Borrow a Little from your Rich Cousin: Using Embeddings and Polarities of English Words for Multilingual Sentiment Classification

Prerana Singhal and **Pushpak Bhattacharyya**
Dept. of Computer Science and Engineering
IIT Bombay, Maharashtra, India
{singhal.prerana,pushpakbh}@gmail.com

Abstract

In this paper, we provide a solution to multilingual sentiment classification using deep learning. Given input text in a language, we use word translation into English and then the embeddings of these English words to train a classifier. This projection into the English space plus word embeddings gives a simple and uniform framework for multilingual sentiment analysis. A novel idea is augmentation of the training data with polar words, appearing in these sentences, along with their polarities. This approach leads to a performance gain of 7-10% over traditional classifiers on many languages, irrespective of text genre, despite the scarcity of resources in most languages.

1 Introduction

Sentiment Analysis deals with extraction of opinion polarity from texts. Extensive work has been done in this field over the past years, mainly for English. Subsequently, rich English resources like SentiWordNet (Esuli and Sebastiani, 2006) and pre-trained word embeddings (Mikolov et al., 2013a) are available publicly. However, lack of rich resources and annotated corpora in other languages like Dutch, Russian or Hindi makes it difficult to analyze texts with as good accuracy as that in English.

Deep learning models have achieved astonishing results in several fields like Speech Recognition and Computer vision, and have shown promising results when used for several NLP tasks (like Convolutional Neural Networks for Sentence Classification (Kim, 2014), LSTMs for tweet classification (Wang et al., 2015) and Recursive Deep Models for Sentiment Analysis (Socher et al., 2013)). A key feature of deep learning models, which seems to attract NLP researchers, is their lack of demand for manual feature engineering, unlike other classical machine learning algorithms (SVM, *etc*.) (Pang et al., 2002).

Multilingual Sentiment Analysis has been a challenging, yet an important area of research since a long time, mainly involving other NLP tools like Sentence-level Machine translation (Joshi et al., 2010; Wan, 2009), Machine Translation with SentiWordNet scores (Denecke, 2008), semantic orientation calculator (Brooke et al., 2009), and Wordnets (Balamurali et al., 2012) to serve the purpose. Machine learning methods have been used and evaluated on different sets of languages (Boiy and Moens, 2009; Seki et al., 2010; Seki et al., 2008), which involve many constraints and manual functionalities. In addition, strategies to build a multilingual corpus (Schulz et al., 2010) have been devised which, quite frankly, are not possible for large number of languages, owing to the diversity involved.

In this paper, we present a simple approach to multilingual sentiment classification which: (a) poses minimal restrictions on the language or the text genre to be used, (b) has minimal demands for pre-processing tools, and (c) shows no aversion to the small size of datasets or inadequacy of resources in any language. **Our approach uses deep learning models for sentiment classification in a given language by *borrowing* word-embeddings and word polarities from the *rich cousin* English.**

The main idea is to combine existing lexical resources and neural network classification techniques and provide an effective solution to the problem of multilingual sentiment classification. Although the

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3053–3062, Osaka, Japan, December 11-17 2016.

individual components (polar words (Pang et al., 2002) and word embeddings (Wang et al., 2015)) have been used by others before, **the novelty of this paper lies in the method of integrating *known components* and *available resources*, for *languages with insufficient resources*.**

2 Motivation

Multilingual Sentiment Analysis faces several challenges including (a) linguistic variations and (b) lack of sufficient resources for supervised classification. We need a simple classification system which can perform efficiently on a dataset, independent of the language or the text genre. Quite often, the size of the annotated corpus available for a particular language and text genre is generally not sufficient enough, in fact, far too small, to be helpful on its own. Hence, we propose to **use deep learning models and leverage publicly available word embeddings and polar words of *English* for sentiment classification** to overcome the limitations in a multilingual framework.

2.1 Why Deep Learning?

Classical machine learning algorithms like SVM require (a) *pre-processing* and (b) *manual feature engineering*, (Pang et al., 2002) both of which vary across languages as well as text genres. Hence, building a classification system for a new language is cumbersome and may not be as effective for all languages. We need a model, which can train effectively on any dataset, irrespective of its language or text characteristics, with minimal or no manual adjustments across datasets. Hence, deep learning models like Convolutional neural networks (Collobert et al., 2011) assume importance, since they are well-known to have no such text dependent constraints.

2.2 Why English word-embeddings?

Deep learning NLP models require word representations (containing context information) as input. One way to do so is to randomly initialize the word vectors and trust the sentiment classification model itself to learn the word representations, besides the network parameters. However, this requires a large annotated corpus, which is difficult to obtain in most languages. The other way is to train a suitable deep learning model (Collobert et al., 2011; Mikolov et al., 2013a) on a raw corpus in that language and then use the obtained embeddings of these *in-language words* as input to the sentiment classification model. However, learning context-rich word embeddings in any language requires large datasets, generally of the order of billions of words, thereby eating up a lot of time as well as system resources.

Hence, the pre-trained word embeddings of *English* prove to be useful, which have already been obtained by training suitable models on billions of data. Their context-rich information can be utilized to compensate for the small size of the available corpora in a language. An interesting property of context-rich word embeddings is that they capture linguistic regularities (Mikolov et al., 2013b), including contextual similarities. Hence, similar words or synonyms will have closer word vectors (measured by cosine distance). Thus, for the purpose of sentiment polarity classification (with no intensity segregation), all synonyms will contribute to the same polarity in a similar manner. So, one can pick any one of the synonyms (say, one of *amazing, splendid, spectacular, etc.*) and it will not affect the underlying sentiment (say positive or negative or neutral) of the text.

We propose to *obtain a mapping between the in-language words and the English word-embeddings through word-to-word translation*. For this, any publicly available (decent) bilingual dictionary or translation tool (like Google translate) can be used. The in-language words are translated to English individually, *solely for the purpose of obtaining a mapping to the pre-trained English word vectors*; hence, the orientation of the input texts is not disturbed.

2.3 Why English polar words?

A small change in the choice of words, or the order in which the words occur in a piece of text, may change the whole opinion underlying the text. For instance, the sentence "*This is not good*" conveys negative sense, as opposed to the positive sense conveyed by "*This is good*", with a single word *not* reversing the polarity. Also, the sentiment polarity of the sentence "*He does not seem bad, he is good*" is clearly opposite to that of "*He does not seem good, he is bad*", in spite of exactly the same words being used in both cases. So the classification system needs to *see all these patterns* for making correct predictions on *unseen* data.

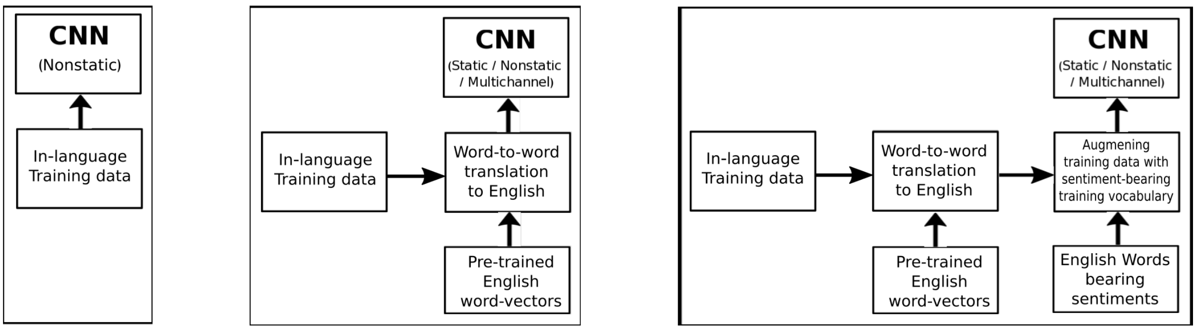
The point is, if enough instances are not provided to the classification system for training, it is likely to learn wrong patterns. For example, if the system has seen "*This may be nice but I do not like it.*" as the only training example with the word *nice*, it tends to associate the word *nice* with negative sentiment and is likely to wrongly predict an unseen sentence, say "*She seems nice to me.*", as negative. Generally, a large annotated corpus is able to overcome this anomaly, as the network gets enough data to learn the correct linguistic patterns. However, when the labeled dataset is small in size, as is often the case in most languages, this problem adversely affects the classification performance.

To overcome this hurdle, one way is to make use of some list of frequently used sentiment-polar words. The idea is to familiarize the network with the fact that the text "*This may be nice but I do not like it*" reflects negative opinion but "*nice*" portrays positive sentiment. This improves the chances of correct prediction, as now the network is able to learn that the word *nice* is generally used for positive opinion but when used in some particular context (say, with discourse particle *but* or negation element *not*), it displays negative sentiment.

Several rich English resources like SentiWordNet (Esuli and Sebastiani, 2006) or list of positive/negative English vocabulary (Hu and Liu, 2004) are publicly available. We propose to make use of this information for supervised sentiment classification in a multilingual scenario, *by augmenting sentiment bearing words with their polarities to the annotated training corpora*.

3 Proposed Method

We use a deep learning model like Convolutional neural network as our classification system. We use randomly initialized word-vectors and no other resources in our baseline model. For example, a simple CNN-Rand (Non-static) model (Kim, 2014) can be used to work as our baseline. The proposed method for sentiment classification, as depicted by the block diagram (Figure 1), can be divided into two stages (Figures 1b and 1c) to be applied on top of this base-line (Figure 1a).



(a) Baseline (b) 1st Stage (c) 2nd Stage

Figure 1: Block Diagrams of the Deep Learning Classification Methods

3.1 First Stage: *Mapping in-language texts to English Word Embeddings*

Given text in one language, we translate each word into English using bilingual dictionary or translation system like Google translate. Then, we initialize the word vectors using the corresponding English word embeddings, obtained from the existing list of pre-trained English word vectors. In case the in-language word fails to get translated, it is *transliterated* into English and its word-vector is randomly initialized during the training phase.

To illustrate our approach, let us consider dummy training data in Hindi:

1. Original Hindi text :: हम इस धारणा के प्रभाव को महसूस करते हैं, लेकिन उत्तेजित और प्रेरित नहीं होते।
   Transliteration :: *Hum is dharnaa ke prabhaav ko mehsoos karte hain, lekin uttejit aur prerit nahi hote.*
   Meaning in English :: ***We feel the effects of this notion, but do not get excited and motivated.***
   Label :: Negative
2. Original Hindi text :: फिल्म का संगीत कमजोर है।
   Transliteration :: *Film ka sangeet kamzor hai.*
   Meaning in English :: ***The film's music is weak.***
   Label :: Negative
3. Original Hindi text :: स्क्रिप्ट की बंदिशों के बावजूद अभिनेताओं ने लाजवाब काम किया है।
   Transliteration :: *Script ki bandishon ke baavajud abhinetaon ne laajawaab kaam kiya hai.*
   Meaning in English :: ***Despite the restrictions of the script, the actors did an amazing job.***
   Label :: Positive

Now, we translate (or transliterate) the training data on **word-level** to English and **map them to the corresponding English word-embeddings**. This results in the following as our training data:
*(<w> implies embedding of the English word **w** to which the in-language word has been translated)*

1. **Instance ::** ***<we> <this> <assumption> <of> <effect> <to> <are> <,> <but> <excited> <and> <inspired> <no> <there> <.>*** (Label :: Negative)
2. **Instance ::** ***<film> <of> <music> <weak> <is> <.>*** (Label :: Negative)
3. **Instance ::** ***<script> <of> <restrictions> <of> <despite> <actors> <has> <excellent> <work> <the> <is> <.>*** (Label :: Positive)

Once the word-vectors have been initialized, the network can be trained on the training data (Figure 1b), which now contains concatenated word vectors as training instances with their corresponding sentiment-labels. The word-to-word translation is done mainly *to obtain a mapping* between the in-language words and the English word embeddings. *The word order in original text sequence is not disturbed* and hence, there is neither any chaos nor any problems in handling phenomenon like negation. As compared to the baseline, this technique is expected to work better, because, unlike in the former case, it now has access to the extensive context information offered by the pre-trained English word embeddings, which can be exploited to improve the training procedure on *scarce* datasets.

3.2 Second Stage: *Augmenting training data with English Polar Words*

After obtaining the word vector mappings, we choose some authentic English resource like SentiWordNet or list of common positive-negative words (Hu and Liu, 2004), to form a list of frequently used polar sentiment-bearing lexicons. Then, once the in-language text is mapped to English on word level, the training data vocabulary is matched against this list and the intersection of the two is appended to the training data with their polarities as their labels. The idea is *to augment training data with polar words that have occurred in the training texts.*

In our example, the training data will now consist of following instances:
*(<w> implies embedding of the English word **w** to which the in-language word has been translated)*

1. **Instance ::** ***<we> <this> <assumption> <of> <effect> <to> <are> <,> <but> <excited> <and> <inspired> <no> <there> <.>*** (Negative)
2. **Instance ::** ***<film> <of> <music> <weak> <is> <.>*** (Negative)
3. **Instance ::** ***<script> <of> <restrictions> <of> <despite> <actors> <has> <excellent> <work> <the> <is> <.>*** (Positive)
4. **Instance ::** ***<excited>*** (Positive)
5. **Instance ::** ***<inspired>*** (Positive)

6. **Instance :: *<weak>*** (Negative)
7. **Instance :: *<restrictions>*** (Negative)
8. **Instance :: *<excellent>*** (Positive)

Now the word vectors are initialized as before and the network is trained as usual on *this extended training data* (Figure 1c). This technique is expected to further enhance the performance of the classification system, as it tries to fill gaps in the sentiment-related information of the small training datasets.

4 Datasets and Experimental Setup

We perform experiments on datasets in two Indian languages[1], and Russian and six European languages[2]. Neutral labels (if any) have been removed from these datasets for the purpose of *binary sentiment* (positive-negative) classification.

The datasets are essentially movie reviews in Hindi (Joshi et al., 2010), tourism reviews in Hindi and Marathi (Balamurali et al., 2012), and tweets from different contexts in Dutch, French, Spanish, Italian, German, Portuguese and Russian languages (Araujo et al., 2016). These datasets reflect the **diversity in terms of language family** (Indian languages are not as close to English as European languages), **text characteristics and length of the texts** (reviews are long and grammatically sane while tweets are short irregular piece of texts) as well as the **relatively small size of the labeled corpora**. Relevant information about the datasets have been summarized in Table 1.

Dataset Language	Average Text Length	Vocabulary Size	Dataset Size #pos	#neg	Total
Reviews in Indian Languages					
Hindi (Movie)	27	1600	127	125	252
Hindi (Tourism)	128	3601	98	100	198
Marathi (Tourism)	89	3766	75	75	150
Tweets in Russian and European Languages					
Russian	15	8021	1145	1188	2333
Dutch	21	1258	88	63	151
French	17	1800	159	160	319
German	13	1254	143	95	238
Portuguese	16	2472	297	213	510
Spanish	21	5092	683	350	1033
Italian	19	8429	820	1422	2242

Table 1: Summary Statistics for the Binary Labeled Datasets in different languages

4.1 Neural Network Architecture

For the purpose of our experiments on all datasets, we use convolutional neural network (CNN), which consists of a convolutional layer with filter windows of sizes 3, 4 and 5, a feature map of size 50 for each of the filters and sigmoid as the activation function, followed by max-pooling and an output layer with softmax as activation function. For the network, we choose negative log likelihood as the error function, learning rate to be 0.95, dropout rate to be 0.4 and number of training epochs to be 30. We train the model through stochastic gradient descent with the Adadelta update rule (Zeiler, 2012).

The hyper-parameters of the CNN model have been chosen via a grid search on the *hindi-movie-review* dataset. Three different variants of this CNN model have been tried on the datasets, as shown by Yoon Kim (2014). which are:

- **CNN Static (C-S) ::** Here, pre-trained word-vectors are used and these undergo no change during training, *i.e.*, only the parameters of the network are learned through back-propagation and not the word embeddings.

[1]available at `http://www.cfilt.iitb.ac.in/Sentiment_Analysis_Resources.html`
[2]available at `http://homepages.dcc.ufmg.br/~fabricio/sentiment-languages-dataset/`

- **CNN Non-Static (C-N) ::** Here, either pre-trained word-vectors are used and/or word vectors are randomly initialized, and during training, apart from the parameters of the network, the word-vectors are also tuned through back-propagation in order to learn word-embeddings for the specific task of sentiment classification.
- **CNN Multi-Channel (C-M) ::** Here, two sets of pre-trained word-vectors are used with the convolutional filters applied separately on each, and during training, apart from the parameters of the network, one set of the word-vectors are tuned through back-propagation (non-static channel) while the other set undergoes no change (static channel).

4.2 Resources and Tools used

For our experiments, we use Google translate[3] for word-to-word translation from a given language to English. The English word embeddings we use are pre-trained vectors trained on a part of Google News dataset (about 100 billion words) [4]. These are 300-dimensional vectors for approximately 3 million words and phrases. We use the list of English words expressing sentiment polarities[5] compiled by Bing Liu and Minqing Hu (2004), which consists of approximately 6800 positive and negative words. We use these resources for our experiments solely because of their richness and easy availability; however, other alternatives can also be used.

	Traditional Classifier	**Deep Learning Models (Convolutional Neural Networks)**						
		Baseline	Our Approach (Stage I)			Our Approach (Stages I & II)		
Dataset Language	**SVM (unigrams)**	**C-N Lang**	**C-S E-wv**	**C-N E-wv**	**C-M E-wv**	**C-S E-wv-pw**	**C-N E-wv-pw**	**C-M E-wv-pw**
Reviews in Indian Languages								
Hindi (Movie)	71.6	73.4	71.5	76.6	74.7	76.2	**80.2**	78.4
Hindi (Tourism)	80.3	80.3	88.1	84.4	85.1	**88.9**	87.1	87.3
Marathi (Tourism)	95.7	93.3	92.4	93.4	88.8	95.8	95.7	**96.1**
Tweets in Russian and European Languages								
Russian	59.7	63.0	69.5	73.2	74.2	71.4	74.2	**74.9**
Dutch	67.0	63.6	75.9	72.7	68.8	77.5	77.0	**78.1**
French	69.7	71.3	76.1	75.2	77.2	79.5	80.4	**81.8**
German	63.3	67.7	77.2	78.0	77.9	**81.1**	80.9	79.5
Portuguese	66.4	67.7	78.3	73.9	75.3	**79.9**	77.9	79.2
Spanish	72.2	75.6	82.9	83.2	83.0	84.8	83.8	**85.2**
Italian	63.7	64.5	74.1	73.1	74.3	**75.2**	74.9	75.1

Table 2: Classification Performance results (Average F-scores) of the models in different languages

4.3 Experimental Configurations

We perform two-fold validation of five repeats on the datasets, where the configurations of training and test documents are randomly chosen for each repeat. The data is not subjected to any tuning prior to training/testing, apart from changing all (English) words to lowercase and inserting space between letters and punctuations. This configuration is maintained across all datasets and model variants, in order to maintain uniformity while comparing results.

To compare the performance of the deep learning models, we also apply SVM (Pang et al., 2002) classifiers (with unigram as features) on the datasets. We perform experiments for the following models:

- Classical Machine Learning (Baseline) Classifier Models:
  1. **SVM (unigram)::** SVM (with unigram as features) on the original in-language training data.

[3]`https://translate.google.co.in/`
[4]`https://code.google.com/archive/p/word2vec/`
[5]`http://www.cs.uic.edu/∼liub/FBS/opinion-lexicon-English.rar`

- In-language Deep Learning (CNN) Models:
  1. **C-N Lang::** CNN model in non-static mode with the *original in-language training data*, and randomly initialized word-vectors.
- Proposed Deep Learning (CNN) Models (only Stage-I Approach :: *mapping to English Word Vectors (E-wv)*):
  1. **C-S E-wv::** CNN model in static mode with the training data mapped to pre-trained English word embeddings through word-level translation.
  2. **C-N E-wv::** CNN model in non-static mode with the training data mapped to pre-trained English word embeddings through word-level translation.
  3. **C-M E-wv::** CNN model in multi-channel mode with the training data mapped to pre-trained English word embeddings through word-level translation.
- Proposed Deep Learning (CNN) Models (Stage-I and Stage-II Approach :: *mapping to English Word Vectors and augmenting training data with Polar Words (E-wv-pw)*):
  1. **C-S E-wv-pw::** CNN model in static mode with the training data mapped to pre-trained English word embeddings through word-level translation and augmented with English polar words.
  2. **C-N E-wv-pw::** CNN model in non-static mode with the training data mapped to pre-trained English word embeddings through word-level translation and augmented with English polar words.
  3. **C-M E-wv-pw::** CNN model in multi-channel mode with the training data mapped to pre-trained English word embeddings through word-level translation and augmented with English polar words.

5 Results

In Table 2, we report F-score values (mean of positive and negative class F-scores) of all our models on the different datasets.

Although an increase in the performance of the deep learning models, aided by the English word-embeddings and word-polarities, was expected, **the magnitude of the gains encountered is overwhelming, in spite of the incredibly small size of the annotated corpora in many cases.** In fact, the improvement in performance is almost consistent across all language datasets in spite of the varied nature of the texts (movie reviews, tourism reviews and tweets in different languages).

The best results, **more than 80%** in many cases, are reported in the last three columns of Table 2, which are the variants of the CNN models, utilizing the two stages of our proposed approach. These are not only better than the CNN baseline (C-N Lang) but also **show astonishing performance gains over the SVM model across all datasets, as high as 10% in some cases**.

6 Observations and Discussions

It is evident that across all languages and datasets, our proposed approach of using deep learning models (one of the variants of CNN in our case), leveraging the pre-trained word-embeddings and the polar words of English, consistently performs better than the classical model (SVM) as well as the CNN baseline.

The most significant property exhibited by our proposed system is effective handling of (a) unknown words and (b) scarcity of datasets; two challenges frequently faced by the task of sentiment classification in most languages.

6.1 Unknown Words

Unknown words pose a great challenge to almost all NLP tasks including Sentiment Analysis. They are infamous for bringing the performance of the NLP systems considerably down, as these can play no role to predict the label of unseen data. Table 3 shows the average number of unknown words encountered during the testing phase with and without being mapped to English word embeddings. Clearly, *when relying solely on the in-language training data, the number of unknown words encountered is more than 50% of the test vocabulary size* in most cases. This is bound to decrease the performance of the classification system. However, *leveraging the context information of the pre-trained English word embedding space helps to 'know' a large fraction of the unknown words*. Consequently, the number of unknown words reduce considerably after applying our proposed method and Table 3 clearly backs up this argument. Furthermore, this 'knowledge' is useful for the task at hand, as clearly reflected by the performance gains by our system (Table 2).

Datasets	**Average Vocabulary Size in test data**	Average number of Unknown Words encountered in test data	
		Without using English words	**Using English words**
Reviews in Indian Languages			
Hindi (Movie)	1162	658	95
Hindi (Tourism)	3030	1710	284
Marathi (Tourism)	3240	2283	506
Tweets in Russian and European Languages			
Russian	6757	5149	1262
Dutch	807	589	115
French	1203	877	164
German	855	612	104
Portuguese	1706	1197	300
Spanish	3654	2510	826
Italian	6289	4416	1442

Table 3: Statistics of Unknown words with and without being mapped to English word embeddings

6.2 Scarcity of data

One basic requirement of a supervised classification model is a substantial amount of data for training the network, which often becomes a major challenge for multilingual sentiment analysis. In-language classification results in far more amount of unseen instances and/or words than that which can be handled, because of which the baseline models do not perform very well on the scarce datasets. This shortcoming is somewhat overcome by our proposed method of *projecting the in-language text to the English word space and augmenting the training data with polar words*. Table 2 substantiates our claim: we see more than 80% accuracy for most of the datasets, in spite of their small size. These numbers are statistically significant too, despite the small size of the datasets, not only because the results show consistent improvements over several datasets which is *not a coincidence*, but also, large annotated corpora are *practically* not available for most languages.

7 Conclusion

In this work, we established that, even with a simple deep learning classification model, and easy usage of publicly available word embeddings and polar words of English, appreciable results can be obtained for *multilingual sentiment classification*. **The main idea of our proposed approach is to map the in-language words with English word embeddings and augment the training dataset with polar words**. Although the individual components like polar words and word embeddings have been utilized before, the proposed approach, as a whole, is substantially different from any of the previous works. The *novelty* of this paper lies in the method of *borrowing known components from the 'rich' language English, and integrating them using deep learning models, for the task of multilingual sentiment classification.*

The experimental results show more than 80% performance F-score values *with as high as 10% performance gain* over the classical models in many cases. These observations substantiate the viability of our proposed approach in handling the key issues of multilingual sentiment classification, namely, diversity of texts and scarcity of datasets across languages.

We currently targeted binary sentiment classification for different languages in this paper. We show that **exploiting the embeddings and sentiment polarities of English words (without relying on any complex tools), and effectively applying deep learning models, is a viable approach to sentiment classification in a multilingual setup**. This work can further be extended to multi-class sentiment classification and possibly aspect classification in different languages. Also, further experimentations can be conducted with other publicly available resources, datasets and tools as well as other deep learning neural network configurations for multilingual sentiment analysis.

References

Matheus Araujo, Julio Reis, Adriano Pereira, and Fabricio Benevenuto. 2016. An evaluation of machine translation for multilingual sentence-level sentiment analysis. In *Proceedings of the 31st ACM symposium on Applied computing*.

AR Balamurali, Aditya Joshi, and Pushpak Bhattacharyya. 2012. Cross-lingual sentiment analysis for indian languages using linked wordnets. In *Proceedings of COLING 2012*, pages 73–82.

Erik Boiy and Marie-Francine Moens. 2009. A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558.

Julian Brooke, Milan Tofiloski, and Maite Taboada. 2009. Cross-linguistic sentiment analysis: From english to spanish. In *RANLP*, pages 50–54.

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

Kerstin Denecke. 2008. Using sentiwordnet for multilingual sentiment analysis. In *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, pages 507–512.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Maria Holmqvist, Sara Stymne, Jody Foo, and Lars Ahrenberg. 2009. Improving alignment for smt by reordering and augmenting the training corpus. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 120–124.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. 2010. A fall-back strategy for sentiment analysis in hindi: a case study. In *Proceedings of the 8th ICON*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of ACL*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on EMNLP*, page 1746–1751.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL*, pages 746–751.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on EMNLP-Volume 10*, pages 79–86.

Julia Maria Schulz, Christa Womser-Hacker, and Thomas Mandl. 2010. Multilingual corpus development for opinion mining. In *LREC*.

Yohei Seki, David Kirk Evans, Lun-Wei Ku, Le Sun, Hsin-Hsi Chen, Noriko Kando, and Chin-Yew Lin. 2008. Overview of multilingual opinion analysis task at ntcir-7. In *NTCIR*.

Yohei Seki, Lun-Wei Ku, Le Sun, Hsin-His Chen, and Noriko Kando. 2010. Overview of multilingual opinion analysis task at ntcir-8. In *Proc. of the 7th NTCIR Workshop*.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, volume 1631, pages 1631–1642.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 235–243.

Xin Wang, Yuanchao Liu, Chengjie Sun, Baoxun Wang, and Xiaolong Wang. 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the 53rd Annual Meeting of ACL and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1343–1353.

Bin Wei and Christopher Pal. 2010. Cross lingual adaptation: an experiment on sentiment classifications. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 258–262.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *CoRR*.