture type, additional CRF layer, char embedding layer) on aspect extraction performance. We proved that combining word embeddings with character-based representations makes neural architectures more powerful and enables us to achieve better, more open representations especially for models with higher OOV rates or infrequent words. It may be notably important for texts with more strongly inflected language. This opens possibilities to use such architectures with word and character embeddings for other languages such as Polish where OOV problem will be an even bigger problem due to inflected language and smaller available corpora. For that reason our future work we will focus on applying the proposed method for the Polish language. Another direction will be focused on the application of the above concepts to building complex relationships between aspects in particular hier-

archies. Finally, we will use the proposed method for aspect extraction to generate abstractive summaries for various opinion datasets.

Acknowledgment

The work was partially supported by the National Science Centre, Poland grant No. 2016/21/N/ST6/02366, 2016/21/B/ST6/01463 and the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 691152 (RENOIR), the Polish Ministry of Science and Higher Education fund for supporting internationally co-financed projects in 2016-2019 (agreement No. 3628/H2020/2016/2) and by the Faculty of Computer Science and Management, Wrocław University of Science and Technology statutory funds.

References

- [1] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.
- [2] Xuezhe Ma and Eduard Hovy. 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, August 2016. Association for Computational Linguistics.
- [3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual String Embeddings for Sequence Labeling. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.
- [4] Xin Li and Wai Lam. Deep Multi-Task Learning for Aspect Term Extraction with Memory Interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2892, Stroudsburg, PA, USA, 2017. Association for Computational Linguistics.
- [5] Lukasz Augustyniak, Krzysztof Rajda, and Tomasz Kajdanowicz. Method for aspect-based sentiment annotation using rhetorical analysis. In *ACIIDS (1)*, volume 10191 of *Lecture Notes in Computer Science*, pages 772–781, 2017.
- [6] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42 – 49, 2016. New Avenues in Knowledge Bases for Natural Language Processing.
- [7] Soujanya Poria, Nir Ofek, Alexander Gelbukh, Amir Hussain, and Lior Rokach. Dependency Tree-Based Rules for Concept-Level Aspect-Based Sentiment Analysis. pages 41–47. Springer, Cham, 2014.
- [8] Thien Hai Nguyen and Kiyoaki Shirai. PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September):2509–2514, 2015.
- [9] Tamara Álvarez López, Jonathan Juncal-Martínez, Milagros Fernández-Gavilanes, Enrique Costa-Montenegro, and Francisco Javier González-Castaño. Gti at semeval-2016 task 5: Svm and crf for aspect detection and unsupervised aspect-based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 306–311, San Diego, California, June 2016. Association for Computational Linguistics.
- [10] Caroline Brun, Diana Nicoleta Popa, and Claude Roux. XRCE: Hybrid Classification for Aspect-based Sentiment Analysis. pages 838–842, 2014.
- [11] Zhiqiang Toh and Wenting Wang. DLIREC: Aspect Term Extraction and Term Polarity Classification System. pages 235–240, 2014.
- [12] Maryna Chernyshevich. IHS R&D Belarus: Cross-domain Extraction of Product Features using Conditional Random Fields. Technical report, 2014.
- [13] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. sep 2016.
- [14] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An Unsupervised Neural Attention Model for Aspect Extraction. Technical report, 2017.

- [15] Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1035–1045, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [16] Quoc Le, Tomas Mikolov, and Tmokolov Google Com. Distributed Representations of Sentences and Documents. 32, 2014.
- [17] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.
- [18] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [19] Wang Ling, Tiago Luis, Luis Marujo, Ramon Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W. Black, and Isabel Trancoso. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. *Emnlp-2015*, (September):1520–1530, 2015.
- [20] Miguel Ballesteros, Chris Dyer, and Noah A. Smith. Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs. aug 2015.
- [21] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. sep 2015.
- [22] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw.*, 5(2):157–166, March 1994.
- [23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [24] *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013.
- [25] L. A. Ramshaw and M. P. Marcus. Text Chunking Using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*, pages 157–176. Springer, Dordrecht, 1999.
- [26] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.
- [27] Robert Speer and Joanna Lowry-Duda. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 85–89. Association for Computational Linguistics, 2017.
- [28] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13*, pages 165–172, New York, NY, USA, 2013. ACM.
- [29] Maria Pontiki, Haris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, and Suresh Manandhar. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. Technical report, 2014.
- [30] Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. Technical report, 2006.
- [31] Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. Unsupervised word and dependency path embeddings for aspect term extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, pages 2979–2985. AAAI Press, 2016.
- [32] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626. Association for Computational Linguistics, 2016.
- [33] Hai Ye, Zichao Yan, Zhunchen Luo, and Wenhan Chao. Dependency-tree based convolutional neural networks for aspect term extraction. In Jinho Kim, Kyuseok Shim, Longbing Cao, Jae-Gil Lee, Xuemin Lin, and Yang-Sae Moon, editors, *Advances in Knowledge Discovery and Data Mining*, pages 350–362, Cham, 2017. Springer International Publishing.