MNCN: A Multilingual Ngram-Based Convolutional Network for Aspect Category Detection in Online Reviews

Erfan Ghadery, Sajad Movahedi, Heshaam Faili, Azadeh Shakery
University of Tehran
Tehran, Iran
{erfan.ghadery, s.movahedi, hfaili, shakery}@ut.ac.ir

Abstract

The advent of the Internet has caused a significant growth in the number of opinions expressed about products or services on e-commerce websites. Aspect category detection, which is one of the challenging subtasks of aspect-based sentiment analysis, deals with categorizing a given review sentence into a set of predefined categories. Most of the research efforts in this field are devoted to English language reviews, while there are a large number of reviews in other languages that are left unexplored. In this paper, we propose a multilingual method to perform aspect category detection on reviews in different languages, which makes use of a deep convolutional neural network with multilingual word embeddings. To the best of our knowledge, our method is the first attempt at performing aspect category detection on multiple languages simultaneously. Empirical results on the multilingual dataset provided by SemEval workshop demonstrate the effectiveness of the proposed method[1].

Introduction

The Internet has caused an explosion in the amount of data produced daily such as user-generated reviews on e-commerce websites. People are often interested in the experiences and opinions of others about different products or services expressed on these e-commerce websites like Amazon [2] and Yelp [3]. These user-generated reviews are valuable resources for researchers in natural language processing fields, especially sentiment analysis. Aspect category detection is a subtask of aspect-based sentiment analysis, which deals with categorizing the review sentences based on a set of pre-defined aspect categories. These aspect categories are common concerns for the consumers. For example in the restaurant domain, aspect categories can be 'FOOD', 'SERVICE', etc.

Previous works in the literature that address aspect category detection are mainly devoted to the English language. However, with the growth of the Internet and the advent of the international market, consumers write reviews in different languages. Only addressing English reviews lead to missing a lot of information existing in other languages. A multilingual method can have two main advantages: First, by using a multilingual method, we can extract information from reviews in different languages simultaneously which makes the analysis more comprehensive and precise. Second, in situations where there are a few or no labeled data available in a particular language, available data in other languages can help in providing a model for aspect category detection in the low-resource language. In this paper, we propose a multilingual method that performs aspect category detection for several languages simultaneously by utilizing multilingual word embeddings (Conneau et al. 2017) which is available for many languages. Our proposed method is able to classify review sentences of a language even when there are no labeled data available for that language by utilizing the available labeled data in other languages.

Many multilingual approaches have been proposed for document-based sentiment analysis, where the goal is determining the polarity of a document like a tweet or a user comment (Banea et al. 2008),(Brooke, Tofiloski, and Taboada 2009). Cross-lingual sentiment analysis is the most common approach for handling this problem, where machine translation techniques are utilized for translating the documents into a target language and performing sentiment analysis in a monolingual fashion using the resources in the target language. However, machine translation-based approaches suffer from various problems such as the effect of cultural differences between the source and the destination language (Duh, Fujino, and Nagata 2011), the quality of the translation method, and the possible sparseness and noisiness of the data (Balahur and Turchi 2012).

Our proposed multilingual method, MNCN, does not need any translation techniques, and performs aspect category detection using an ngram-based deep convolutional neural network with the multilingual word embeddings as the input of the network. Initially, as the sentence representation, we construct a matrix of multilingual word embeddings whose rows are embedding vectors of the sentence words, and the columns are the embedding features. Then, three parallel ngram-based convolution layers convolve with the sentence representation matrix to extract ngram features. These obtained features are then concatenated together to

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] The code for this paper is available at https://github.com/erfan-ghadery/MNCN

[2] https://www.amazon.com/

[3] https://www.yelp.com/

provide the features to be fed into the three parallel dense feedforward network with three different output targets. We consider three objectives for the network: predicting the 'Entity' label, predicting the 'Attribute' label, and predicting the aspect category label which is the combination of the 'Entity' and 'Attribute' labels in the form of 'Entity#Attribute', like 'FOOD#QUALITY'.We believe that considering the 'Entity' and 'Attribute' prediction as the objective of the network plays the role of regularization to avoid overfitting. Several experiments conducted on multilingual datasets of SemEval-2016 (Pontiki et al. 2016) in the restaurant domain demonstrate the effectiveness of MNCN in multilingual aspect category detection.

To summarize, the main contributions of this paper are as follows. First, to the best of our knowledge, our work is the first attempt in addressing aspect category detection in a multilingual fashion. Second, we propose a deep convolutional network architecture with three different objective functions at the same time to address aspect category detection. Finally, we conduct extensive experiments to investigate the effectiveness of the MNCN in multilingual aspect category detection.

The rest of the paper is organized as follows: the next section reviews related works. Then we will describe the details of the proposed method. After describing our method, we will discuss experiments and results. Finally, the last section contains the conclusion and the proposed future directions.

Related Work

Aspect category detection as a subtask of aspect-based sentiment analysis was first introduced by SemEval workshops in (Pontiki et al. 2014). In recent years, numerous works have addressed aspect-based sentiment analysis. Earlier works were more dependant on approaches such as frequent pattern mining, dependency relations, and lexical patterns (Hu and Liu 2004) ,(Qiu et al. 2011), (Popescu and Etzioni 2007). Later, authors proposed to use machine learning approaches for this task. In (Kiritchenko et al. 2014) several features such as lexicon features and ngrams were used to train a set of one-vs-all SVMs for each aspect category. (Toh and Su 2016) used the output of a convolutional neural network along with several other features as features for a set of one-vs-all linear classifiers for each aspect category.

There are no multilingual methods that address aspect category detection. Some research efforts are devoted to multilingual methods in document-based sentiment analysis, where the goal is to determine the polarity ('Positive' or 'Negative') of the texts at the document level. The most common approach for multilingual sentiment analysis at the document level is cross-lingual sentiment analysis which is mainly based on machine translation techniques (Banea et al. 2008). The main idea behind these works is usually translating all the data into a target language and performing monolingual classification using resources in the target language. In (Mihalcea, Banea, and Wiebe 2007), authors proposed to use a bilingual lexicon and a manually translated parallel corpus to provide the required resources for training a subjectivity classifier in a new language. (Brooke, Tofiloski, and Taboada 2009) used translation to transfer knowledge from English to Spanish and then by utilizing a lexicon-based approach and machine learning tried to perform document sentiment classification in the Spanish language. Unlike these translation-based methods, our method does not need any translation effort to perform multilingual aspect category detection.

Some authors proposed multilingual methods to address sentiment analysis without using machine translation techniques. (Zhang and LeCun 2015) introduced a character based convolutional neural network with two convolutional layers. In this word, the first convolutional layer is assumed to act as a word embedding learning layer. Following this character based architecture, in (Wehrmann et al. 2017) a character based convolutional neural network was introduced utilizing a single convolutional layer to address twitter sentiment analysis in a multilingual fashion without the use of any translation. This method is the closest method to our approach with two differences: first, we utilized the multilingual word embeddings as the sentence representation instead of characters. Second, the aforementioned method addresses the document-based sentiment analysis, while our method performs aspect category detection.

Earlier, it was noticed by (Mikolov, Le, and Sutskever 2013) that word embeddings across different languages exhibit similar behaviors. Following this observation, they proposed to exploit this similarity by learning a linear mapping from a source embedding to a target embedding, by using a parallel vocabulary of five thousand words as anchor points. Other studies have tried to improve cross-lingual word embeddings, using this idea (Smith et al. 2017) (Artetxe, Labaka, and Agirre 2016). All these works suffer from the need of parallel data as anchor points. Later, (Conneau et al. 2017) introduced a method to learn the mappings without a parallel corpus. In this paper, we utilize word embeddings trained using this method for word representation.

Model Description

In this section, we introduce the design of our Multilingual Ngram-based Convolution Network (MNCN). The overall architecture of MNCN is depicted in figure 1. Our model is inspired by (Kim 2014) and is composed of several components: a sentence representation component, three parallel convolution layers, and three parallel dense layers. Details of each component is described in the following subsections.

Sentence Representation with Multilingual Word Embeddings

For multilingual word embeddings, we use the word vectors provided by the MUSE library following (Conneau et al. 2017). Using an adversarial learning setting, (Conneau et al. 2017) have proposed an unsupervised method to create word translation pairs. Given the set of monolingual word embeddings from the source language, and the set of monolingual word embeddings from the target language, using an adversarial algorithm, this method initially performs two tasks. It first learns a projection matrix between the two embedding spaces. Then a multi-layer perceptron model is trained that

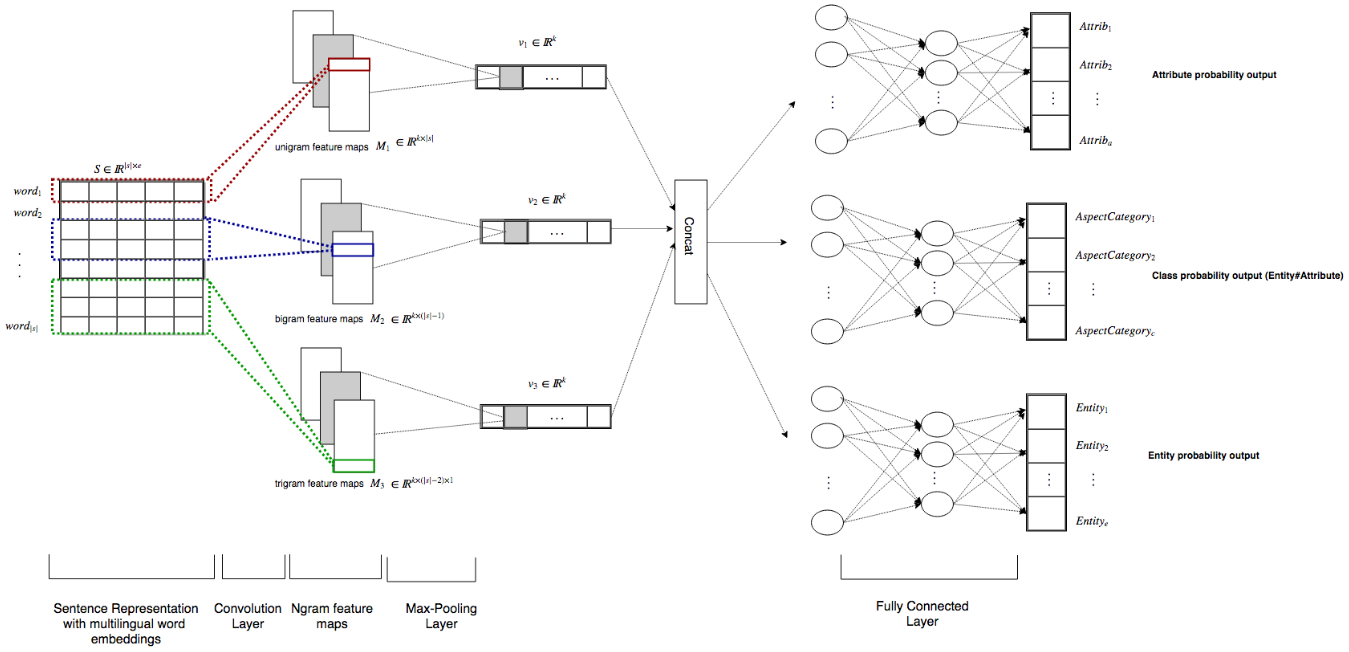

$S \in \mathbb{R}^{|s| \times e}$
$word_1$
$word_2$
$word_{|s|}$
unigram feature maps $M_1 \in \mathbb{R}^{k \times |s|}$
bigram feature maps $M_2 \in \mathbb{R}^{k \times (|s|-1)}$
trigram feature maps $M_3 \in \mathbb{R}^{k \times (|s|-2) \times 1}$
$v_1 \in \mathbb{R}^k$
$v_2 \in \mathbb{R}^k$
$v_3 \in \mathbb{R}^k$
Concat
$Attrib_1$
$Attrib_2$
$Attrib_a$
Attribute probability output
$AspectCategory_1$
$AspectCategory_2$
$AspectCategory_c$
Class probability output (Entity#Attribute)
$Entity_1$
$Entity_2$
$Entity_e$
Entity probability output
Sentence Representation with multilingual word embeddings
Convolution Layer
Ngram feature maps
Max-Pooling Layer
Fully Connected Layer

Figure 1: The architecture of our proposed model.

given an embedding vector, predicts whether the vector belongs to the source language embedding space or the destination language embedding space. In the next step, the projection matrix is used for finding the mutual nearest neighbor words between the vocabulary of the two languages. Finally, using an SVD based method similar to (Smith et al. 2017), the obtained mapping is used for finding a refined projection function. This method aligns the word embeddings of the source and the destination languages in the same vector space. For example, considering the source language as English and the destination language as Spanish, in the new embedding space the words "cat" in English and "gatto" in Spanish should be relatively close.

For a given sentence, we use the multilingual word embeddings to represent the sentence in the multilingual embedding vector space. Therefore, the representation of the sentence is a 2-dimensional vector $S \in \mathbb{R}^{|s| \times e}$ where $|s|$ is the length of the sentence and e is the embedding size. The vector representations are then padded to a fixed length.

Ngram-based Convolutional Layer

We consider three parallel convolutional layers to capture ngram features from the input sentence. Let $S_i \in \mathbb{R}^e$ be the word vector of the i^{th} word in the input sentence. We define $F \in \mathbb{R}^{n \times e}$ as the convolution filter where n is the size of ngram. The filter F convolves with the ngram at every position in the input sentence which allows us to extract features independent of their position in the sentence. We convolve filter F with the word window $S_{i:i+n-1}$ with a stride of 1 which results in a column feature map vector $fm \in \mathbb{R}^{|s|-n+1}$. Each element fm_i is calculated as follows:

$$fm_i = f(S_{i:i+n-1} \odot F + b_0) \qquad (1)$$

where f is a non-linear activation fucntion (we used ReLU), $\odot$ is pair-wise multiplication, and b_0 is a bias term. For each ngram size, k filters is applied to the input sentence to generate a matrix of feature maps $M \in \mathbb{R}^{k \times (|s|-n+1)}$ where k is the number of filters.

In order to obtain unigram, bigram, and trigram feature map matrices, we apply three set of k convolution operations between sentence representation matrix and kernel matrices $F \in \mathbb{R}^{1 \times e}$, $F \in \mathbb{R}^{2 \times e}$ and $F \in \mathbb{R}^{3 \times e}$ respectively. The convolution operations results in feature map matrices $M_1 \in \mathbb{R}^{k \times |s|}$, $M_2 \in \mathbb{R}^{k \times (|s|-1)}$, and $M_3 \in \mathbb{R}^{k \times (|s|-2)}$ for unigram, bigram and trigram convolutions respectively.

After obtaining the feature map matrices, we apply a max-over-time pooling operation following (Collobert et al. 2011) over each of the matrices M_1, M_2, and M_3 separately to capture the most important features among the unigram, bigram, and trigram features. This operation results in a vector $v \in \mathbb{R}^k$ for each of the feature map matrices where k is the number of feature maps. Finally, three vectors v_1, v_2, and v_3 obtained by the max-pool operation are concatenated together to provide the sentence representation.

Aspect Category Detection

We consider three objectives for our model:

- Predicting the aspect category of the input
- Predicting the entity of the input
- Predicting the attribute of the input

For each aspect category, we also consider the entity and attribute of the aspect category (for example both food and price for FOOD#PRICE) as well for regularization effects. Therefore, after concatenating the max-pooled output of the convolutional layers to form a sentence representation vector, we feed this vector to three Multi-Layer Perceptrons (MLP) to predict the aspect category, the entity, and the attribute of the input as demonstrated in figure 1. These MLPs are composed of a hidden layer of size h and an output layer of size of the number of classes.

During the test phase, we only use the aspect category predictions of the network and discard the attribute and entity predictions. If the probability of an aspect category surpasses a threshold, we assign the category to the given sentence. We optimize the threshold on the validation set by a simple grid search.

Training Objective

In order to maximize our objective functions, we use the Mean Square Error loss function. Given the weights of the model as θ, the Mean Square Error loss function can be formalized as follows:

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \qquad (2)$$

where n is the number of elements to calculate the error on, y_i is the ground truth value of the i^{th} element, and $\hat{y_i}$ is the predicted value for the i^{th} element. Overall, we define the loss function of our model as follows:

$$\mathcal{J}(\theta) = \mathcal{L}_{AC}(\theta) + \alpha\mathcal{L}_{AT}(\theta) + \beta\mathcal{L}_{EN}(\theta) \qquad (3)$$

where L_{AC} is the mean square error on the aspect category labels, L_{AT} is the mean square error on the attribute labels, L_{EN} is the mean square error on the entity labels, and α and β are the coefficients of the regularization objectives. We optimized these coefficients using grid search.

Following (Shrivastava, Gupta, and Girshick 2016), we utilized online hard example sampling in order to make the training of our model more efficient and effective. This method helps the model to focus on harder samples of the training data and prevent the model from overfitting on the easier samples. Therefore, for each minibatch, our model is trained on the top k percent of the samples with the largest loss values of the minibatch. We set k to 40.

Experiments

Datasets

We used SemEval-2016 task 5 dataset in the restaurant domain in our experiments. To evaluate our proposed method, we used the SemEval-2016 dataset for 4 languages English (Pontiki et al. 2016), Dutch (De Clercq and Hoste 2016), Spanish, and French (Apidianaki, Tannier, and Richart 2016). The detailed statistics of the 4 datasets are summarized in Table 1.

During the training phase, we discarded objective sentences that do not contain any opinions and therefore do not belong to any of the aspect categories.

Language	Train		Test	
	#Sent.	#Tuples	#Sent.	#Tuples
English	2000	2507	676	859
Dutch	1711	1860	575	613
French	1733	2530	696	954
Spanish	2070	2720	881	1072
Total	7514	9617	2828	3498

Table 1: Datasets statistics. The first column for train and test data is the number of sentences and the second column the number of aspect - sentiment tuples.

Baseline Methods

In order to show the effectiveness of our model, we implemented multiple baselines to compare our method with. The baseline methods are as follows:

- **Conv-Emb.** In order to show the effect of using the multilingual word embeddings, we train our convolutional model on word embeddings initialized randomly from a uniform distribution. The word embeddings are part of the trainable parameters of the network in this model.
- **Conv-Emb-Freeze.** This model is similar to Conv-Emb with the difference that the random word embeddings are freezed.
- **SVM.** This method is an SVM classifier similar to the baseline provided by (Pontiki et al. 2016). We provide different ngram features as input features for the SVM classifier. The performance of the SVM baseline method is evaluated using unigram (SVM-U), bigram (SVM-B), trigram (SVM-T), unigram+bigram (SVM-UB), and unigram+bigram+trigram (SVM-UBT) as features. We train a set of one-vs-all SVMs for each category.
- **Translation-Conv.** This method is implemented to show the effectiveness of using multilingual word embeddings compared to using machine translation techniques. In this baseline we translate all the datasets to English using Google Translate [4] and utilize word embeddings trained on the unlabeled Yelp Dataset challenge [5] using the skip-gram method (Mikolov et al. 2013). Then we train our convolutional model on the translated data. In order to make fair comparison, we optimized all the hyperparameters for this baseline as well.
- **Conv-Char-S.** We adapted the Conv-Char-S proposed in (Wehrmann et al. 2017) to our task. We gathered the alphabet of the languages English, French, Spanish, and Dutch and implemented a convolutional layer that provides embeddings followed by a max pooling layer and a hidden layer, similar to MNCN.

Experiment Settings

As the evaluation metric, we use F1 measure, precision, and recall. The pre-processing step is composed of stop-word and punctuation removal using the NLTK toolkit in python (Bird, Klein, and Loper 2009). In the Translation-Conv baseline, the pre-processing step is done after the translation phase. For the randomely initialized word embeddings used in Conv-Emb and Conv-Emb-Freeze, we used numpy (Oliphant 2006). For the multilingual word embeddings, we

[4] http://translate.google.com/

[5] https://yelp.com/dataset/challenge

used the pre-trained word embeddings provided by MUSE (Conneau et al. 2017). The word embeddings used in the translation baseline for the English language is trained on the Yelp challenge dataset using skip-gram (Mikolov et al. 2013). The size of all the word embeddings used in this paper are 300 and all the other parameters are set to default values. For the monolingual word embeddings of other languages, we used the word embeddings provided by (Ruder, Ghaffari, and Breslin 2016).

We select 10 percent of the train data for validation. The validation data is picked uniformly from each of the categories and each of the languages. All the hyperparameters of our model are tuned using the validation data via grid search. The best kernel size was chosen from 32, 64, 128, 256, 512, and 1024. We select the optimum dropout probability from 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. The best attribute and entity coefficients are also found from 0 to 1 with 0.1 steps. We found the optimum learning rate from 1×10^{-2}, 1×10^{-3} , 2×10^{-3}, 3×10^{-3} and 4×10^{-3}.

We trained all the deep neural network models using a minibatch size of 128 and Adam optimizer with a learning rate of 2×10^{-3}, except for Conv-Emb, Conv-Emb-Freeze, and Conv-Char-S for which we used the learning rate of 1×10^{-3}. The number of kernels in Conv-Char-S was set to 256 instead of 128 because of the poor performance of the model. Also, we set the drop out probability and the kernel size in Conv-Char-S to 0.9 and 140 respectively according the (Wehrmann et al. 2017). All the model are trained for a maximum of 100 epochs for which early stopping is performed with patience set to 10.

All the deep neural network baselines were implemented using PyTorch (Paszke et al. 2017) version 0.4.1. All the experiments were performed on an Nvidia GeForce GTX 1080. For training word embeddings, the gensim package (Řehůřek and Sojka 2010) was utilized. We used the implementation of (Pedregosa et al. 2011) with an RBF kernel and default parameters for the SVM baselines.

Method	P(%)	R(%)	F1(%)
SVM-T	25.39	67.30	36.87
SVM-B	26.36	71.61	38.54
SVM-U	25.57	**91.83**	40.00
SVM-UB	28.19	90.01	42.93
SVM-UBT	29.84	88.58	44.64
Conv-Char-S	48.78	40.97	44.53
Conv-Emb-Freeze	59.71	58.74	59.22
Conv-Emb	60.14	61.10	60.62
Translation-Conv	**66.49**	68.12	67.29
MNCN	65.48	73.21	**69.13**

Table 2: F1-score, precision, and recall of MNCN compared to different baselines. The MNCN result is statistically significant with $p < 0.05$ based on one-tailed t-test.

Evaluation and Results

In this section, we conduct several experiments for evaluating the effectiveness of MNCN. We try to answer two questions:

Method	English	French	Spanish	Dutch
Conv-Char-S	37.49	36.76	47.50	36.24
SVM-UBT	48.03	44.67	47.57	41.15
Conv-Emb	61.07	60.23	64.58	56.86
MNCN	**71.77**	**65.43**	**69.64**	**64.81**

Table 3: F1-score per-language.

- Is it able to effectively classify a set of reviews in different languages simultaneously?
- Is it able to classify a set of reviews in a specific language using training data in other languages?

For the first question, we combine all the datasets in English, French, Dutch, and Spanish in a single dataset. Then, we perform aspect category detection on this single combined dataset and compare the results of MNCN against several baselines including a translation-based baseline. Note that, for the Translation-Conv baseline we translate all the French, Dutch, and Spanish datasets to English and then combine them with English dataset and perform aspect category detection on them. We used to word embeddings trained on an unlabeled English corpus in the restaurant domain as the word representation in this baseline. In our experiments, we found that the stability of MNCN is dependent on the distribution of the sentences of different languages in the mini-batches for training the model. More specifically, the model is sensitive to the order in which we feed train data in different languages. In order to preserve the robustness of the reported results of MNCN, in all the experiments (except for the per-language experiment shown in table 3, for which the train and the test data both are in the same language) we repeat the training process for every permutation of the order of languages in the train data and report the average result of all the executions. We perform a similar strategy for Translation-Conv and Conv-Char-s for a fair comparison.

Table 2 shows the results in terms of precision, recall, and F1-score. MNCN achieves results superior to the other baselines in terms of F1-score, which indicates the effectiveness of MNCN in performing aspect category detection in several languages simultaneously. The highest recall is gained by SVM-U baseline, but considering the very low precision of this baseline, it is obvious that SVM-U approximately assigns all categories to all test sentences, which leads to a high recall and low precision. The SVM-UBT baseline achieves better result among SVM-based baselines. One of the reasons for the poor performance of the SVM-based baselines is the increasing sparsity of the feature representation when considering multiple languages. The poor performance of the Conv-Char-S baseline might be because this method was proposed for polarity detection in document-based sentiment analysis and adapting this method for performing aspect category detection - which is a more complex task - may yield worse results. Comparing the results of Conv-Emb against Conv-Emb-Freez shows that training the word vectors during the training yields better performance. Also, we observe that the Translation-Conv baseline is quite competitive with MNCN. However, this baseline requires a

Train	Test	MNCN	Translation-Conv
En+Fr+Du+Sp	Sp	69.89	**70.20**
En+Fr+Sp+Du	Du	**66.72***	65.44
En+Du+Sp+Fr	Fr	**65.74***	64.86
Du+Sp+Fr+En	En	**73.73***	71.65

Table 4: The F1-score obtained by MNCN and Translation-Conv when the test language participates in the train languages, for different test languages.* indicates the improvement is statistically significant with $p < 0.05$ based on one-tailed t-test.

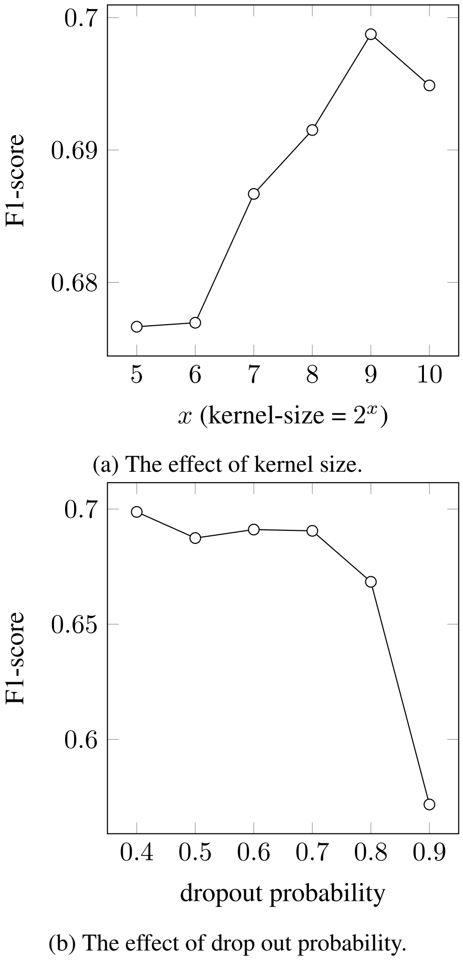
Train	Test	MNCN			Translation-Conv		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
En+Fr+Du	Sp	64.00	59.49	**61.66**	61.65	61.52	61.58
En+Fr+Sp	Du	61.04	48.04	53.76	59.68	58.54	**59.10***
En+Du+Sp	Fr	66.33	50.50	57.34	61.28	57.23	**58.18**
Du+Sp+Fr	En	69.72	65.72	67.66	68.19	71.02	**69.57***

Table 5: F1-score, precision, and recall of MNCN compared to the Translation-Conv baseline in different languages. * indicates the improvement is statistically significant with $p < 0.05$ based on one-tailed t-test.

translation phase for which the performance of the method is dependant on the quality of the translation.

We also present the per-language results of MNCN, SVM-UBT, Conv-Char-s, and Conv-Emb in Table 3. These results show the monolingual performance of MNCN in each of the languages compared to the other baselines, and also shows the variation of results per-language.

To answer the second question, we conduct an experiment to find out if MNCN is able to classify a set of reviews in a specific language using the training data from the other languages. In this experiment, for each language, we assume that there is no train data available for this language. Therefore, to perform aspect category detection in a specific language, we remove the data of that language from the train data and evaluate the model on the test samples of that language. We consider the case that the train data of the given test language is a participant in the train data of the model as the desired result for this experiment which is presented in Table 4. We compare the result of MNCN against Translation-Conv baseline for this experiment. Note that, for the Translation-Conv baseline, for each language, we translate all the train data of the other languages to that language. For example, to conduct the experiment for the Spanish (Sp) language, the training data of English (En), French (Fr), and Dutch (Du) is translated into Spanish, and then, the monolingual word embeddings trained on an Spanish corpus is used as the word representations. Table 5 shows the experimental results of MNCN compared to Translation-Conv baseline. The results show that Translation-Conv is able to achieve better results in terms of F1 in Dutch, French, and English. MCNC achieves a higher F1 in Spanish, but the difference is not significant. Although MCNC cannot outperform Translation-Conv in this experiment, it is able to achieve a reasonably good performance. We can also observe that MNCN gains better precision in all of the languages. It is worth noting that, MNCN does not need any translation process. By comparing the obtained results against the desired results in Table 4, we observe that MNCN performs reasonably in aspect category detection in situations that there is no train data available for a specific language. Another interesting point is that, in the desired results, MNCN performs better than Translation-Conv approximately in all the languages.



(a) The effect of kernel size.

(b) The effect of drop out probability.

Figure 1: Sensitivity of MNCN to kernel-size and drop-out probability parameters.

Parameter Sensitivity

In order to assess the sensitivity of our model to the parameters, we conducted experiments on the size of the kernels of the convolutional layer and the dropout probability. Figure 1 demonstrates the result of these experiments.

As can be seen in figure 1(a), the best result is achieved for a kernel-size of 512 (2^9). A model with a kernel-size smaller than 512 has troubles in learning and suffers from under-fitting, while evidently, a model with a kernel-size greater than 512 has troubles in learning and suffers from over-fitting. On the other hand, figure 1(b) the best probability for applying dropout appears to be 0.4. While slightly larger values for dropout don't appear to have much effect on the performance of our model, the performance starts to suffer for probabilities larger than 0.7.

Conclusion

In this paper, we proposed MNCN, a multilingual Ngram-based convolutional network for addressing aspect category detection in several languages at the same time. To the best of our knowledge, our work is the first attempt to address multilingual aspect category detection. MNCN does not rely on machine translation and performs multilingual aspect category detection utilizing multilingual word embeddings. We conducted extensive experiments and compared it against several baselines to show the effectiveness of the method. Empirical results on multilingual datasets demonstrate that MNCN has good performance in performing aspect category detection in multilingual fashion. For future works, we intend to test other network structures and also we will try to adapt multilingual embeddings for different domains.

References

Apidianaki, M.; Tannier, X.; and Richart, C. 2016. Datasets for aspect-based sentiment analysis in french. In *LREC*.

Artetxe, M.; Labaka, G.; and Agirre, E. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2289–2294.

Balahur, A., and Turchi, M. 2012. Multilingual sentiment analysis using machine translation? In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, 52–60. Association for Computational Linguistics.

Banea, C.; Mihalcea, R.; Wiebe, J.; and Hassan, S. 2008. Multilingual subjectivity analysis using machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 127–135. Association for Computational Linguistics.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Brooke, J.; Tofiloski, M.; and Taboada, M. 2009. Cross-linguistic sentiment analysis: From english to spanish. In *Proceedings of the international conference RANLP-2009*, 50–54.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Conneau, A.; Lample, G.; Ranzato, M.; Denoyer, L.; and Jégou, H. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

De Clercq, O., and Hoste, V. 2016. Rude waiter but mouthwatering pastries! an exploratory study into dutch aspect-based sentiment analysis. In *Tenth International Conference on Language Resources and Evaluation (LREC2016)*, 2910–2917. ELRA.

Duh, K.; Fujino, A.; and Nagata, M. 2011. Is machine translation ripe for cross-lingual sentiment classification? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, 429–433. Association for Computational Linguistics.

Hu, M., and Liu, B. 2004. Mining opinion features in customer reviews. In *AAAI*, volume 4, 755–760.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kiritchenko, S.; Zhu, X.; Cherry, C.; and Mohammad, S. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 437–442.

Mihalcea, R.; Banea, C.; and Wiebe, J. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, 976–983.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Mikolov, T.; Le, Q. V.; and Sutskever, I. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Oliphant, T. E. 2006. *A guide to NumPy*, volume 1. Trelgol Publishing USA.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. 27–35.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Mohammad, A.-S.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, 19–30.

Popescu, A.-M., and Etzioni, O. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*. Springer. 9–28.

Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics* 37(1):9–27.

Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA. http://is.muni.cz/publication/884893/en.

Ruder, S.; Ghaffari, P.; and Breslin, J. G. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745*.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 761–769.

Smith, S. L.; Turban, D. H.; Hamblin, S.; and Hammerla, N. Y. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*.

Toh, Z., and Su, J. 2016. Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, 282–288.

Wehrmann, J.; Becker, W.; Cagnini, H. E.; and Barros, R. C. 2017. A character-based convolutional neural network for language-agnostic twitter sentiment analysis. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, 2384–2391. IEEE.

Zhang, X., and LeCun, Y. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.