Improving Opinion-Target Extraction with Character-Level Word Embeddings

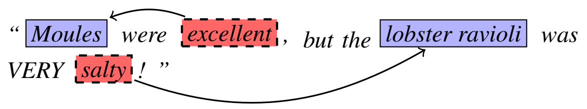
Soufian Jebbara and **Philipp Cimiano**
Semantic Computing Group, Bielefeld University
{sjebbara, cimiano}@cit-ec.uni-bielefeld.de

Abstract

Fine-grained sentiment analysis received increasing attention in recent years. Extracting opinion target expressions (OTE) in reviews is often an important step in fine-grained, aspect-based sentiment analysis. Retrieving this information from user-generated text, however, can be difficult. Customer reviews, for instance, are prone to contain misspelled words and are difficult to process due to their domain-specific language. In this work, we investigate whether character-level models can improve the performance for the identification of opinion target expressions. We integrate information about the character structure of a word into a sequence labeling system using character-level word embeddings and show their positive impact on the systems performance. Specifically, we obtain an increase by 3.3 points F_1-score with respect to our baseline model. In further experiments, we reveal encoded character patterns of the learned embeddings and give a nuanced view of the performance differences of both models.

1 Introduction

In recent years, there has been an increased interest in developing sentiment analysis models that predict sentiment at a more fine-grained level than at the level of a complete document. A key task within fine-grained sentiment analysis consists in identifying so called opinion target expressions (OTE). These are the objects of a sentiment expression. Consider the following example:



where blue boxes mark opinion targets, (dashed) red boxes the opinion terms and arrows the respective relations. In this example, there are two sentiment statements, one positive and one negative. The positive one is indicated by the word *excellent* and is expressed towards the *Moules*. The second, negative sentiment, is indicated by the word *salty* and is expressed towards the *lobster ravioli*.

In this work, we consider the task of identifying such opinion target expressions in reviews as a sequence labeling problem. A particular challenge involved in OTE identification stems from the fact that online reviews can be of low quality and contain misspelled words, novel word creations, rare words etc. We thus hypothesize that including character-embeddings might be beneficial in the context of OTE extraction, allowing a model to be robust to spelling errors as well as generalize to unseen words. A further challenge is that an OTE can span multiple tokens.

In this work, we thus investigate whether a character-based approach is capable of using the additional low-level information to improve upon a standard word-based baseline. We hypothesize that character-level word embeddings capture relevant information for opinion target expression extraction that regular (skip-gram) word embeddings lack. We propose a neural network model that learns and utilizes character-level word embeddings to extract opinion target expressions and examine its characteristics. Our experimental analysis shows that with an increase of 3.3 points F_1-score, the character information is indeed valuable for the task. Further experiments reveal encoded character patterns of the learned embeddings and give a nuanced view of the performance differences of both models.

The rest of the paper is structured as follows: Section 2 discusses related work from two domains: fine-grained sentiment analysis and character-level neural text processing. In Section 3, we describe our approach to address opinion target extraction and present the recurrent neural network models that we use to measure the impact of character information on the task. We carry out our evaluation and analysis in Section 4 and examine the learned character-level word embeddings in more detail. Finally, Section 5 summarizes our findings and presents directions for future work.

2 Related Work

Our work brings together the domains of fine-grained sentiment analysis on the one side and character-level neural text processing on the other side. In this section we give a brief overview of both domains and point out parallels to previous work.

Fine-Grained Sentiment Analysis San Vicente et al. (2015) present a system that addresses opinion target extraction as a sequence labeling problem based on a perceptron algorithm with local features.

Toh and Wang (2014) propose a Conditional Random Field (CRF) as a sequence labeling model that includes a variety of features such as Part-of-Speech (POS) tags and dependencies, word clusters and WordNet taxonomies.

Jakob and Gurevych (2010) follow a very similar approach that addresses opinion target extraction as a sequence labeling problem using CRFs. Their approach includes features derived from words, POS tags and dependency paths, and performs well in a single and cross-domain setting.

Klinger and Cimiano (2013a,b) model the task of joint aspect and opinion term extraction using probabilistic graphical models and rely on Markov Chain Monte Carlo methods for inference. They demonstrate the impact of a joint architecture on the task with a strong impact on the extraction of aspect terms, but less so for the extraction of opinion terms.

Character-Level Neural Network Models Character-level neural network models are gaining interest in many research areas such as language modeling (Kim et al., 2016), spelling correction (Sakaguchi et al., 2017), text classification (Zhang et al., 2015) and more. Most similar works from the area of character-level word representations can be found in (dos Santos and Zadrozny, 2014; dos Santos et al., 2015; Ma and Hovy, 2016). In these works, word and character level representations are successfully learned and combined to improve Part-of-Speech (POS) tagging and Named Entity Recognition (NER).

dos Santos and Zadrozny (2014) and dos Santos et al. (2015) apply a convolutional neural network (CNN) to the raw character sequence that detects character patterns and represents them as a fixed-sized embedding vector. The concatenated sequence of word and character-level embeddings is then used to predict POS or NER tags for each word.

Ma and Hovy (2016) use a similar CNN-based word structure model. However, the subsequent processing of the embedded word sequence is carried out using a bidirectional Long Short-Term Memory network (LSTM).

An example of character-level text classification not requiring any tokenization is given by Zhang et al. (2015). In their work, the authors perform text classification using character-level CNNs on very large datasets and obtain comparable results to traditional models based on words. Their findings suggest that the standard tokenization of text is indeed something to be reconsidered.

3 Model

In this work, we approach the task of extracting opinion target expressions by phrasing it as a sequence labeling problem. Doing so allows us to extract an arbitrary number of multi-word expressions in a given text. We use the **IOB** scheme (Tjong Kim Sang and Veenstra, 1999) to represent OTEs as a sequence of tags. According to this scheme, each word in our text receives one of 3 tags, namely **I**, **O** or **B** that indicate if the word is at the **B**eginning[1], **I**nside or **O**utside of an expression:

The	*wine*	*list*	*is*	*also*	*really*	*nice*	*.*
O	**I**	**I**	O	O	O	O	O

The task is thus reduced to mapping a sequence of words to a sequence of tags. We model the sequence labeling task using recurrent neural networks (RNN). RNNs allow us to easily integrate character-level knowledge into the model in the form of character-level word embeddings. To

[1]Note that the **B** token is only used to indicate the boundary of two consecutive phrases.

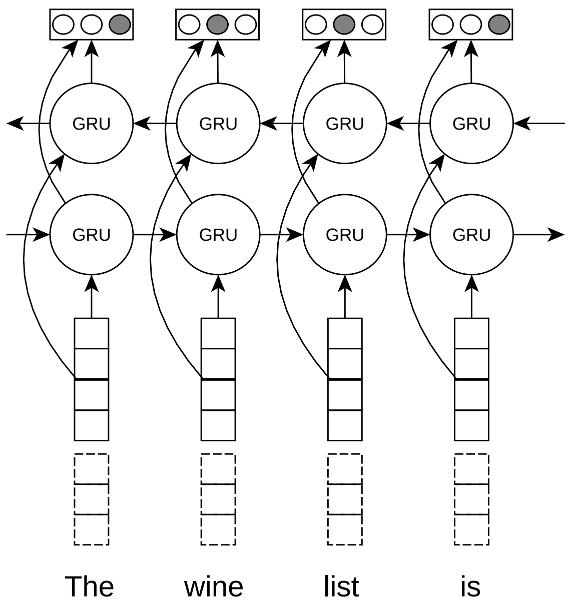


Figure 1: Illustration of the RNN sequence labeling model. The dashed boxes represent the character-level word embeddings that are only present in the character-enhanced model.

quantify the impact of these embeddings, we compare it to a baseline model that only uses word level embeddings.

3.1 Baseline Model

The proposed baseline model is a recurrent neural network that receives a word sequence $\mathbf{w} = \{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ as input features and predicts an output sequence of IOB tags $\mathbf{t} = \{\mathbf{t}_1, \ldots, \mathbf{t}_n\}$. Figure 1 illustrates the baseline neural network.

Formally, the word sequence is passed to a word embedding layer that maps each word $\mathbf{w}_i$ to its d^{wrd}-dimensional embedding vector $\mathbf{x}_i^{wrd}$ by means of an embedding matrix $\mathbf{W}^{wrd} \in \mathbb{R}^{d^{wrd} \times |V^{wrd}|}$:

$$\mathbf{x}_i^{wrd} = \mathbf{W}^{wrd} \mathbf{e}^{\mathbf{w}_i}$$

where V^{wrd} is the vocabulary of the word embeddings and $\mathbf{e}^{\mathbf{w}_i}$ is a one-hot vector of size $|V^{wrd}|$ representing the word $\mathbf{w}_i$.

The sequence of word embedding vectors is passed to a bidirectional layer (Schuster and Paliwal, 1997) of Gated Recurrent Units (GRU, Cho et al. (2014)). The GRU uses a combination of update and reset gates to improve its ability to learn long range information comparable to Long Short-Term Memory cells (Chung et al., 2014). The computation of a single GRU layer at timestep[2] i is as follows:

$$\begin{aligned}
\mathbf{z}_i &= \sigma(\mathbf{W}^z \mathbf{x}_i + \mathbf{U}^z \mathbf{h}_{i-1} + \mathbf{b}^z) \\
\mathbf{r}_i &= \sigma(\mathbf{W}^r \mathbf{x}_i + \mathbf{U}^r \mathbf{h}_{i-1} + \mathbf{b}^r) \\
\mathbf{h}_i &= (1 - \mathbf{z}_i) \odot \mathbf{h}_{i-1} + \mathbf{z}_i \odot \mathbf{g}_i \\
\mathbf{g}_i &= f(\mathbf{W}^h \mathbf{x}_i + \mathbf{U}^h (\mathbf{r}_i \odot \mathbf{h}_{i-1}) + \mathbf{b}^h)
\end{aligned}$$

where $\mathbf{x}_i$ is an element of a generic input sequence and $\mathbf{g}_i$ the computed output. $\mathbf{z}_i$ is the update gate and $\mathbf{r}_i$ the forget gate, σ is the sigmoid activation function and f is a non-linearity for which we chose the ELU (Clevert et al., 2016) activation function.

The bidirectional GRU is a variant of the GRU that processes the input sequence in forward and backward direction. The hidden states of the forward pass and the backward pass are concatenated to produce a single hidden state sequence:

$$\mathbf{g} = \{[\overrightarrow{\mathbf{g}}_1 : \overleftarrow{\mathbf{g}}_1], \ldots, [\overrightarrow{\mathbf{g}}_n : \overleftarrow{\mathbf{g}}_n]\}$$

where $\overrightarrow{\mathbf{g}}_i$ and $\overleftarrow{\mathbf{g}}_i$ are the hidden states for the forward and backward GRU layer, respectively. We choose the dimensionality of the parameters of the word-level GRU layers such that $\overrightarrow{\mathbf{g}}_i, \overleftarrow{\mathbf{g}}_i \in \mathbb{R}^{r^{wrd}/2}$, where r^{wrd} is a hyperparameter of the model.

The bidirectional connections allow the model to include words appearing before and after each timestep into the computation of the hidden states. The resulting sequence of hidden states $\mathbf{g}$ presumably incorporates the necessary context for each word in its corresponding hidden state. In a last step, each hidden state $\mathbf{g}_i$ is projected to a probability distribution $\mathbf{q}_i$ over all possible output tags, namely **I**, **O** and **B**, using a standard feedforward layer with a softmax activation function:

$$\mathbf{q}_i = softmax(\mathbf{W}^{tag} \mathbf{g}_i + \mathbf{b}^{tag})$$

with $\mathbf{W}^{tag} \in \mathbb{R}^{d^{tag} x r^{wrd}}$ and $\mathbf{b}^{tag} \in \mathbb{R}^{d^{tag}}$. For each word, we choose the tag with the highest probability as the predicted IOB tag. The predicted tag sequence can be decoded into a set of opinion term expressions using the IOB scheme in reverse.

The trainable parameters of this model are $\mathbf{W}^{wrd}$, $\mathbf{W}^{tag}$, $\mathbf{b}^{tag}$, and the parameters of the GRU $\mathbf{W}^h$, $\mathbf{U}^h$, $\mathbf{b}^h$, $\mathbf{W}^z$, $\mathbf{U}^z$, $\mathbf{b}^z$, $\mathbf{W}^r$, $\mathbf{U}^r$, $\mathbf{b}^r$ (for both directions).

[2]Each word in the input sequence is considered a timestep.

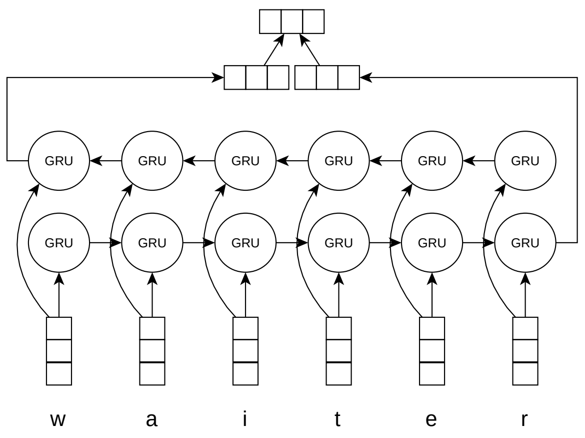


Figure 2: Illustration of the RNN character-level word embedding model. The output of this sub network is later concatenated with the regular word embeddings.

3.2 Character-Enhanced Model

We propose a variation of the baseline model from Section 3.1 that incorporates character-level information in the process of opinion target extraction. Our goal is to confirm the hypothesis that character information poses a valuable source of information for this task. Following previous work in this direction, we incorporate the character information in the form of character-level word embeddings. Figure 2 illustrates the character-level word model. Given the character sequence $\mathbf{c} = \{c_1, \ldots, c_n\}$ of a word w, we first transform each character c_i to its corresponding d^{chr}-dimensional character embedding $\mathbf{x}_i^{chr}$ using a character embedding matrix $\mathbf{W}^{chr} \in \mathbb{R}^{d^{chr} \times |V^{chr}|}$:

$$\mathbf{x}_i^{chr} = \mathbf{W}^{chr} \mathbf{e}^{c_i}.$$

Analogously to the procedure for word embeddings, V^{chr} is the character vocabulary and $\mathbf{e}^{c_i}$ is a one-hot vector of size $|V^{chr}|$ representing the character c_i. As before, the sequence of character embeddings is passed through a bidirectional GRU layer that produces two sequences of hidden states, $\overrightarrow{\mathbf{g}}$ and $\overleftarrow{\mathbf{g}}$. We choose the dimensionality of the parameters such that $\overrightarrow{\mathbf{g}}_i, \overleftarrow{\mathbf{g}}_i \in \mathbb{R}^{d^{chr}}$.

To represent the sequence of characters as a fixed-sized vector, we concatenate the final hidden states[3] of both sequences and obtain a single representation $\mathbf{g} = [\overrightarrow{\mathbf{g}_n} : \overleftarrow{\mathbf{g}_1}]$ for the character sequence. Lastly, the concatenated hidden state $\mathbf{g}$ is transformed to the final character-level word embedding using a linear feedforward layer:

$$\mathbf{x}^{cw} = \mathbf{W}^{cw}\mathbf{g} + \mathbf{b}^{cw}$$

with $\mathbf{W}^{cw} \in \mathbb{R}^{d^{chr} \times 2 \cdot d^{chr}}$ and $\mathbf{b}^{cw} \in \mathbb{R}^{d^{chr}}$.

To incorporate the word model in the overall neural network model, we pass the corresponding character sequence of each word in $\mathbf{w} = \{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ through the character model to obtain $\mathbf{x}^{cw} = \{\mathbf{x}_1^{cw}, \ldots, \mathbf{x}^{cw}\}$. The resulting character-level embeddings are then concatenated with the word level embeddings:

$$\widetilde{\mathbf{x}} = \{[\mathbf{x}_1^w : \mathbf{x}_1^{cw}], \ldots, [\mathbf{x}_n^w : \mathbf{x}_n^{cw}]\}$$

The augmented sequence $\widetilde{\mathbf{x}}$ replaces $\mathbf{x}$ in the baseline model and is passed through the remaining layers of the network. Since $\widetilde{\mathbf{x}}$ contains word and character-level information, the subsequent RNN and projection layers can make use of the additional information to improve the extraction of opinion target expressions.

The trainable parameters of this model are $\mathbf{W}^w$,$\mathbf{W}^c$,$\mathbf{W}^{cw}$, $\mathbf{b}^{cw}$, $\mathbf{W}^{tag}$, $\mathbf{b}^{tag}$, and the parameters of the GRU $\mathbf{W}^h$, $\mathbf{U}^h$, $\mathbf{b}^h$, $\mathbf{W}^z$, $\mathbf{U}^z$, $\mathbf{b}^z$, $\mathbf{W}^r$, $\mathbf{U}^r$, $\mathbf{b}^r$ for the word and character-level RNN (and for both directions).

3.3 Network Training

The optimization of the model parameters is done by minimizing the classification error for each word in the sequence using the cross-entropy loss. The optimization is carried out using a mini-batch size of 5 with the stochastic optimization technique *Adam* (Kingma and Ba, 2015). We clip the norm of the gradients to 5 and regularize our network quite rigorously using L2 regularization of 10^{-5} on $\mathbf{W}^{tag}$ and $\mathbf{W}^{cw}$, as well as Dropout (Srivastava et al., 2014) in various positions in our network. Specifically, we apply Dropout with a drop probability of 0.5 to the character and word embeddings, the output of the character-level GRUs, as well as the input and hidden sequence of the word-level GRUs as proposed in (Gal and Ghahramani, 2016). Initial experiments suggested that this strong regularization is necessary due to the moderate size of the training dataset. The networks are implemented using the machine learning framework *Keras* (Chollet, 2015).

The word embedding matrix $\mathbf{W}^{wrd}$ is initialized with a pretrained matrix of skip-gram embeddings trained on a corpus of amazon reviews

[3]Note that the final hidden state of the backwards directed GRU is the hidden state that corresponds to the first character in the sequence.

Dataset	#Sent.	#OTEs	#Chars per OTE
Train	2000	1880	2 -80
Test	676	650	3-50

Table 1: Relevant statistics of the SemEval 2016 dataset (Task 5, restaurant domain).

(McAuley et al., 2015). Earlier work showed that using a domain specific corpus in the pretraining stage significantly improves performance for similar tasks (Jebbara and Cimiano, 2016).

4 Experiments and Evaluation

In this section, we evaluate the impact of using character-level word embeddings on the task of extracting opinion target expressions from user-generated reviews. For this, we compare the character-enhanced model from Section 3.2 to the baseline RNN of Section 3.1. We start by describing the used dataset in Section 4.1. To select a fitting set of hyperparameters for each model, we perform a 5-fold cross validation on the training portion of our dataset. Using the best hyperparameters, we evaluate both models on the test portion of the data and investigate the models' properties with respect to the induced character information in Sections 4.3 and 4.4. Evaluation is carried out in terms of F_1-score of expected opinion target expressions and retrieved opinion term expressions using exact matches[4]. The research code is publicly available at `https://github.com/sjebbara/clwe-ote`.

4.1 Dataset

In our experiments, we use the data for the SemEval aspect-based sentiment analysis challenge of the year 2016 (Task 5, (Pontiki et al., 2016)). The used dataset consists of review sentences from the restaurant domain with annotations for opinion target expressions. Table 1 gives a summary of the dataset.

4.2 Hyperparameter Selection

We set the dimensionality d^{wrd} of the pretrained word embeddings to 100 and perform a grid search on a subset of the hyperparameters to find a suitable solution to be used in the final system configuration. We evaluate each candidate set of hyperparameters using a 5-fold cross validation on the training data. The search is performed for each model (`word-only` and `char+word`). We experiment with:

- the size of the word vocabulary[5] $|V^{wrd}| \in \{10000, 20000, 50000\}$ (with respect to the most frequent words),
- the size of the sentence level RNN hidden layer $r^{wrd} \in \{60, 100, 200\}$,
- and the size of the character-level RNN and the corresponding character-level word embedding vector $d^{chr} \in \{20, 50, 100\}$.

Model	$\vert V^{wrd}\vert$	r^{wrd}	d^{chr}	$\varnothing F_1$
`word-only`	50000	60	–	0.6713
`char+word`	50000	100	100	**0.6936**

Table 2: Results of a search for hyperparameters. The column $\varnothing F_1$ gives the best mean F_1-score for the best performing training epoch across cross-validation models.

Table 2 shows the best hyperparameters for each model. As expected, the search indicates that it is always better to increase the size of the word vocabulary V^{wrd}. The best model using both word and character-level information performs on average about 2.2 points F_1-score better than the best model that only uses word-level information. For the following evaluations, we instantiate and train our models according to these hyperparameters.

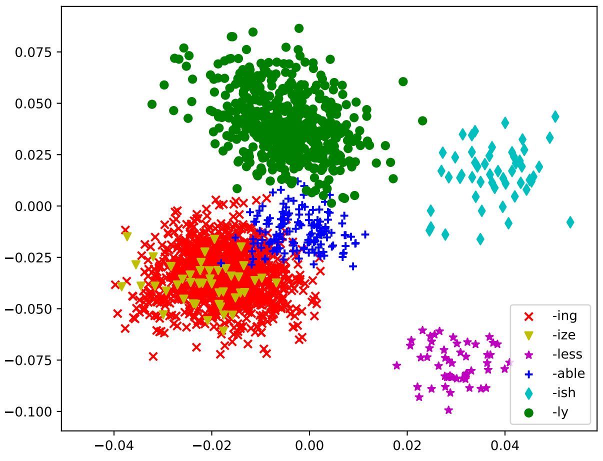
4.3 Results on Test

For the evaluation on the test set, we use the previously found hyperparameters and instantiate our models. We train both models on 80% of the training set and use the remaining 20% as a validation set for early stopping (Caruana et al., 2001). The `word-only` model reaches its best performance at epoch 35 and the `char+word` model peaks at epoch 73.

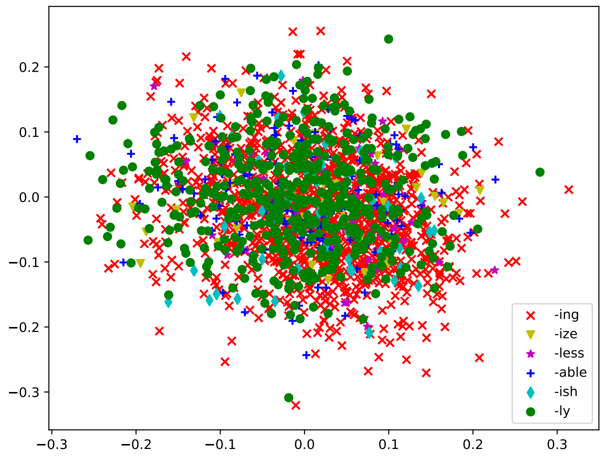
The performances of both models are given in Table 3. The results confirm our hypothesis and the findings from the cross validation that the character-level word embeddings offer a substantial improvement (3.3 points F_1-score) over the `word-only` baseline model.

[4]We use the provided evaluation code from the organizers of the SemEval 2016 challenge.

[5]The size of the word vocabulary is the main factor in terms of (GPU) memory usage.



(a) Character-level embeddings



(b) Word-level embeddings

Figure 3: Visualization of suffix information of the two employed types of embeddings.

Model	F_1-score
`word-only`	0.6260
`char+word`	**0.6586**

Table 3: Results on the test set for best performing hyperparameters. The previous findings of the usefulness of character-level word embeddings are confirmed by the results of the test set.

4.4 Analysis

In this Section, we investigate what the character-level word embeddings encode and if there are specific cases in which the character-enhanced model performs better than the baseline.

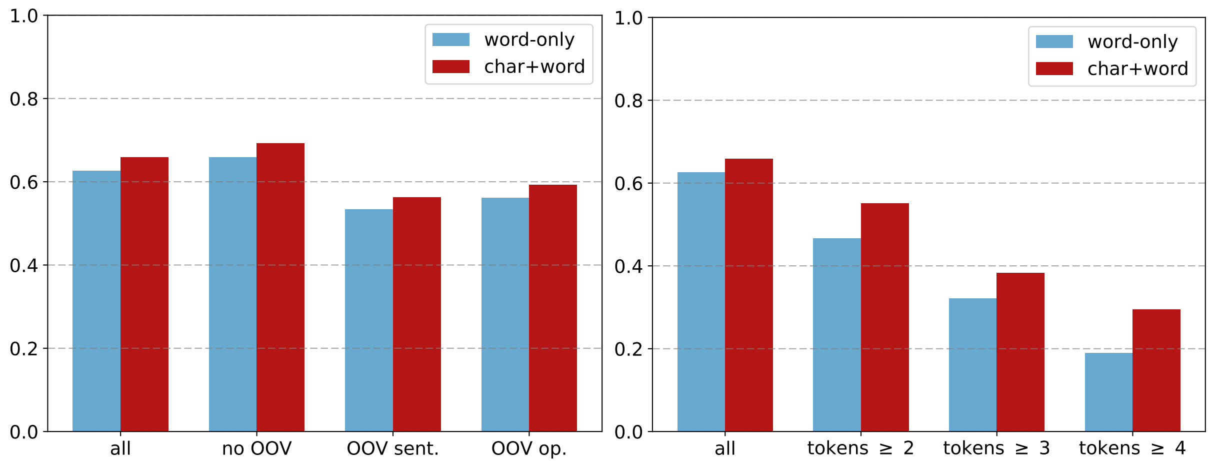
Visualization Our initial experiments in visualizing the learned model suggested that the character-level word embeddings encode morphological features of a word. To confirm this assumption, we visualize the learned embeddings using suffix information. We extract a subset of the 2000 most frequently occurring words from reviews that end on one of the following suffixes: -ing, -ly, -able, -ish, -less, -ize. We project the character-level word embeddings of the words to a 2 dimensional space using T-SNE (van der Maaten and Hinton, 2008) and plot them as a scatter plot. By highlighting each word according to its suffix, we see that the character-level embeddings are grouped according to their suffixes (see Figure 3a). Performing the same procedure with the regular skip-gram word embeddings results in no clear separation between the 6 suffix groups (see Figure 3b).

Previous work in the direction of aspect-based sentiment analysis shows a positive impact of POS tag features for the extraction of opinion phrases and opinion target expressions (Toh and Wang, 2014; Jebbara and Cimiano, 2016). It stands to reason if the character-level word embeddings act in a similar way. The morphological information of character-level word embeddings (as shown in Figure 3a) might help to disambiguate word occurrences with respect to their linguistic function in the sentence, similar to the positive effect of POS tags for this task. We leave the verification of this hypothesis for future work.

Out-of-Vocabulary Errors Next, we are interested in seeing if the improvement in F_1-score can be backtraced to Out-of-Vocabulary (OOV) word errors. For this, we compute the F_1-score on 3 different subsets of sentences for the `word-only` model and the `char+word` model:

- `no OOV`: This subset only contains sentences for which all words are part of the known vocabulary.
- `OOV sent.`: This subset contains sentences that contain an unknown word at some position in the sentence.
- `OOV op.`: The subset of sentences that contain at least one opinion target expression with an unknown word.

Figure 4a shows F_1-scores for different subsets. Surprisingly, we can see that the F_1-scores rise and fall similarly for both models regardless of



(a) Performance on OOV-related subsets.

(b) Performance on multi-word phrases.

Figure 4: Illustration of performance differences for different subsets of sentences.

Word	atmosphere	restaurant	service
Nearest Neighbors	*atomosphere*	*restaraunt*	customer
	ambience	eatery	*serivce*
	atmoshpere	*restuarant*	costumer

Table 4: Three commonly used words in restaurant reviews and their 3 nearest neighbors in the embedding space. Often, misspelled versions (*italic*) of the original word are among its closest neighbors.

the evaluated subset. This suggests, that the positive influence of the character information does not particularly help in those cases where the text contains previously unseen words (e.g. misspelled words). We assume that the positive impact on these cases is mitigated since the domain specific skip-gram word embeddings already contain various writing errors that frequently occur in customer reviews. This can be seen in Table 4, which shows the nearest neighbors of exemplary words in the skip-gram embedding space. We see that common writing mistakes are often already captured by the word embeddings.

Multi-Word Expressions Another possible cause for the performance difference of both models might be related to the length of opinion target expressions[6]. This hypothesis is motivated by the idea that e.g. variations in spelling with respect to hyphenation (e.g. *bartenders* vs. *bar tenders* or *wait staff* vs. *wait-staff*) could have less of an influence on the character-based model than on the word-based model. To test this idea, we consider subsets of sentences that contain at least one OTE that is a multi-word expression of more than or equal to k words. The performance differences for $k \in \{2, 3, 4\}$ are visualized in Figure 4b.

The first thing to notice is that both models are strongly affected by the length of the OTEs. Longer expressions seem to be harder to extract in general. However, we can observe that the character model is influenced by the length of an OTE to a lesser degree. While the difference in F_1-score for all sentences between the `word-only` model and `char+word` model is about 3.3, the differences for OTEs composed of more than or equal to 2, 3, and 4 words are 8.4, 6.1 and 10.4, respectively.

5 Conclusion

There is a growing interest in character and subword-level models for natural language processing in recent years. Tokenization is a crucial step for many applications, yet neglects the information that can be gained from the character structure of a word itself.

In this work, we were able to show that character-level information assists in the task of opinion target extraction, an important step in aspect-based sentiment analysis. We compared a

[6]In terms of words.

model using only word-level features to a more sophisticated model that also includes character-level word embeddings. We showed that the more complex character model consistently outperforms the baseline model with a substantial margin of 3.3 points F_1-score. A visualization of the learned embeddings revealed encoded morphological regularities that we could not find in our skip-gram word embeddings. Through experiments on different subsets of the data, we linked the positive influence of the character-level word embeddings to the difficulty of extracting multi-word expressions. We did not observe a performance difference for Out-of-Vocabulary cases.

However, it is not entirely clear how exactly the additional character information contributes to the task of extracting opinion target expression. In general, we suspect that the morphological information of character-level word embeddings helps to disambiguate word occurrences similarly to the positive effect of POS tags for OTE extraction. A confirmation of this hypothesis remains for future work.

Another interesting direction for future work is the pretraining of parts of the network to enrich the character-based word representation. We believe that character-level language models pose an interesting candidate for this.

The positive results of this work and the remaining research questions suggest a need to focus further research effort in the direction of character-level neural network models in order to improve token-based approaches or even replace the need for tokenization altogether.

Acknowledgments

This research was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

References

Rich Caruana, Steve Lawrence, and C. Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, MIT Press, pages 402–408.

Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734.

Francois Chollet. 2015. Keras -theano-based deep learning library. https://github.com/fchollet/keras.

Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Deep Learning Workshop*.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Cicero dos Santos, Victor Guimaraes, RJ Niterói, and Rio de Janeiro. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*. page 25.

Cicero dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning*. pages 1818–1826.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 1035–1045.

Soufian Jebbara and Philipp Cimiano. 2016. Aspect-Based Relational Sentiment Analysis Using a Stacked Neural Network Architecture. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*. pages 1123—-1131. https://doi.org/10.3233/978-1-61499-672-9-1123.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*. pages 2741–2749.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*.

Roman Klinger and Philipp Cimiano. 2013a. Bi-directional inter-dependencies of subjective expressions and targets and their value for a joint model. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), Volume 2: Short Papers*. pages 848–854.

Roman Klinger and Philipp Cimiano. 2013b. Joint and pipeline probabilistic models for fine-grained sentiment analysis: Extracting aspects, subjective phrases and their relations. In *Proceedings of the 13th IEEE International Conference on Data Mining Workshops (ICDM)*. pages 937–944. https://doi.org/10.1109/ICDMW.2013.13.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, volume 1.

Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, New York, NY, USA, pages 785–794. https://doi.org/10.1145/2783258.2783381.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. pages 19–30. http://aclweb.org/anthology/S/S16/S16-1002.pdf.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*. pages 3281–3287.

Iñaki San Vicente, Xabier Saralegi, and Rodrigo Agerri. 2015. Elixa: A modular and flexible ABSA platform. In *Proceedings of the 9th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Denver, Colorado, pages 748–752.

M. Schuster and K. K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681. https://doi.org/10.1109/78.650093.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of European Chapter of the ACL (EACL)*. Bergen, Norway, pages 173–179.

Zhiqiang Toh and Wenting Wang. 2014. DLIREC: Aspect Term Extraction and Term Polarity Classification System. In *Proceedings of the 8th International Workshop on Semantic Evaluation*. pages 235–240.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research* 9:2579–2605.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. pages 649–657.