Aspect-Based Relational Sentiment Analysis Using a Stacked Neural Network Architecture

Soufian Jebbara and Philipp Cimiano

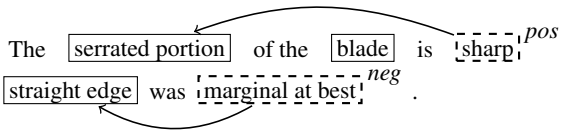
Semantic Computing Group, Cognitive Interaction Technology – Center of Excellence (CITEC), Bielefeld University, Germany, {sjebbara,cimiano}@cit-ec.uni-bielefeld.de

Abstract. Sentiment analysis can be regarded as a relation extraction problem in which the sentiment of some opinion holder towards a certain aspect of a product, theme or event needs to be extracted. We present a novel neural architecture for sentiment analysis as a relation extraction problem that addresses this problem by dividing it into three subtasks: i) identification of aspect and opinion terms, ii) labeling of opinion terms with a sentiment, and iii) extraction of relations between opinion terms and aspect terms. For each subtask, we propose a neural network based component and combine all of them into a complete system for relational sentiment analysis.

The component for aspect and opinion term extraction is a hybrid architecture consisting of a recurrent neural network stacked on top of a convolutional neural network. This approach outperforms a standard convolutional deep neural architecture as well as a recurrent network architecture and performs competitively compared to other methods on two datasets of annotated customer reviews. To extract sentiments for individual opinion terms, we propose a recurrent architecture in combination with word distance features and achieve promising results, outperforming a majority baseline by 18% accuracy and providing the first results for the USAGE dataset. Our relation extraction component outperforms the current state-of-the-art in aspect-opinion relation extraction by 15% F-Measure.

1 Introduction

Sentiment analysis can be regarded as a relation extraction problem in which the sentiment of some opinion holder towards a certain aspect of a product, theme or event needs to be extracted. While most sentiment analysis methods extract an overall polarity score for a complete text, the following example clearly shows that this is not sufficient:



The example shows an excerpt of a customer review regarding a kitchen knife. The opinion expression "sharp" constitutes a positive opinion towards the aspect "serrated portion". In the same sentence, a negative opinion is expressed by the phrase "marginal at best" towards the aspect "straight edge". Sentiment analysis needs to be regarded thus as a relation extraction problem consisting of three parts:

1. the extraction of *aspect* and *opinion terms* with respect to the discussed product, theme or event,
2. the labeling of these opinion terms with a *sentiment* (e.g. "positive", "neutral", "negative"), and
3. the extraction of *relations* between aspect and opinion terms.

In this work, we propose a complete and modular architecture that addresses all three tasks. The extraction of aspect and opinion terms can essentially be regarded as a tagging task and can potentially be tackled by sequence modeling techniques such as Hidden Markov Models, Conditional Random Fields (CRFs) etc. Besides, deep neural networks have received increasing interest in recent years and have been applied very successfully to a great variety of natural language processing (NLP)-related tasks. In particular, Convolutional Neural Networks (CNN) have been proposed as a general method for solving sequence tagging problems [5]. Also recurrent neural networks (RNN) have been applied to NLP-related tasks [2].

Building on these encouraging results, in this paper, we employ several neural network based components which we combine into a single architecture to address relational sentiment analysis in three steps. Firstly, we propose a component that combines convolutional neural networks with recurrent neural networks to extract aspect and opinion terms. Roughly, the method combines a CNN based sequence tagger with an RNN based tagger by stacking the RNN onto the CNN's produced feature sequence. The motivation behind this approach is that the RNN on top of the CNN provides a way to preserve information over longer distances in the processed text. While the deep CNN based tagger is able to capture local dependencies around a word of interest, it cannot incorporate knowledge that appears far away from its current focal position. The RNN on top, however, might still be able to incorporate locally extracted features of the CNN from far preceding positions in a text through its recurrent hidden layer.

Secondly, a recurrent neural network extracts the expressed sentiment of each opinion term by using Part-of-Speech (POS) tags, word and distance embedding features. Our method considers the opinion terms in a wide context while still being able to label multiple opinion terms in a single sentence.

Thirdly, we extract aspect-opinion relations by using a similar RNN model to classify extracted aspect and opinion terms in a pairwise fashion. The combination of all these components allows us to realize sentiment analysis on a very fine-grained level, putting individual aspect and opinion mentions in relation. A schematic visualization of the complete architecture can be seen in Figure 1. Our contributions are the following:

- We present a complete architecture that addresses sentiment anal-

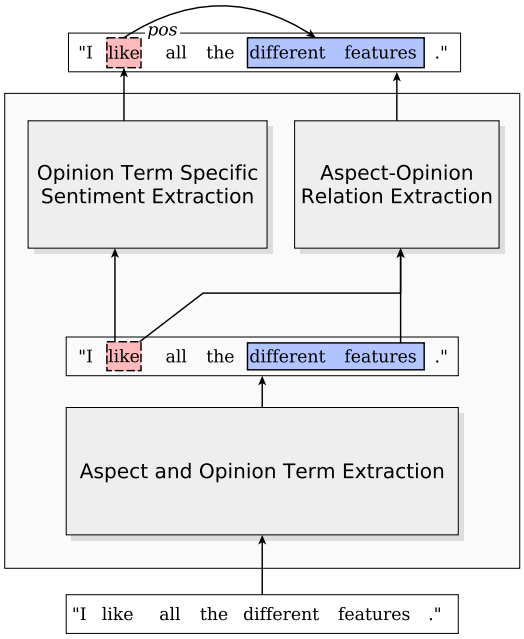


Figure 1. An architecture for sentiment analysis as a relation extraction problem. The architecture comprises 3 components that each address a subtask of the problem: i) identification of aspect and opinion terms, ii) extraction of opinion term specific sentiment and iii) extraction of relations between opinion terms and aspect terms.

ysis as a relation extraction problem to offer a very fine-grained analysis. The architecture works in a three step approach by extracting aspect and opinion terms, opinion-term specific sentiment and aspect-opinion relations separately.

- For all three subtasks, we present a neural network based component that achieves competitive and state-of-the-art results without extensive, task-specific feature engineering.
- Using the example of aspect and opinion term extraction, we show the impact of training the component with word embeddings initialized from a domain-specific corpus, showing that using domain-specific embeddings increases performance by 6.5% F-Measure as compared to randomly initialized embeddings.
- We show the impact of a component performing aspect and opinion term extraction jointly versus predicting each type of phrase separately, demonstrating that joint prediction increases F-measure performance by 1% for aspect and 5% for opinion terms.
- We present a novel approach that is able to extract opinion term specific sentiment. Our approach achieves performances high above our baseline and provides the first results on the USAGE dataset on the task of opinion term specific sentiment prediction, thus setting a strong baseline for future research.
- Finally, we show that the relation extraction component is applicable to the sentiment analysis problem and show that our approach outperforms the current state-of-the-art in aspect-opinion relation extraction by 15% F-measure.

The paper is structured as follows: In the following section, we discuss related work from the domains of aspect based sentiment analysis and relation extraction with respect to the individual subtasks these systems address. Next, in the sections 3, 4, and 5, we describe the components we use for our three subtasks. Section 6 describes our evaluation of the individual components that we apply to two datasets. We describe how parameters of the networks were optimized, the training procedure and the results for the different components on the two datasets. Lastly, we give a conclusion and discuss issues for future work.

1.1 Related Work

Our work is inspired by different related approaches. Overall, our work is in line with the growing interest of providing more fine-grained, aspect-based sentiment analysis [16, 13, 22], going beyond a mere text classification or regression problem that aims at predicting an overall sentiment for a text.

Vicente et al. [25] present a system that addresses opinion target extraction as a sequence labeling problem based on a perceptron algorithm with local features. The system also implements a sentiment polarity classifier to classify individual opinion targets. The approach uses a window of words around a given opinion target and classifies it with an SVM classifier based on a set of features such as word clusters, Part-of-Speech tags and polarity lexicons.

Aspect and opinion term extraction for sentiment analysis has also been addressed using probabilistic graphical models. Toh and Wang [32] for instance propose a Conditional Random Field (CRF) as a sequence labeler that includes a variety of features such as POS tags and dependencies, word clusters and WordNet taxonomies. Additionally, the authors employ a logistic regression classifier to address aspect term polarity classification. Klinger and Cimiano [12, 13] have modeled the task of joint aspect and opinion term extraction using probabilistic graphical models and rely on Markov Chain Monte Carlo methods for inference. They have demonstrated 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. The approach to semantic role labeling proposed by Fonseca et al. [6] is closely related to our approach to aspect term extraction in that the task is phrased as a sequence tagging problem to which a convolutional neural network is applied.

Most relevant in terms of aspect and opinion term extraction are the works of Liu et al. [17] and Irsoy and Cardie [10]. Liu et al. address the extraction of opinion expressions while Irsoy and Cardie focus on the extraction of opinion targets. Both approaches frame the respective tasks as a sequence labeling task using RNNs.

Our work is related to other approaches using deep neural network architectures for sentiment analysis. Lakkaraju et al. [15] for example present a recursive neural network architecture that is capable of extracting multiple aspect categories[1] and their respective sentiments jointly in one model or separately using two softmax classifiers.

Our approach to relation extraction is inspired by the work of Zeng et al. [34], who address a relation extraction task between pairs of entities using a convolutional neural network architecture. Their approach combines lexical features for both entities with sentence level features learned by a CNN model.

All of the above works address a subtask of relational sentiment analysis as we define it in this work, namely i) aspect and opinion term extraction ii) opinion-term specific sentiment extraction and iii) relation extraction. However, none of the above publications target *all* subtasks in a single architecture or offers the same degree of granularity in the sentiment analysis. We are thus the first to propose a

[1] Here, we distinguish between the terminologies of aspect *category* extraction and aspect *term* extraction: The set of possible aspect categories is predefined and rather small (e.g. `Price, Battery, Accessories, Display, Portability, Camera`) while aspect terms can take many shapes (e.g. “sake menu”, “wine selection” or “French Onion soup”).

neural architecture that addresses all three subtasks within one system.

2 Datasets

For the evaluation of this work, we employ two datasets that provide annotated reviews for the task of relational sentiment analysis.

2.1 SemEval2015

The SemEval2015 Task 12 dataset [22] is used to evaluate our systems for the subtask of aspect term extraction. The dataset provides a collection of reviews from different domains (Restaurant, Laptops, Hotels), annotated on different aspect and sentiment levels. We only make use of the data from the restaurant domain as it contains annotations for explicitly mentioned aspect terms. The datasets for the laptop and hotel domains only contain annotations for aspect categories without annotated textual mentions. Note that we can only use this dataset to evaluate our architecture on the task of extracting aspect terms since the dataset is not annotated with respect to opinion terms, and therefore also without aspect-opinion relations.

2.2 USAGE

The USAGE corpus [14] is a collection of annotated English and German Amazon reviews of different product categories. The annotations include (among others) the mentioned aspect terms, the opinion terms marked with a sentiment and relations between aspect and opinion terms. We refer to Klinger and Cimiano [14] for a more detailed description of the dataset.

We restrict our use of this corpus to the annotations for the English reviews and follow the evaluation procedure based on 10-fold cross-validation and strict matching proposed by Klinger and Cimiano [14]. This dataset provides annotations for all our subtasks hence we will evaluate all our components on this dataset.

3 Aspect and Opinion Term Extraction

In this work, we compare different choices for neural network based components on the task of aspect and opinion term extraction. We interpret the extraction task as a sequence labeling task, similar to other sequence labeling tasks [32, 6] and predict sequences of tags for sequences of words. We use the IOB2 scheme [31] to represent our aspect and opinion annotations 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, **I**nside or **O**utside of an annotation:

The	**sake**	**menu**	should	not	be	overlooked	!
O	B	I	O	O	O	O	O

In this example the bold "**sake menu**" is an aspect term annotation that we encoded with the IOB2 scheme. We decided on encoding aspect term annotations separately from opinion term annotations thus resulting in two separate tag sequences per review. This procedure seems reasonable, since the USAGE dataset allows overlapping aspect and opinion term annotations. Encoding them into a single tag sequence would require to use a bigger tag set[2] which we would expect to hinder the learning procedure.

[2] The tag set would need to cover single *and* overlapping annotations using the following tag set: {I-Aspect, I-Opinion, I-Aspect-Opinion, B-Aspect, B-Opinion, B-Aspect-Opinion, O}

For most experiments, we instantiate two separate models – one for extracting aspect terms and another for extracting opinion terms – and train both separately to predict their respective tag sequences. However, as shown in Section 3.5 it is also possible to extract aspect and opinion terms jointly without using a larger tag set. For the actual sequence labeling, we compare convolutional neural networks, recurrent neural networks and combinations thereof since these are capable of dealing with sequential data.

In the following, we discuss the features used by our systems, which include both Part-Of-Speech tags as well as word embeddings. The word embeddings are learned from a domain-specific corpus of Amazon reviews using a skip-gram model [20]. Then, we present our different choices of components that we experimentally examine for the aspect and opinion term extraction subtask. Here, we describe a convolutional network architecture as well as a recurrent network architecture as baseline systems. We then describe a stacked architecture that feeds features of multiple convolutional layers to a recurrent layer that, in turn, produces a tag sequence. As a preview for future work, we also propose a component that extracts aspect and opinion terms jointly.

3.1 Features

Distributed word embeddings have proven to be a useful feature in many NLP tasks [5, 26, 16] in that they often encode semantically meaningful information about words [20, 26]. In this work, we employ word embeddings which we train on huge amounts of reviews since this data is closely related to our main application domain: relational sentiment analysis of customer reviews. The corpus includes the dataset of McAuley et al. [18, 19] which consists of $\approx$ 83 million reviews from 1996 to 2014. We train the skip-gram model [20] with hierarchical softmax as it is implemented in the topic modeling library *gensim* [24]. All reviews are lowercased and the dimensionality of the word vectors is set to $D_{word} = 100$. Rare words that appear less than 10 times in the corpus are replaced with a special token `<UNK>`. This token is later used to represent previously unseen words in order to provide a vector for each word at test time. The resulting vocabulary contains $\approx$ 1 million unique words, which we trim to the 200000 most frequent words.

To quantify the impact of using these domain-specific embeddings, we also compute word embeddings on a domain-independent corpus of Wikipedia articles. As we show in Section 6.2, the domain-specific word embeddings indeed outperform the more general Wikipedia word embeddings.

Additionally to the sequence of word embeddings, we evaluate the use of corresponding Part-Of-Speech tags for each word that we obtain from the Stanford POS tagger [33]. The tag set contains 45 tags plus one additional "padding" tag. We encode these tags in the One-Hot[3] encoding, which results in a POS tag feature vector with $D_{pos} = 46$ dimensions for each word.

3.2 Convolutional Neural Network Model

While convolutional neural networks were originally intended for image processing, they have been successfully applied to several natural language processing tasks as well [23, 11, 27]. When working with sequences of words, convolutions allow to extract local features around each word.

The CNN component for sequence tagging that we propose is composed of several sequentially applied layers that transform an

[3] A vector of 0s with a single 1 to represent the specific tag.

initial sequence of words (i.e. a review) into a sequence of IOB2 tags. This sequence of tags encodes predicted aspect and opinion annotations for the given review. More formally, the process from word sequence to tag sequence can be described as follows:

Given a sequence of arbitrary length N of words:

$$[w]_1^N = \{w_1, \ldots, w_N\}$$

that correspond to a vocabulary V, our model applies a word embedding layer to each sequence element to retrieve a sequence of word embeddings $u_n \in \mathbb{R}^{D_{word}}$:

$$[u]_1^N = \{u_1, \ldots, u_N\}.$$

This is done by treating the embedding matrix $W_{word} \in \mathbb{R}^{D_{word} \times |V|}$ as a lookup table and returning the column vector that corresponds to the respective word index[4].

Then, each window of l_{conv} consecutive vectors[5] in $[u]_1^N$ around u_n is convolved into a single vector $h_n^1 \in \mathbb{R}^{D_{conv}}$, where D_{conv} specifies the number of feature maps for this convolution. Precisely, the convolution is performed on the concatenated vector $z_n \in \mathbb{R}^{D_{word} \cdot l_{conv}}$ that we define as:

$$z_n = u_{n-(l_{conv}-1)/2} \oplus \ldots \oplus u_{n+(l_{conv}-1)/2},$$

where $\oplus$ represents the concatenation operation. The convolution at position n in the sequence is then:

$$h_n^1 = \sigma(W_{conv} z_n + b_{conv}),$$

where the kernel matrix $W_{conv} \in \mathbb{R}^{D_{conv} \times D_{word} \cdot l_{conv}}$ and the bias vector b_{conv} are shared across all windows for this convolution. The function σ is an element-wise non-linear activation function such as the rectified linear function $f(x) = max(0, x)$. The resulting sequence is then:

$$[h^1]_1^N = \{h_1^1, \ldots, h_N^1\}.$$

This convolution operation can be applied several times (with different W_{conv}, b_{conv}, l_{conv}, and on the output sequence of the previous convolution) to yield a sequence of more abstract representations:

$$[h^m]_1^N = \{h_1^m, \ldots, h_N^m\}.$$

In a last step we apply a standard affine neural network layer with a softmax activation function to each individual sequence element that projects the hidden representation h_n^m to a vector of $D_{IOB2} = 3$ probability scores that represent the word's affiliation to the corresponding tags I, O or B.

In our following experiments for aspect and opinion term extraction, this CNN architecture consists of a word embedding lookup table for $D_{word} = 100$ dimensional vectors, 3 convolution layers and a dense layer that is applied to each single sequence element. We chose a kernel size $l_{conv} = 3$ and $D_{conv} = 50$ feature maps with a rectified linear activation function. We do not employ a max pooling operation after convolutions in order to retain the initial sequence length. However, we employ dropout with a drop probability of 0.5 after each convolution as a regularization to prevent overfitting. The final dense layer uses a softmax activation function to compute probabilities for each of the 3 tags (I, O or B). The hidden layer sizes for this model are therefore 100-50-50-50 (including the embedding layer). Figure 2 depicts the architecture of this network.

[4] When using POS tags as additional features, we concatenate the corresponding one-hot vector p_n to the word embedding u_n and use the resulting sequence $[u']_1^N = \{(u_1 \oplus p_1), \ldots, (u_N \oplus p_N)\}$ in the next steps.

[5] Since we want to apply the convolution operation to the first and the last element in a sequence, too, we pad the input sequence with vectors of 0s.

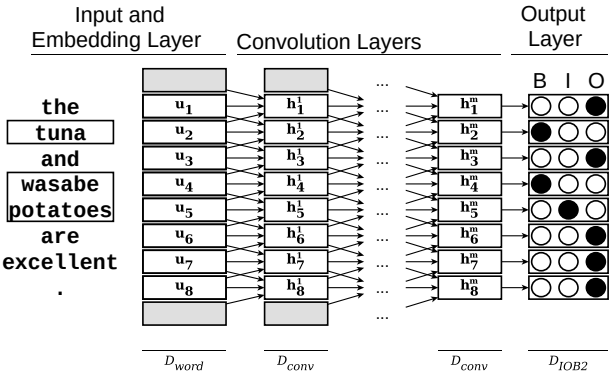


Figure 2. The CNN network for sequence labeling. The words marked with boxes are annotated aspect terms and are tagged with the correct IOB2 tags at the output layer. The gray vectors are padding vectors for the convolution operation.

3.3 Recurrent Neural Network Model

Besides the CNN based sequence tagger, we use an RNN based component to perform aspect and opinion term extraction. Sequence tagging with an RNN is much more straightforward due to the network's natural handling of sequential input data. In short, the RNN architecture transforms the input sequence of word indices into a sequence of hidden states using forward and recurrent connections. Analogously to the CNN approach, the produced hidden states are then mapped to tag probabilities for each of the 3 tags I, O, or B.

For this work, we chose the Gated Recurrent Unit Network (GRU, [2]) as the core of our recurrent architecture. It has been shown that the GRU is a competitive alternative to the well-known Long Short-Term Memory [9] despite its simpler architecture and less demanding computations [4].

Our RNN architecture comprises of an embedding layer for our $D_{word} = 100$ dimensional (pretrained) word embeddings, a GRU layer with $D_{gru} = 100$ hidden units, and a dense layer with a softmax activation applied to each single output vector of the GRU's output sequence. The hidden layer sizes for this component are therefore 100-100.

3.4 Stacked Model

The previous two sections describe neural network based models that we consider to tackle aspect and opinion term extraction. In this section, we propose a combination of the previous two models.

The intuition for combining the CNN and the RNN model is that both models present quite different approaches for the same task. While the CNN uses locally connected weights to compute localized features around a word of interest, the RNN uses recurrent connections. The latter connections make it possible to capture important pieces of information from preceding and potentially distant parts of the text. A combination of both models might benefit from both the local connectivity of the CNN and the recurrence of the RNN. Similar architecture combinations have been used by Zhou et al. [35] and He et al. [8].

We design the new combined model as follows. First, we apply convolutional layers to the input sequence, similar to the model in 3.2, yet only up to the final hidden layer. On top of this sequence

of high-level features, we stack a GRU layer that learns temporal dependencies of its input sequence. Again, a dense layer with a softmax activation is used to map the recurrent hidden states to tag probabilities. The stacked CNN-RNN model uses the hidden layer sizes 100-50-50-50-100 (or 146-50-50-50-100 when using additional POS tag features).

3.5 Joint Model

Klinger and Cimiano [12, 13] present models that are able to extract aspect and opinion terms jointly, leveraging knowledge about aspect terms in order to find opinion terms and vice versa. Using the stacked model as a basis, we try to replicate this behavior and construct a model that can infer aspects and opinions jointly.

This model has a very similar structure as the stacked architecture from the previous section. It differs, however, in a second output layer that we connect to the GRU layer. With that, the model is able to predict two tag sequences at once: one for extracted aspect terms, the other for opinion terms. See Figure 3 for a depiction of the joint architecture.

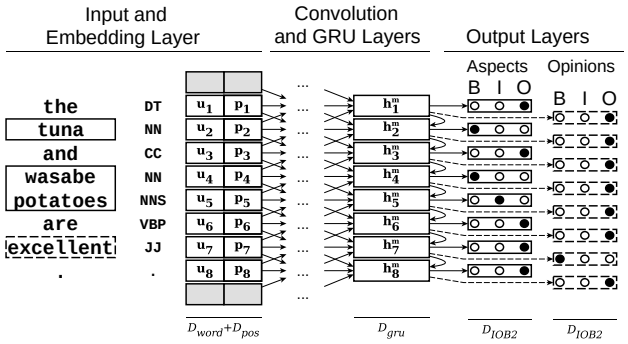


Figure 3. The CNN-RNN network for joint aspect and opinion term extraction. Solid boxes at the input mark aspect terms and are tagged with the correct tags at the aspect output layer (solid arrows and boxes). The dashed box at the input marks an opinion terms. The corresponding output layer is marked with dashed arrows and boxes.

4 Opinion Term Specific Sentiment Extraction

With the previously described models, we are able to detect mentioned aspect and opinion terms, however, without extracting the individual sentiment. The second step in our pipeline for relational sentiment extraction is to extract sentiments for these opinion terms. The difficulty here is that it is not enough to extract an overall sentiment for a given review. Rather, the sentiment needs to be extracted with respect to one of possibly several opinion terms in a text. In this section, we propose a recurrent neural network architecture together with a combination of word and distance embedding features to address this task.

Given an already detected opinion term in a review, we tag each word in the review with its relative distance to the opinion term in question, as shown in the following example:

Coffee	**stays**	**fresh**	and	*hot*	in	the	Carafe	(Text)
NNP	VBZ	JJ	CC	JJ	IN	DT	NN	(POS)
-1	0	0	1	2	3	4	5	(O)

where the bold words "**stays fresh**" form the opinion term for which we want to extract the sentiment. The italic word "*hot*" is another annotated opinion term that is neglected in this extraction step. Below, the sequence of corresponding POS tags (as obtained with the Stanford POS-tagger) and the relative word distances (O) to the opinion term are shown.

We extract a window of $l_{pol} = 20$ words centered around the opinion term instead of considering the whole review text. Sequences for review texts with less than l_{pol} words are padded at the left with 0s. Analogously, the sequence of POS tags is extracted. Using a subsequence of words and POS tags around each opinion term is reasonable since the lengths of the reviews in the USAGE corpus reach up to several hundred words. Taking the whole review text into account for all opinion terms is computationally very demanding.

We convert each word into its respective vector representation using the pretrained lookup table of word embeddings, thus obtaining a sequence of word vectors. Similar to this, we also use an embedding layer for the relative distances that provides us with a learned vector representation of dimensionality $D_{dist} = 10$ for each distance value, similar to the approach proposed by Zeng et al. [34]. While we did not evaluate those distance embeddings extensively, first results suggest that there is indeed a benefit in mapping the distances to real valued vectors instead of using the raw distance values. The benefits of distance and position embeddings are supported by the results of other works [21, 34, 29].

The individual vectors of the three sequences – word embeddings u_n, POS tags p_n and distance embeddings d_n – are concatenated, resulting in a single sequence of length l_{pol} with $D_{word}+D_{pos}+D_{dist}$ dimensional elements. We feed the resulting sequence to a recurrent neural network consisting of three layers. The first hidden layer is a GRU layer with $D_{gru} = 100$ hidden units that reads in the sequence of vectors and produces a sequence of hidden states. The second layer is a densely connected layer of maxout units [7] that transforms the final hidden state of the GRU layer into a vector h' of $D_{pol} = 100$ dimensions. Lastly, another maxout layer maps the previous hidden layer to 4 output units using a softmax activation function. Each of the 4 output units corresponds to one of four possible sentiments: `positive`, `neutral`, `negative` and `unknown`. The sentiment with the highest probability at the corresponding output unit is the predicted sentiment. Figure 4 visualizes the network.

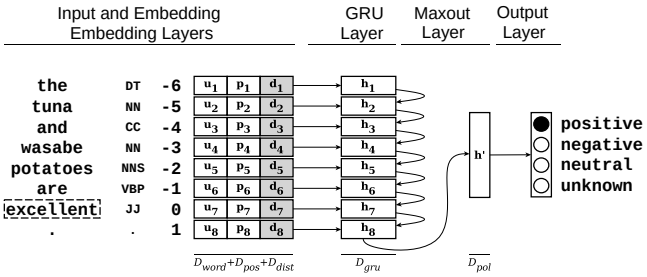


Figure 4. The component for opinion term specific sentiment extraction. The dashed box at the input marks an opinion term. Besides, the corresponding POS tags and the relative word distances are shown. Here, the input vectors are composed of three parts: one vector for the word embeddings, one vector for the POS tag encoding and one vector for the distance embedding. The output layer contains one unit for each possible sentiment label: `positive`,`neutral`, `negative` and `unknown`.

5 Aspect-Opinion Relation Extraction

This section introduces the component that is responsible for extracting relations between already extracted aspect and opinion terms.

Besides the annotations for aspect and opinion terms, the USAGE corpus provides annotations of relations between pairs of these aspects and opinions, which we simply refer to as aspect-opinion relations[6]. The presence of such a relation implies that the opinion term targets the annotated aspect term. Due to this binary nature of the relations (*present* or *not-present*) in the USAGE corpus, our relation extraction approach predicts a boolean tag for any given pair of aspect term and opinion phrase. This pairwise approach allows us to predict the many-to-many relations between aspect and opinion term that are present in the corpus.

Again, we employ a neural network based model that is very similar in structure to the component for opinion term specific sentiment extraction (see previous section). We adopt a similar strategy as presented in Zeng et al. [34] to address relation extraction. In contrast to Zeng et al. [34] however, we use a recurrent neural network instead of a convolutional architecture to perform the actual relation extraction.

Our approach employs four types of features for a given pair of aspect and opinion term:

- the sequence of word embeddings of length $l_{rel} = 20$ that is centered around the aspect and the opinion,
- the sequence of corresponding POS tags for each word,
- a sequence of relative distances of each word to the aspect term, and
- a sequence of relative distances of each word to the opinion term.

As a first step, our model computes the distance (in words) between the two terms and automatically rejects all pairs which are more than 20 words apart from each other. While this does reject some valid relations, we can still predict 98% of relations correctly for this maximum distance of 20 words. For those pairs which are below the maximum distance, our model extracts a subsequence of word embeddings of length 20, centered around the aspect term and opinion term. Sequences for review texts with less than 20 tokens are padded at the left with 0s. Analogously, the sequence of POS tags is extracted.

To encode information about the distance between the aspect and opinion phrases with respect to their position in the review, we follow a similar approach as Zeng et al. [34] and Nogueira dos Santos et al. [21]. We guide the network's attention to the targeted aspect and opinion terms by computing the relative distances of each word in the sequence to the aspect term and to the opinion term, respectively.

I	like	all	the	**different**	**features**	.	(Text)
PRP	VBP	PDT	DT	JJ	NNS	.	(POS)
-4	-3	-2	-1	0	0	1	(A)
-1	0	1	2	3	4	5	(O)

Here, the underlined word "like" marks an opinion term and the bold words "**different features**" mark the aspect term. Below, the corresponding sequence of POS tags is shown. The sequences labeled with A and O show the sequence of relative distances of each word to the aspect and opinion term, respectively.

[6] The creators of the USAGE corpus actually refer to these relations as TARG-SUBJ. To keep a consistent terminology in this work, we call them aspect-opinion relation.

Again, we use an embedding layer to obtain a sequence of $D_{dist} = 10$ dimensional embedding vectors for the relative distances. The individual vectors of the four sequences (word embeddings u_n, POS tags p_n, aspect distance embeddings d_n and opinion distance embeddings d'_n) are concatenated, resulting in a single sequence of length l_{rel} and $D_{word} + D_{pos} + 2 \cdot D_{dist}$ dimensional elements.

We feed the resulting sequence to a GRU layer with $D_{gru} = 100$ hidden units that computes a sequence of hidden states. The final hidden state is passed on to a densely connected layer h' of $D_{rel} = 100$ maxout units. As a last step, the output of the previous hidden layer is passed to a final fully-connected maxout layer with a single output unit and a sigmoid activation function to map its output to a value between 0 and 1. We interpret the network's output as the probability that the pair of aspect and opinion terms form an aspect-opinion relation. Figure 5 visualizes the network.

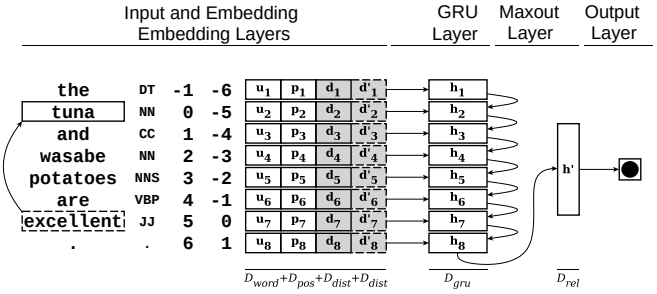


Figure 5. The component for aspect-opinion relation extraction. The solid and dashed boxes at the input mark aspect and opinion terms, respectively. Besides, corresponding POS tags and the relative word distances to the aspect and opinion term are shown. The input vectors are split into four parts: word embeddings, POS tag encoding, aspect distance embeddings and opinion distance embeddings. The output layer contains one single sigmoid unit. An output value > 0.5 signifies a relation between the aspect and opinion terms.

6 Experiments and Results

This section evaluates our proposed architecture for relational sentiment analysis. Our evaluation targets the individual components of the overall system, measuring their performances in isolation. For the evaluation, we consider the two datasets described in Section 2. Both datasets offer a different granularity in their annotations. The SemEval dataset does not provide annotations for opinion terms, opinion term specific sentiment or aspect-opinion relations. As such, we only evaluate our component for aspect term extraction on this dataset. The USAGE dataset, on the other hand, offers annotations for all our subtasks, allowing us to evaluate all our components on this dataset. Where possible, we give precision, recall and F_1 scores for our own and baseline approaches.

All experiments were performed with the deep learning library *Keras* [3] and employ many of its pre-implemented algorithms. Tag sequences predicted by our approach are post-processed to yield only sequences that are valid according to the IOB2 scheme.

6.1 Training

This section briefly outlines our training procedure. Since we deal with sequences of variable lengths, training our models in mini batches would require to pad shorter sequences with special padding elements. Due to the large differences in the length of reviews in the USAGE corpus (18 to $>$ 2000 words), we avoid padding sequences with too many padding elements. Instead, we train our models on one data sample at a time. The optimization of the models' weights is performed with *RMSProp* [30], which converges after a few epochs.

Since the performance did not depend strongly on the number of iterations over the training data, we trained all aspect and opinion term extraction models for 15 epochs. The component for sentiment extraction was trained for 14 epochs and the relation extraction component performed best with 28 epochs of training.

6.2 Initialization of Word Embeddings

In our first experiment, we compare the performance impact of initializing the weights of the word embedding lookup table in the stacked CNN-RNN model randomly to the initialization with our pretrained embeddings. Our intuition is that both sets of pretrained embeddings are helpful for the extraction of aspect and opinion terms but more so the domain-specific embeddings. These are expected to capture the most task relevant semantics which might help with the extraction task.

Since many of our network parameters are initialized randomly and our training data is processed in a random order, the performance of our model might be influenced by these external factors. To mitigate these effects, we performed each experiment three times and averaged the outcomes.

The experiments for aspect term extraction on the SemEval2015 dataset show that our model achieves on average an F_1 score of 0.581 with the randomly initialized embeddings. The initialization with the Wikipedia embeddings leads to a much higher score, namely $F_1 = 0.618$, while the domain-specific review embeddings yield an F-Measure of 0.646. As expected, we can observe a large benefit in pretraining our word embeddings on huge collections of natural language texts with the largest gain using domain-specific embeddings.

Considering these first results, we only use the domain-specific review embeddings as initialization in further experiments.

6.3 Evaluation: Aspect and Opinion Term Extraction

This section evaluates our relational sentiment analysis architecture focusing on the component for aspect and opinion term extraction. We perform the evaluation in two steps: First, we evaluate the component on the SemEval dataset measuring its performance for aspect term extraction only. Keep in mind that we can neither evaluate opinion term extraction nor opinion term specific sentiment analysis or aspect-opinion relation extraction on this particular dataset due to granularity of the provided annotation. Those subtasks, however, are evaluated in Sections 6.3.2, 6.4 and 6.5.

6.3.1 CNN vs. RNN vs. Stacked Models

This section evaluates our aspect term extraction component and considers different neural models. We train and test the CNN, the RNN and the stacked CNN-RNN model from the sections 3.2, 3.3 and 3.4 on the SemEval2015 dataset using the official training and test split. Again, we perform each experiment three times to alleviate differences due to the networks' initializations and training sample orders. Table 1 shows the average precision, recall and F_1 score for each model.

	Aspects		
Model	P	R	F_1
RNN	0.592	0.646	0.618
CNN	0.558	0.702	0.621
Stack	0.599	**0.703**	0.646
Stack+POS	**0.633**	0.689	0.659
EliXa	–	–	**0.701**

Table 1. Results for aspect term extraction on the SemEval2015 test dataset using different model architectures and additional POS tag features. *EliXa* represents the current state-of-the-art.

The model based on convolutional layers and the model based on recurrent layers both perform similarly regarding the overall F_1 score. Combining both types of models in a stack-like architecture results in an increased averaged F_1 score that is statistically significant with $p = 0.05$. Providing additional features in the form of POS tags does also improve the model's performance. While we still perform not quite as good as the current state-of-the-art system EliXa [25] we do perform well with respect to the overall ranking of the SemEval2015 task as can be seen in Pontiki et al. [22].

In spite of being a few percentage points below the current state-of-the-art system EliXa, our proposed method still constitutes a meaningful contribution. The benefit of our method is that it is not restricted to the mere extraction of aspect terms but that it is capable of extracting aspect and opinion terms jointly. The following section shows that our method achieves state-of-the-art performance on the USAGE dataset with this more complex joint extraction.

Based on the results of these experiments, we perform all following aspect and opinion phrase extraction tasks using the CNN-RNN model with additional POS tag features.

6.3.2 Joint vs. Separate Models

This section investigates the benefits of predicting aspect and opinion phrases of the USAGE corpus jointly in one model, in contrast to two separate models. For this, we consider two joint models with different hidden layer sizes, namely with 100-50-50-50-100 (dubbed *Joint small*) and 100-100-100-100-200 (dubbed *Joint large*). This should account for differences in performance that solely result from different network capacities. As proposed by Klinger and Cimiano [14], we evaluate our models by 10-fold cross validation. Table 2 shows the results for the joint and the separate models.

We can see that extracting aspect and opinion terms jointly does indeed enhance the model's performance, but only so for a larger network configuration. The extraction of opinion terms benefits from the joint setting in particular. However, this might also be attributed to the increased size of the hidden layers in our neural architecture. The extraction of opinion terms might simply require more network parameters which it is able to claim in the larger joint architecture. A more detailed investigation needs to be conducted in future work.

Bear in mind that we do not compare our results on this dataset with the EliXa system referenced in Section 6.3.1. The EliXa system is applicable to aspect term extraction in isolation and is not designed for opinion term extraction.

	Aspects			Opinions		
Model	P	R	F_1	P	R	F_1
Klinger2014	–	–	0.56	–	–	0.48
Aspect only	0.63	**0.70**	0.66	–	–	–
Opinion only	–	–	–	0.44	0.48	0.45
Joint small	0.57	0.65	0.61	0.40	0.40	0.40
Joint large	**0.65**	0.69	**0.67**	**0.47**	**0.53**	**0.50**

Table 2. Performances for aspect and opinion phrase extraction on the USAGE dataset for joint and separate models. *Joint small* and *Joint large* are the joint models with hidden layer size 100-50-50-50-100 and 100-100-100-100-200, respectively.

6.4 Evaluation: Opinion Term Specific Sentiment Extraction

Next, we show our evaluation of the component proposed in Section 4. The model considers individual opinion terms in a wide context and predicts one of four sentiment labels for each presented opinion term.

We perform the sentiment extraction on the opinion terms of the gold standard annotations of the USAGE corpus in order to measure the sentiment extraction in isolation. To keep the experiments consistent across our different components, we perform the prediction on a 10-fold cross validation of the USAGE documents. The results in Table 3 show the average accuracy[7] of predicting the sentiment label of an opinion term with the number of correctly and incorrectly labeled opinion terms. We also show the results of a naive approach that always predicts the (most frequent) sentiment label `positive` to act as a simple baseline.

Model	$Accuracy$	$\#Correct$	$\#Incorrect$
Positive Only	0.647	342.6	189.5
Our Approach	**0.831**	**441.6**	**90.5**

Table 3. $Accuracy$ for opinion term specific sentiment extraction on gold annotations.

We see that our method achieves an accuracy high above our baseline. Unfortunately, up to date, no results are published for sentiment extraction on the USAGE dataset that we can use as a further baseline. Hence, with this work, we contribute the first results for opinion term specific sentiment extraction for this dataset.

6.5 Evaluation: Aspect-Opinion Relation Extraction

This part of our evaluation focuses on the last component in the overall architecture for relational sentiment analysis that is responsible for the extraction of aspect-opinion relations. As before, we perform the relation extraction on the aspect and opinion terms from the gold standard annotations of the USAGE corpus, in order to measure our system's performance for relation extraction in isolation. This methodology is adopted from Klinger and Cimiano [14] and allows us to compare our method to their work. Table 4 shows the results of a 10-fold cross-validation of our proposed component. The

[7] We report accuracy since precision, recall and F_1 score are all equal in the case where the number of annotations is fixed.

Model	P	R	F_1
Klinger2014	–	–	0.65
Our Approach	0.87	0.75	**0.81**

Table 4. F_1 score for aspect-opinion relation extraction on gold annotations.

results show that our RNN-based model improves relation extraction by 15% F-Measure compared to the probabilistic graphical model of proposed by Klinger and Cimiano [14].

7 Conclusion and Future Work

In this work, we presented a modular architecture that addresses sentiment analysis as a relation extraction problem. The proposed architecture divides the problem into three subtasks and addresses each with a dedicated component. This highly flexible approach offers a fine-grained solution for sentiment analysis.

As part of this overall architecture, we presented possible implementations for the individual components: First, we presented different neural network models that are capable of aspect and opinion term extraction and which achieved competitive and state-of-the-art results on different datasets. We could report a benefit for this task in using domain-specific word embeddings compared to domain-independent and randomly initialized embeddings. We investigated the extraction of aspect and opinion terms separately and jointly and found the joint approach to produce superior results in one setting. Thus, we confirm previous results from Klinger et al. [12] who used a probabilistic graphical model instead of a neural network model.

Secondly, we addressed opinion term specific sentiment extraction with a recurrent neural network model and distance embedding features and achieved promising results; the first sentiment extraction results on the considered dataset.

Thirdly, as another contribution, we designed and evaluated a model for the extraction of relations between aspect and opinion terms which outperformed prior results on the same dataset. Our work shows that it is possible to divide sentiment analysis in a flexible and fine-grained way using a highly modular architecture.

All proposed components stand out by their minimal use of hand-engineered features that are strongly tuned to their specific tasks. The only external resources that were used are machine-generated POS tags and word embeddings which were created with a data-driven approach. Nevertheless, all components perform competitive on their individual subtasks. It is easily conceivable to provide further task-specific features to improve the performances of the individual components even further. This, however, is not the goal of this work, which is why we leave this part for future work.

Furthermore, we expect our components to benefit from bidirectional recurrent connections [28] so that words appearing later in a sentence can be taken into account. The investigation of attention mechanisms for RNNs [1] seems also very promising to allow the models to focus more strongly on important parts of the input sequence.

ACKNOWLEDGEMENTS

This work 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

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, 'Neural machine translation by jointly learning to align and translate', *CoRR*, **abs/1409.0473**, (2014).

[2] Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, '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)*, pp. 1724–1734, Doha, Qatar, (October 2014). Association for Computational Linguistics.

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

[4] Junyoung Chung, Çalar Gülçehre, Kyunghyun Cho, and Yoshua Bengio, 'Empirical evaluation of gated recurrent neural networks on sequence modeling', Technical report, Université de Montréal, (2014). Presented at the NIPS Deep Learning workshop.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, 'Natural language processing (almost) from scratch', *Journal of Machine Learning Research*, **12**, 2493–2537, (2011).

[6] Erick Rocha Fonseca and João Luís Garcia Rosa, 'A two-step convolutional neural network approach for semantic role labeling', in *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, (2013).

[7] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C. Courville, and Yoshua Bengio, 'Maxout networks', in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1319–1327, (2013).

[8] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang, 'Reading Scene Text in Deep Convolutional Sequences', in *Proceedings of the 13th Conference on Artificial Intelligence*, (2016).

[9] Sepp Hochreiter and Jürgen Schmidhuber, 'Long short-term memory', *Neural Computation*, **9**(8), 1735–1780, (November 1997).

[10] Ozan İrsoy and Claire Cardie, 'Opinion mining with deep recurrent neural networks', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 720–728, (2014).

[11] Yoon Kim, 'Convolutional neural networks for sentence classification', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, (2014).

[12] Roman Klinger and Philipp Cimiano, '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)*, pp. 848–854, (August 2013).

[13] Roman Klinger and Philipp Cimiano, 'Joint and pipeline probabilistic models for fine-grained sentiment analysis: Extracting aspects, subjective phrases and their relations', in *13th IEEE International Conference on Data Mining Workshops (ICDM)*, pp. 937–944, (December 2013).

[14] Roman Klinger and Philipp Cimiano, 'The USAGE review corpus for fine grained multi lingual opinion analysis', in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pp. 2211–2218, (May 2014).

[15] Himabindu Lakkaraju, Richard Socher, and Chris Manning, 'Aspect Specific Sentiment Analysis using Hierarchical Deep Learning', *NIPS Workshop on Deep Learning and Representation Learning*, (2014).

[16] Qv Le and Tomas Mikolov, 'Distributed Representations of Sentences and Documents', in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 1188–1196, (2014).

[17] Pengfei Liu, Shafiq Joty, and Helen Meng, 'Fine-grained opinion mining with recurrent neural networks and word embeddings', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1433–1443. Association for Computational Linguistics, (September 2015).

[18] J. J. McAuley, R. Pandey, and J. Leskovec, 'Inferring networks of substitutable and complementary products', in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, (2015).

[19] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, 'Image-based recommendations on styles and substitutes', in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, (2015).

[20] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean, 'Efficient Estimation of Word Representations in Vector Space', *Workshop at the International Conference on Learning Representations (ICLR)*, (2013).

[21] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou, 'Classifying Relations by Ranking with Convolutional Neural Networks', in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 626–634, (2015).

[22] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos, 'Semeval-2015 task 12: Aspect based sentiment analysis', in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, pp. 486–495, Denver, Colorado, (June 2015). Association for Computational Linguistics.

[23] Soujanya Poria, Erik Cambria, and Alexander Gelbukh, 'Deep Convolutional Neural Network Textual Features and Multiple Kernel Learning for Utterance-Level Multimodal Sentiment Analysis', *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2539–2544, (2015).

[24] Radim Řehůřek and Petr Sojka, 'Software Framework for Topic Modelling with Large Corpora', in *Proceedings of the LREC Workshop on New Challenges for NLP Frameworks*, pp. 45–50, (2010).

[25] Iñaki San Vicente, Xabier Saralegi, and Rodrigo Agerri, 'Elixa: A modular and flexible ABSA platform', in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, pp. 748–752, Denver, Colorado, (June 2015). Association for Computational Linguistics.

[26] Cicero D Santos and Bianca Zadrozny, 'Learning character-level representations for part-of-speech tagging', in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 1818–1826, (2014).

[27] Cicero Nogueira Dos Santos and Victor Guimarães, 'Boosting Named Entity Recognition with Neural Character Embeddings', in *Proceedings of the Fifth Named Entity Workshop, joint with 53rd ACL and the 7th IJCNLP*.

[28] M. Schuster and K. K Paliwal, 'Bidirectional recurrent neural networks', *IEEE Transactions on Signal Processing*, **45**(11), 2673–2681, (1997).

[29] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang, 'Modeling mention, context and entity with neural networks for entity disambiguation', in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1333–1339, (2015).

[30] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[31] Erik F. Tjong Kim Sang and Jorn Veenstra, 'Representing text chunks', in *Proceedings of the 9th Conference on European Chapter of the Association for Computational Linguistics (EACL)*, pp. 173–179, (1999).

[32] Zhiqiang Toh and Wenting Wang, 'DLIREC: Aspect Term Extraction and Term Polarity Classification System', in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*, pp. 235–240, (2014).

[33] Kristina Toutanova, Dan Klein, and Christopher D Manning, 'Feature-rich part-of-speech tagging with a cyclic dependency network', in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 252–259, (2003).

[34] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao, 'Relation Classification via Convolutional Deep Neural Network', *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2335–2344, (2014).

[35] Xiaoqiang Zhou, Baotian Hu, Jiaxin Lin, Yang Xiang, and Xiaolong Wang, 'ICRC-HIT: A Deep Learning based Comment Sequence Labeling System for Answer Selection Challenge', *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, (2013), 210–214, (2015).