Information Processing and Management 54 (2018) 451–462

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Multilingual opinion mining on YouTube – A convolutional N-gram BiLSTM word embedding

Huy Tien Nguyen, Minh Le Nguyen*

Japan Advanced Institute of Science and Technology (JAIST), Japan

ARTICLE INFO

Keywords:
Sentiment analysis
Multilingual opinion mining
Convolutional
N-gram word embedding
BiLSTM

ABSTRACT

Opinion mining in a multilingual and multi-domain environment as YouTube requires models to be robust across domains as well as languages, and not to rely on linguistic resources (e.g. syntactic parsers, POS-taggers, pre-defined dictionaries) which are not always available in many languages. In this work, we i) proposed a convolutional N-gram BiLSTM (CoNBiLSTM) word embedding which represents a word with semantic and contextual information in short and long distance periods; ii) applied CoNBiLSTM word embedding for predicting the type of a comment, its polarity sentiment (positive, neutral or negative) and whether the sentiment is directed toward the product or video; iii) evaluated the efficiency of our model on the SenTube dataset, which contains comments from two domains (i.e. automobile, tablet) and two languages (i.e. English, Italian). According to the experimental results, CoNBiLSTM generally outperforms the approach using SVM with shallow syntactic structures (STRUCT) – the current state-of-the-art sentiment analysis on the SenTube dataset. In addition, our model achieves more robustness across domains than the STRUCT (e.g. 7.47% of the difference in performance between the two domains for our model vs. 18.8% for the STRUCT)

1. Introduction

The emergence of web 2.0, which allows users to generate content, is causing the rapid increase in the amount of data. Platforms (e.g. Twitter, Facebook, and YouTube), which enable millions of users to share information and comments, have a high demand for extracting knowledge from user-generated content. Useful information to be analyzed from those comments are opinions/sentiments, which express subjective opinions, evaluations, appraisals, attitudes, and emotions of particular users towards entities. These sentiment comments affect the reputation of a person, and an organization, or a specific product. Sentiment analysis, also called opinion mining, is the field of study that analyzes opinions/sentiments from text. This work is a fundamental task and has attracted a huge amount of research in recent years (Liu, 2012; Pang & Lee, 2008). The task calls for identifying the sentiment polarity (positive or negative or neutral) of a comment or review.

YouTube is a social media platform with many facets. This platform is multi-domain, multilingual and multicultural since users from different countries can upload videos as well as comments about various topics in different languages. According to Aliaksei Severyn's study (2016), 60–80% of the YouTube comments do actually contain opinions. Therefore, a robust method of sentiment analysis in such an environment is a high interest for both the industry and the research community. For this reason, our research focuses on YouTube. This environment raises some challenges to opinion mining such as i) many comments may not be in well-grammar text; ii) YouTube covers a variety of domains (e.g. phone, education) that requires a robust approach to extract opinions

* Corresponding author.
E-mail addresses: ntienhuy@jaist.ac.jp (H.T. Nguyen), nguyenml@jaist.ac.jp (M. Le Nguyen).

https://doi.org/10.1016/j.ipm.2018.02.001
Received 8 September 2017; Received in revised form 11 January 2018; Accepted 7 February 2018
Available online 16 February 2018
0306-4573/

from different topics; iii) words showing sentiment can refer to either the content itself of video or the advertised product; iv) some comments are unrelated to topics or spams; v) YouTube's content has a large variety of languages; thus it requires a method to be independent to grammar of natural languages.

To address these challenges, we proposed the Convolutional N-gram BiLSTM word embedding model for sentiment analysis by capturing semantic and contextual information. The advantage of this approach does not require any linguistic resources or handcraft features, but achieves a robust performance in multi-lingual environment.

The remainder of this paper is organized as follows: Section 2 outlines the motivation and contribution of the work, Section 3 reviews the previous research on opinion mining, Section 4 introduces the architecture of our model, Section 5 describes the SenTube dataset and the tasks, Section 6 reports and discusses the results of the experiments, and Section 7 concludes our work.

2. Motivation and contribution of the work

Most prior research on sentiment analysis relied on Bag of Word (BOW) representation. For instance, Wang and Manning (2012) used a Support Vector Machine variant with Naive Bayes feature (NBSVM). Presenting a document or a sentence with Bag of bi-gram features, NBSVM consistently performs well across datasets of long and short reviews. The winning system (Mohammad, Kiritchenko, & Zhu, 2013) of the SemEval 2013 shared task used a BOW representation together with a sentiment lexicon in Support Vector Machine. However, BOW representation loses the ordering of words, and it also ignores semantics of words. The following example illustrates the problem of BOW representation:

*iPad 2 is <u>better</u>. the <u>superior</u> apps just <u>destroy</u> the **xoom**.*

In the above comment about the product **xoom**, there are one negative word and two positive words, but the sentiment toward the product is negative. This situation is common in YouTube environment where people can mention about a video and/or the product in that video and/or another product. Under BOW representation, we cannot determine which word a polarity word give an opinion toward. To address the weakness of BOW representation, Severyn, Moschitti, Uryupina, Plank, and Filippova (2016) encoded a comment into a shallow syntactic tree with enriched tags (STRUCT). The advantage of the STRUCT is to capture sentiment words as well as essential concepts about the product and negation words. This approach requires a POS-tagger tool, a chunker tool and a set of sentiment lexicons for each language. Therefore, the applicability of this method in a multilingual environment is limited. In addition, which polarity a sentiment word is depends on the context of that word. This information could not be captured by the tree structure.

Bengio, Ducharme, Vincent, and Jauvin (2003) introduced an unsupervised framework that learns continuous vector for each word. In this vector space, semantically similar words have similar vector representations (e.g., ``*strong*" is close to ``*powerful*"), whereas BOW representation gives the same distances between two words (e.g., *distance*(``*strong*"", ``*powerful*"") = *distance*(``*strong*"", ``*weak*"")). This word embedding representation has contributed to the success of deep learning methods in natural language processing, especially sentiment analysis.

A word can have different functions/meanings under different contexts. For example, the two words *"has"* from comment #1 and #2 in Table 1 have two different functions (i.e. verb vs auxiliary verb); or in comment #3 and #4, the adjective *"cheap"* bears different sentiments (i.e. negative vs positive). In spite of different functions/meanings which a word could have, the word embedding model gives a unique vector for each word. Consequently, this representation loses the word's function as well as the word's contextual meaning.

By observing the neighbor words of a word, a human could identify the function/meaning/sentiment of that word. Understanding correctly every word in a sentence helps to capture clearly the meaning of that sentence. This inspires us to design convolutional filters to encode words and their contextual information into a convolutional N-gram word embedding representation. However, the convolutional filters have two limitations: i) long distance contextual information is missing because of the relatively small size of filters. For example, the word *"Although"* restrains the negative sentiment of the word *"outdated"* in comment #5 in Table 1. Because these two words have a long distance relationship, convolution filters miss this contextual information; ii) the position of a word in a sentence/document often describes how important that word is (e.g. placing an adjective in the first position of a review gives an emphasis on that adjective). However, the convolutional operator does not consider word's position, it applies the same filters to each word. To address the weakness of convolutional filters, Bidirectional Long Short Term Memory (BiLSTM) (Dyer, Ballesteros, Ling, Matthews, & Smith, 2015) is applied to the convolutional N-gram word embedding representation for capturing long distance contextual information and taking into account of word's position. This representation is called the convolutional N-gram BiLSTM (CoNBiLSTM) word embedding. Fig. 1 shows an overview of our proposed framework for YouTube sentiment analysis.

The contribution of our research is:

Table 1
Some YouTube comments from the SenTube dataset.

No. #	Comment
1	*as would I, jaguar always **has** a place in my heart. a place that BMW cannot fill*
2	*I agree however now Jaguar **has** been bought out the reliability should increase.*
3	*... Nobody wants it because it's made of **cheap** materials...*
4	*... i couldnt believe it when my friend told me about this site. and i can tell u, ive seen this car selling ridiculously **cheap** on this site.*
5	***Although** people say the iPads multitasking is **outdated**, i think it makes more sense – you pause one app and flick to another. I would very rarely want an app to run in the background on a tablet.*

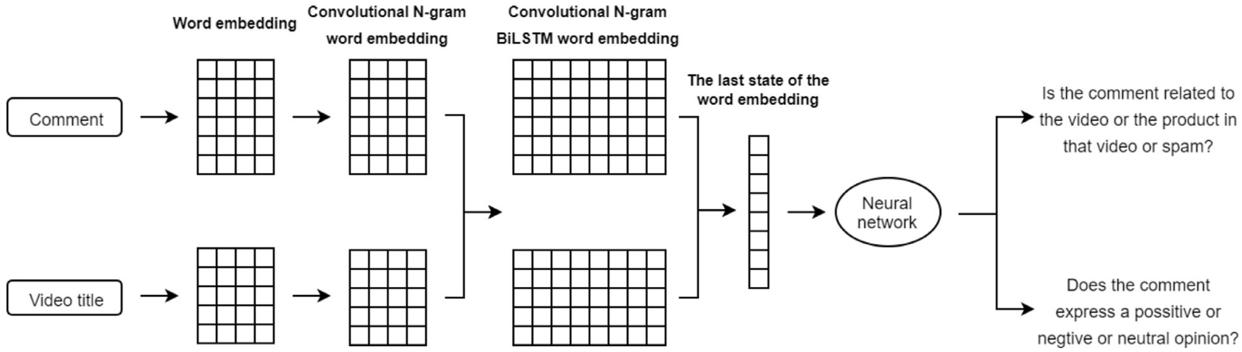

Fig. 1. An overview of our sentiment analysis model.

1. *CoNBiLSTM word embedding representation*: To enhance the conventional word embedding representation for capturing contextual information, we designed multiple convolutional filters with variant sizes. The convolution N-gram vectors generated by applying these filters on the word embedding representation are fed to a Bidirectional Long Short Term Memory (BiLSTM) for encoding long distance contextual dependencies and information of word's position.
2. *Applicable to any language*: because our approach relies on the word embedding representation learned in an unsupervised manner, our model does not require any linguistic preprocessor (e.g. POS-tagger, chunker). In some languages (e.g. Japanese, Chinese, Vietnamese) where words are not delimited by spaces, word embeddings could be trained on syllables or characters instead of words. Phung and De Vine (2015) found that word segmentation does not give any advantage in learning Vietnamese word embedding representations for text summarization. In Vietnamese social media, Vo, Nguyen, Le, and Nguyen (2017) observe that comments are usually informal and contain several grammar mistakes. These facts challenge word segmentation tools. As a result, syllable embedding models outperforms word embedding models for Vietnamese sentiment analysis. Therefore, the applicability of our approach for multilingual environments is promising.
3. *Multilingual experiments*: to validate the robustness of this approach across languages, we carried out experiments on English and Italian comments in the SenTube dataset (Uryupina, Plank, Severyn, Rotondi, & Moschitti, 2014).
4. *Cross domain experiments*: our novel word representation is learned from context data, while each domain (e.g. tablets, automobiles) has its own word distribution. Therefore, it is important to evaluate the model's robustness in a cross domain manner where the model is trained on a domain and tested on another domain. In our work, we performed experiments on two domains: automobiles and tablets in the SenTube dataset.
5. *CoNBiLSTM vs. BiLSTM*: since BiLSTM has an ability to capture long and short dependencies, there is an argument over whether or not convolutional filters have a support to BiLSTM for capturing contextual information. To confirm the efficiency of using convolutional filters for encoding contextual information, we performed experiments to compare the performances of the two models (BiLSTM and CoNBiLSTM) for classification tasks.

In most of the experiments, our model outperforms BiLSTM model and the prior work (Severyn et al., 2016), which uses the shallow syntactic tree with Support Vector Machine (STRUCT). Especially in the cross domain experiment, the proposed approach is more robust than the prior work.

3. Related work

Sentiment analysis is a study of determining people's opinions, emotions toward entities. We firstly review the work on English sentiment analysis and then focus on the work applied in multi-lingual settings.

3.1. Sentiment analysis in English

3.1.1. Feature based approach

Taboada, Brooke, Tofiloski, Voll, and Stede (2011) assigned sentiment labels to text by extracting sentiment-bearing words. To apply supervised machine learning techniques for this task, Liu (2012) formulated the sentiment analysis task as a classification problem. In this approach, dominant research concentrated on designing effective features such as word ngram (Wang & Manning, 2012), emoticon (Zhao, Dong, Wu, & Xu, 2012), sentiment words (Kiritchenko, Zhu, & Mohammad, 2014). Saif, He, Fernandez, and Alani (2016) introduced a lexicon-based approach for representing semantic sentiment information of words from their co-occurrence patterns, which can perform for both entity-level and tweet-level sentiment detection. For sentiment detection, Fersini, Messina, and Pozzi (2016) investigates the impact of several expressive signals (i.e. adjectives, pragmatic particles and expressive lengthening). These signals have been employed to enrich the feature space of baseline and ensemble classifiers. According to the experimental results, the author concluded that adjectives are more discriminative and impacting than pragmatic particles and expressive lengthening. The polarity of a single word is impacted by its context (e.g. cheap price (positive) vs. cheap material (negative)). To disambiguate contextual sentiment polarity at word-level, Vechtomova (2017) introduced an information retrieval approach which uses reference corpora with sentiment annotated documents. Although this approach was shown to be an effective alternative to machine learning approach for disambiguating word-level contextual sentiment polarity, the method has not shown an improvement compared to other methods in sentence-level sentiment analysis. Instead of using additional reference corpora for disambiguating sentiment polarity, we design the CoNBiLSTM model to learn contextual sentiment polarity for each word. The experimental results show the efficiency of this approach at sentence-level.

Metaheuristic-based methods have also been applied to opinion mining. Gupta, Reddy, Ekbal et al. (2015) proposed a Particle Swarm Optimization approach to select features and applied these features to the Conditional Random Field method for classifying sentiment. Compared to existing other systems, the method attained the promising performance with the much reduced feature set. Pandey, Rajpoot, and Saraswat (2017) employed K-means to resolve the problem of the random initialization in the cuckoo search. By optimizing the cluster-heads of sentiment dataset, the method outperformed the cuckoo search and the improved cuckoo search.

However, designing handcrafted features requires an intensive effort as well as linguistic preprocessing tools (e.g. POST-taggers, chunkers). This weakness limits the applicability on a multilingual environment like YouTube. In the next section, we review some deep learning techniques for sentiment analysis. One of deep learnings main advantages is its capacity to learn new features from a limited set of features. Therefore, it is a promising approach for a multilingual environment.

3.1.2. Deep learning approach

Recently, the emergence of deep learning models has provided an efficient way to learn continuous representation vectors for sentiment classification. Bengio et al. (2003) and Mikolov, Chen, Corrado, and Dean (2013) introduced learning techniques for semantic word representation. By using a neural network in the context of a word prediction task, the authors generated word embedding vectors carrying semantic meanings. The embedding vectors of words which share similar meanings are close to each other. However, semantic information might provide opposite opinions in different contexts. Therefore, some research (Maas et al., 2011; Socher, Pennington, Huang, Ng, & Manning, 2011; Tang et al., 2014) worked on learning sentiment-specific word representation by employing sentiment text. For sentence and document level, composition approach attracted many studies. Yessenalina and Cardie (2011) modeled each word as a matrix and used iterated matrix multiplication to present a phrase. Deep recursive neural networks (DRNN) over tree structures were employed to learn sentence representation for sentiment classification such as DRNN with binary parse trees (Irsoy & Cardie, 2014), Recursive tensor neural network with sentiment treebank (Socher et al., 2013). Convolutional neural network (CNN) has recently been applied efficiently for semantic composition (Kalchbrenner, Grefenstette, & Blunsom, 2014; Kim, 2014). This technique uses convolutional filters to capture local dependencies in term of context windows and applies a pooling layer to extract global features. Le and Mikolov (2014) applied paragraph information into the word embedding technique to learn semantic document representation. Tang, Qin, and Liu (2015) used CNN or LSTM to learn sentence representation and encoded these semantic vectors in document representation by gated recurrent neural network. Zhang, Lee, and Radev (2016) proposed Dependency Sensitive CNN to build hierarchically textual representations by processing pretrained word embeddings. Huy Tien and Minh Le (2017) propose a freezing scheme to learn sentiment features. This technique efficiently integrates the advantages of LSTM-CNN and avoids overfitting. Although contextual information might change the sentiment polarity of a word, this property is still not carefully considered in the prior work. To confirm the efficiency of the proposed CoNBiLSTM for encoding contextual sentiment polarity, we carried out experiments and quality analysis to compare the performances of CoNBiLSTM and BiLSTM.

3.2. Sentiment analysis in multi-lingual setting

Severyn et al. (2016) proposed a shallow syntactic tree with enriched tags. This structure captures not only words from the sentiment lexicons, but also important concepts about the product and negation words. For evaluation, the work has released a YouTube corpus (in Italian and English). According to the experimental results, the method improves performance for both the languages. Vilares, Alonso, and Gómez-Rodríguez (2017) evaluated the performance of classifying multilingual polarity in various settings such as a multilingual model trained on a multilingual dataset, a dual monolingual model with/without language identification. The experimental results on English and Spanish tweets showed the efficiency and robustness of the multilingual approach.

To avoid using syntactic features, Giatsoglou et al. (2017) proposed a hybrid vectorization approach for integrating emotional words along with the word embedding approach. The experiments are carried out on English and Greek languages. By combining word embedding and Bag-of-Words representations, the hybrid method outperformed existing other approaches.

As relying on linguistic resources (e.g. emotional words, sentiment lexicons, POS-tagger), these above approaches could not be applied for low-resource languages. In contrast, our work enhanced the word embedding representation by capturing contextual information without using any additional linguistic resources. Employing convolutional filters and BiLSTM, the proposed contextual word embedding model achieves a better generalization across different domains, where the word distribution and vocabulary changes, compared to the prior work (Severyn et al., 2016).

4. Convolutional N-gram BiLSTM word embedding

In this section, we introduce i) the background of BiLSTM architecture, and then ii) the proposed model – CoNBiLSTM word embedding.

4.1. Bidirectional Long Short Term Memory (BiLSTM)

LSTM was introduced by Hochreiter and Schmidhuber (1997). By designing a memory cell preserving its state over a long period of time and non-linear gating units regulating information flow into and out of the cell, Hochreiter made LSTM able to capture efficiently long distance dependencies of sequential data without suffering the exploding or vanishing gradient problem of Recurrent neural network (Goller & Kuchler, 1996).

Sentences of variable length are transformed to fix-length vectors by recursively applying an LSTM unit to each input word x_t of sentences and the previous step h_{t-1}. At each time step t, the LSTM unit with l-memory dimension defines 6 vectors in $\mathbb{R}^l$: input gate i_t, forget gate f_t, output gate o_t, tanh layer u_t, memory cell c_t and hidden state h_t as follows (from Tai, Socher, & Manning (2015)):

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$u_t = \tanh(W_u x_t + U_u h_{t-1} + b_u) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot u_t \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where σ, $\odot$ respectively denote a logistic sigmoid function and element-wise multiplication; W_i, U_i, b_i are respectively two weights matrices and a bias vector for input gate i. The denotation is similar to forget gate f, output gate o, tanh layer u, memory cell c and hidden state h. Intuitively, the forget gate makes a decision of which previous information in the memory cell should be forgotten, while the input gate controls what new information should be stored in the memory cell. Finally, the output gate decides the amount of information from the internal memory cell should be exposed. These gate units help an LSTM model remember significant information over multiple time steps. However, the LSTMs hidden state h_t only takes information of the left context, and knows nothing about the right context. To address this weakness, Dyer et al. (2015) has proposed Bi-directional LSTM (BiLSTM). For capturing the left and right context, BiLSTM applies two separate LSTM units, one for forward direction and one for backward direction. Two hidden states $h_t^{fordward}$ and $h_t^{backward}$ from these LSTM units are concatenated into a final hidden state h_t^{bilstm}:

$$h_t^{bilstm} = h_t^{forward} \oplus h_t^{backward} \tag{7}$$

where $\oplus$ is concatenation operator.

4.2. Convolutional N-gram BiLSTM word embedding

We present a document of length s as a matrix $d \times s$, where each column is a d-dimension word embedding vector of each word. Given a document matrix $\mathbf{S}$, we performs convolution on this input via linear filters H_k. Each filter H_k is denoted as a weight matrix W_{H_k} of length 1 and region size h_k and a bias value b_{H_k}. W_{H_k} will have $h_k + 1$ parameters to be estimated. Given an input matrix $S \in \mathbb{R}^{d \times s}$ and a filter H_k, the matrix S is converted into $S_k \in \mathbb{R}^{d \times h_k/2+s+h_k/2}$ by padding zero, which makes the result of convolutional operator be the same dimension as the matrix S. Then a convolutional N-gram word embedding matrix $C_k \in \mathbb{R}^{d \times s}$ is obtained as follows:

$$C_k[i, j] = W_{H_k} \cdot S_k[i, j - h_k/2 : j + h_k/2] + b_{H_k} \tag{8}$$

where $\cdot$ is dot product operation and $S_k[i, l: t]$ is the sub-matrix of S_k from column l to t of row i.

To obtain a final convolutional N-gram word embedding matrix, an average pooling is applied over those convolutional N-gram word embedding matrices C_k as follows:

$$C[i, j] = \frac{1}{N} \sum_{k}^{N} C_k[i, j] \tag{9}$$

where N is the number of filters.

In YouTube context, a comment can give sentiment to either a video or the product in that video or even other products. This challenge makes sentiment analysis in YouTube environment more difficult. For facilitating the model's ability to determine which subject a comment refers to, we use the video title as an additional feature. The reason for this choice is that a video title usually describes the product in that video. Given a comment and the title of the video which the comment belongs to, we apply the above process to achieve two convolutional N-gram word embedding matrices $C^{comment}$ and C^{title} for the title and the comment respectively.

To make our convolutional N-gram word embedding take into account the word's position as well as capture long distance contextual information, we apply BiLSTM to this word embedding:

$$T = BiLSTM(C^{comment} \oplus C^{title}) \tag{10}$$

where $\oplus$ is concatenation operator, each column in $T \in \mathbb{R}^{2l \times s}$ is a $2l$-dimension CoNBiLSTM word embedding vector of each word, which is constructed by the equation (7), and l is the LSTM unit's dimension.

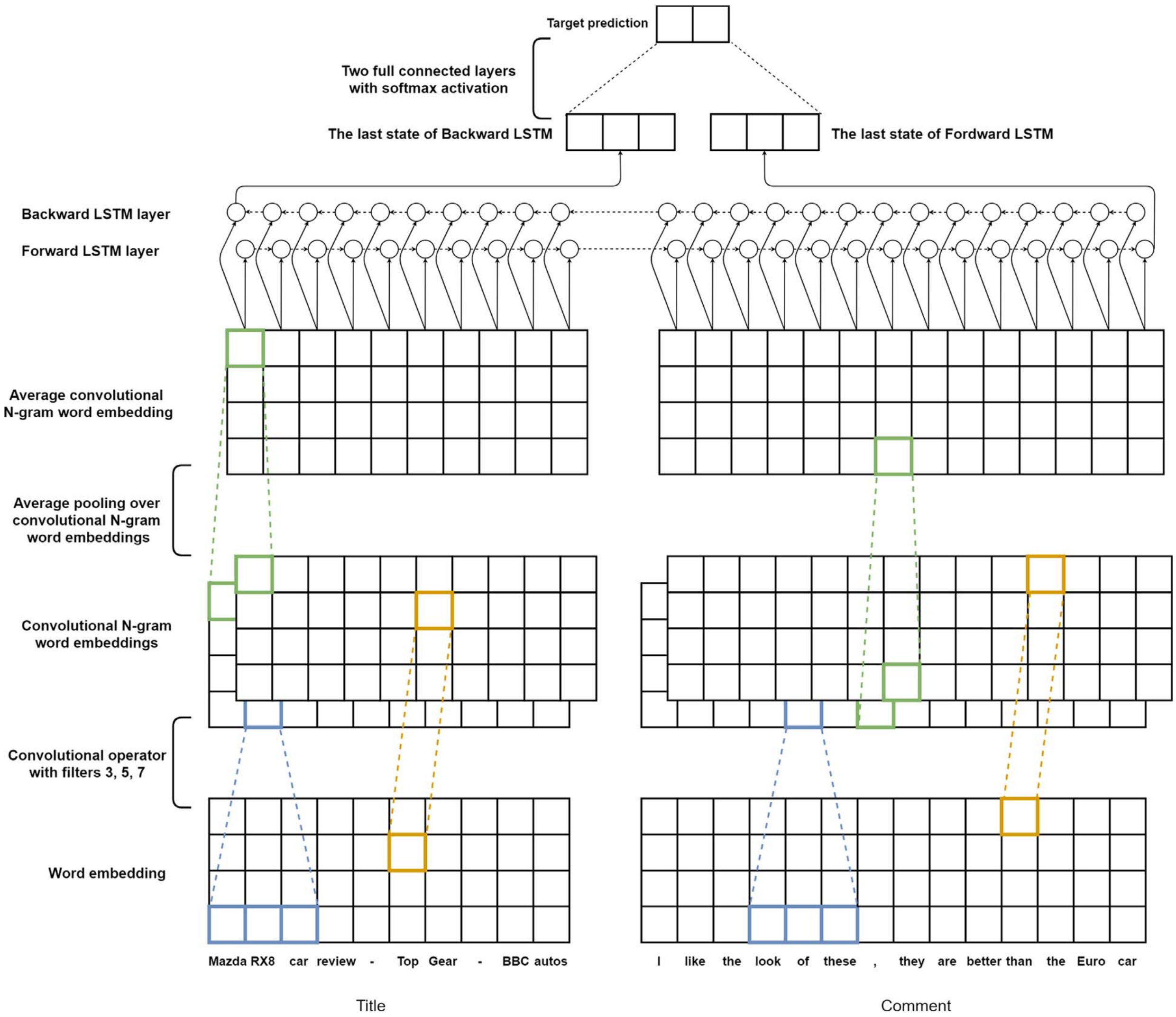

Fig. 2. Illustration of our convolutional N-gram BiLSTM word embedding model for classification.

For classification tasks with CoNBiLSTM word embedding, the first and last columns of T, which are the word embedding vectors capturing a whole context in forward and backward direction respectively, are fed to a two full-connected-layers neural network:

$$x_0 = T[1, :] \oplus T[s, :] \tag{11}$$

$$x_1 = sigmoid(x_0 W_{l_1} + b_{l_1}) \tag{12}$$

$$\hat{y} = softmax(x_1 W_{l_2} + b_{l_2}) \tag{13}$$

where x_0 is a $4l$ dimension vector constructed by concatenating the first column $T[1, :]$ and the last column $T[s, :]$; $W_{l_1} \in \mathbb{R}^{4l \times d_1}$, $b_{l_1} \in \mathbb{R}^{d_1}$, $W_{l_2} \in \mathbb{R}^{d_1 \times d_2}$, and $b_{l_2} \in \mathbb{R}^{d_2}$ are the parameters of the neural network; and $\hat{y}$ is the prediction output of our model.

In summary, the input includes a comment and the title of the video which that comment belongs to. From this input, our model builds the CoNBiLSTM word embedding and then uses this word embedding to predict a target class. In Fig. 2, we illustrate our CoNBiLSTM word embedding model, in which we use two filters with size 1, 3. In the training phase, the gradient descent optimization called ADADELTA is used to learn model parameters. Details of ADADELTA method can be found in (Zeiler, 2012)

5. SenTube dataset and task description

In this section, we introduce (i) the description of three classification tasks, (ii) the SenTube dataset as well as the procedure of preparing train/validation/test sets.

Table 2
Corpus statistics for the SenTube dataset. *In-vocabulary size* denotes the number of terms existing in the pre-trained word embedding.

	English			Italian		
	AUTO	TABLET	ALL	AUTO	TABLET	ALL
Video in total	78	139	217	98	100	198
Comment in total	16,787	22,073	38,860	4725	5539	10,264
Avg.len. of comment	98	99	99	154	111	131
Avg.len. of title	41	36	38	39	47	43
Vocabulary size	39,936	38,790	66,862	25,362	20,137	39,955
In-vocabulary size	11,536(29%)	10,491(27%)	16,013(24%)	10,802(43%)	7729(38%)	14,517(36%)

5.1. Task description

In our experiment, we evaluated the proposed model on three tasks:

- **Sentiment task**: This task detects whether a comment expresses a positive, a negative, or a neutral sentiment. The sentiment can be general or about a specific topic (e.g. product or video).
- **Type task**: One challenge in YouTube environment is that a comment expresses its sentiment toward not only the product in the video but also the video itself. Therefore, it is important to determine the target subject which the comment gives its sentiment to. Additionally, a comment could be irrelevant for both the product and the video (off-topic) or even contain spam. In this task, we classify a comment into video, product or uninformative (off-topic or spam) type.
- **Full task**: In this task, we want to jointly predict the sentiment and the type of each comment. The problem is cast into a multi-label classification with 7 labels: the Cartesian product between {product, video} type labels and {positive, neutral, negative} sentiment labels, and uninformative class.

5.2. SenTube dataset

SenTube (Uryupina et al., 2014) contains about 38,000 English comments and 10,000 Italian comments. The author compiled a list of products in two domains: automobiles (AUTO) and tablets (TABLET) (e.g. Apple iPad, Motorola xoom, Fiat 500), then collected and annotated comments from either commercials or review videos of those products. For several products, there is no corresponding Italian commercial or review video. Table 2 highlights some differences between the two languages as well as the two domains. In YouTube, the number of Italian audiences is much less than the number of English audiences, so the number of Italian comments is quite small compared to English comments. However, the average length of Italian comments is quite longer than English comments.

For each task, we prepare the dataset as follows:

- **Sentiment task**: Each comment is labeled as positive, negative or neutral sentiment. Comments with ambiguous sentiments (i.e. contain both positive and negative sentiments) and comments which are irrelevant for both the product and the video (off-topic) or contain spam are excluded.
- **Type task**: Each comment is labeled as video, product or uninformative (off-topic or spam) type.
- **Full task**: Each comment is labeled as product-positive, product-neutral, product-negative, video-positive, video-neutral, video-negative or uninformative. We excluded comments annotated as both video and product types as well as comments with ambiguous sentiments.

For both of the languages, we split the videos into 45% training set, 5% validation set and 50% test set, such that each video contains all its comments. Tables 3 and 4 show the data distribution of each class for English and Italian respectively. Since the number of comments in each video is different, the number of comments in the train and validation sets is not the same as the number of comments in the test set. Generally, the class distributions of the train set, validation set, and test set are similar. For example, in English **Sentiment** task for AUTO domain, the neutral class is the most frequent, and the negative class is the least frequent in the train set, validation set, and test set.

6. Experiment and discussion

In this section, we i) describe the model setting for English and Italian, and report the experimental results of ii) the in-domain experiment, iii) the cross-domain experiment, iv) the per class experiment, and v) give a quality analysis for our model.

6.1. Model configuration

6.1.1. English model

To tune the hyper-parameters of our models, we do a grid search on 30% of the dataset for **Full** task:

Table 3
Summary of English YouTube comments data used in **Sentiment, Type** and **Full** tasks.

Task	Class	AUTO			TABLET		
		Train	Validation	Test	Train	Validation	Test
Sentiment	Positive	1536(30%)	501(30%)	1601(30%)	2393(29%)	270(16%)	1666(27%)
	Negative	917(18%)	330(20%)	871(16%)	1441(17%)	505(29%)	1267(21%)
	Neutral	2591(52%)	843(50%)	2895(54%)	4476(54%)	963(55%)	3171(52%)
	Total	5044	1674	5367	8310	1738	6104
Type	Product	2755(43%)	952(44%)	2536(36%)	5938(57%)	1331(55%)	4411(59%)
	Video	2078(33%)	664(31%)	2525(36%)	1961(19%)	361(15%)	1435(19%)
	Uninfo	1565(25%)	538(25%)	1917(28%)	2608(25%)	723(30%)	1630(22%)
	Total	6398	2154	6978	10,507	2415	7476
Full	Product-pos	1107(11%)	284(16%)	349(10%)	1096(12%)	472(13%)	712(11%)
	Product-neg	931(9%)	212(12%)	217(6%)	1152(12%)	452(12%)	869(14%)
	Product-neu	1921(20%)	376(20%)	447(13%)	2837(30%)	1073(29%)	2400(37%)
	Video-pos	924(9%)	175(10%)	444(13%)	698(7%)	247(7%)	412(6%)
	Video-neg	459(5%)	65(4%)	104(3%)	314(3%)	101(3%)	150(9%)
	Video-neu	1892(19%)	199(11%)	939(28%)	844(8%)	300(8%)	599(9%)
	Uninfo	2618(27%)	513(28%)	897(26%)	2583(27%)	1071(28%)	1314(20%)
	Total	9852	1824	3397	9524	3716	6456

Table 4
Summary of Italian comments data used in **Sentiment, Type** and **Full** tasks.

Task	Class	AUTO			TABLET		
		Train	Validation	Test	Train	Validation	Test
Sentiment	Positive	665(35%)	115(32%)	265(24%)	521(22%)	50(22%)	364(23%)
	Negative	462(24%)	91(25%)	262(24%)	481(20%)	32(15%)	357(22%)
	Neutral	787(41%)	159(44%)	590(53%)	1379(58%)	139(63%)	881(55%)
	Total	1914	365	1117	2381	221	1602
Type	Product	1005(45%)	73(20%)	736(44%)	2045(63%)	287(56%)	836(62%)
	Video	647(29%)	181(50%)	525(31%)	498(15%)	120(24%)	249(18%)
	Uninfo	601(27%)	107(30%)	426(25%)	685(21%)	104(20%)	266(20%)
	Total	2253	361	1687	3228	511	1351
Full	Product-pos	253(11%)	50(7%)	176(14%)	218(10%)	172(16%)	154(10%)
	Product-neg	216(10%)	132(18%)	190(15%)	355(16%)	145(13%)	211(13%)
	Product-neu	283(13%)	148(20%)	272(21%)	719(33%)	451(41%)	551(35%)
	Video-pos	271(13%)	50(7%)	146(11%)	92(4%)	22(2%)	112(7%)
	Video-neg	127(6%)	51(7%))	36(3%)	41(2%)	11(1%)	62(4%)
	Video-neu	351(16%)	113(15%)	171(13%)	246(11%)	64(6%)	195(12%)
	Uninformative	661(31%)	206(15%)	294(13%)	527(24%)	243(22%)	285(18%)
	Total	2162	750	1285	2198	1108	1570

- *Word embedding layer*: we obtained the pre-trained word vectors *Word2Vec*.[1] It was trained on 100 billion words from English Google News, and its vectors have the dimension of 300. During the training process, these pre-trained word vectors are optimized.
- *Convolutional filters*: two filters (h = 1, 3) are employed to construct a convolutional N-gram word embedding.
- BiLSTM layer: the dimension for each direction the same as the word embedding layer (l = 300).
- First full-connected layer: the dimension is the same as BiLSTM layer (d_1 = 600)
- Dropout layer: For avoiding overfitting, we employ a dropout layer (p=0.5) between the first and second full-connected layers.
- Second full-connected layer: the dimension is the number of target labels. We used a softmax activation for this layer.

6.1.2. Italian model

To tune the hyper-parameters of our models, we do a grid search on 30% of the dataset for **Full** task:

- *Word embedding layer*: we obtained the pre-trained word vectors from Bojanowski, Grave, Joulin, and Mikolov (2017). It was trained on Italian Wikipedia data, and its vectors have the dimension of 300. During the training process, these pre-trained word vectors are optimized.
- *Convolutional filters*: two filters (h = 1, 17) are employed to construct a convolutional N-gram word embedding.
- The other layers are the same as the English model.

[1] https://code.google.com/p/word2vec/.

Table 5
The result of the in-domain experiment.

Language	Task	AUTO			TABLET		
		STRUCT	BiLSTM	CoNBiLSTM	STRUCT	BiLSTM	CoNBiLSTM
English	Sentiment	55.7	66.35	**66.89**[a,b]	**70.5**	68.89	70.17[b]
	Type	59.4	66.78	**68.36**[a,b]	78.6	78.57	**79.49**[a,b]
	Full	41.5	51.84	**53.81**[a,b]	60.3	60.08	**61.28**[a,b]
Italian	Sentiment	**61.6**	59	61.41[b]	64.4	64.47	**65.6**[a,b]
	Type	70.7	74.1	**74.75**[a,b]	77.3	78.46	**79.64**[a,b]
	Full	45.6	47.47	**51.05**[a,b]	52.4	53.24	**55.03**[a,b]

[a] Denotes result statistically significant at $p < 0.05$ via the pairwise t-test compared with STRUCT.
[b] Denotes result statistically significant at $p < 0.05$ via the pairwise t-test compared with BiLSTM.

6.2. In-domain experiment

To evaluate CoNBiLSTM word embedding model, we compared with BiLSTM and the previous work (Severyn et al., 2016) – STRUCT method on three tasks mentioned in Section 5.2. Table 5 shows the accuracy performance of three models on the SenTube dataset for the two languages and domains.

In most of the cases, the proposed method outperforms the others. In the previous work – STRUCT method, we observed that the performance on AUTO is much lower than on TABLET across all tasks. The author explained that i) TABLET contains more training data and (ii) the different types of audiences in AUTO and TABLET domains: well-informed users and geeks expressing better-motivated opinions about a product for the former vs. more general audiences for the latter. This makes the comments in AUTO more challenging to analyze. In our model, we achieved a smaller gap between AUTO and TABLET domains (e.g. 7.47% of the difference in performance between the two domains for our model vs. 18.8% for the STRUCT in English **Full** task). It shows the robust of our model to domains with different types of audiences. More specifically, the performance on AUTO domain is much improved across all tasks except **Sentiment** task for Italian AUTO and English TABLET.

In the case of **Sentiment** task, although STRUCT outperformed our model in English TABLET and Italian AUTO domains, the difference is not statistically significant. STRUCT used a pre-defined list of sentiment words to determine sentiment words in a comment. Intuitively, this facilitates the process of sentiment analysis, especially when training data is small. In our approach, we aim to a multilingual task where labeled resources (e.g. sentiment dictionary, synonym dictionary) and linguistic preprocessing tools (e.g. POS-tagger, syntactic parser) are limited; hence we do not use any additional labeled data as sentiment dictionaries.

6.3. Cross-domain experiment

In this experiment, we trained a model on the data from one domain and tested on the data from the other domain. This experiment examines the adaptability of our models as well as whether we need training data for a new domain. Table 6 reports the accuracy for the three tasks in the cross-domain setting.

According to the experimental results, our model shows a more robust and stable performance in the cross-domain setting. For example, in English **Full** task, the difference between the performances of our model trained on AUTO (48.03%) and on TABLET (47.6%) is 0.43%, while the difference of STRUCT is 6,4%. This proved that our model has more robustness and stronger generalization.

Generally, the proposed model provides an improvement in accuracy in both of the languages compared with BiLSTM and STRUCT, except the **Type** task with Italian AUTO as a source domain. In the Italian AUTO domain, the average length of comments

Table 6
The results of the cross-domain experiment.

Language	Source	Target	Task	STRUCT	BiLSTM	CoNBiLSTM
English	AUTO	TABLET	Sentiment	66.6	68.23	**69.01**[a,b]
			Type	64.1	66.51	**67.81**[a,b]
			Full	38.3	46.2	**48.03**[a,b]
	TABLET	AUTO	Sentiment	61.9	63.54	**64.94**[a,b]
			Type	55.6	63.15	**64.57**[a,b]
			Full	44.7	47.2	**47.6**[a,b]
Italian	AUTO	TABLET	Sentiment	61.2	60.54	**62.92**[a,b]
			Type	**63.8**	61.1	62.24[a,b]
			Full	29.7	33.82	**37.34**[a,b]
	TABLET	AUTO	Sentiment	54.3	56.27	**57.07**[a,b]
			Type	56.4	59.8	**60.8**[a,b]
			Full	31.7	38.24	**39.5**[a,b]

[a] Denotes result statistically significant at $p < 0.05$ via the pairwise t-test compared with STRUCT.
[b] Denotes result statistically significant at $p < 0.05$ via the pairwise t-test compared with BiLSTM.

Table 7
The precision, recall and F1 scores of CoNBiLSTM for each class in the English experiments.

Task	Class	AUTO				TABLET			
		Precision	Recall	F1	Support	Precision	Recall	F1	Support
Sentiment	Positive	66.78	62.4	64.51	1601	78.95	60.56	68.55	1666
	Negative	41.46	17.57	24.68	871	55.29	33.39	41.63	1267
	Neutral	69.62	84.21	76.22	2895	70.2	89.91	78.84	3171
	Acc		66.89				70.17		
Type	Product	71.43	78.59	74.84	2536	84.55	91.20	87.75	4411
	Video	69.50	65.43	67.40	2525	79.15	63.48	70.46	1435
	Uninfo	62.12	58.69	60.35	1917	64.39	61.90	63.12	1630
	Acc		68.36				79.49		
Full	Product-pos	62.95	55.01	58.72	349	54.39	39.19	45.55	712
	Product-neg	35.71	34.56	35.13	217	45.67	37.63	41.26	869
	Product-neu	43.03	54.59	48.13	447	67.1	79.71	72.86	2400
	Video-pos	68.2	46.85	55.54	444	78.45	68.93	73.39	412
	Video-neg	21.05	7.69	11.27	104	27.68	20.67	23.66	150
	Video-neu	68.71	44.2	53.79	939	52.05	46.58	49.16	599
	Uninfo	50.15	76.48	60.57	897	61.71	64.16	62.91	1314
	Acc		53.81				61.28		

(l = 154) is quite long compared with TABLET domain (l = 111) and the in-vocabulary size (in Table 2) of target domain TABLET (38%) is small compared with TALBET domain (43%). These differences give a challenge to our model.

6.4. Performance on each class

In this section, we analyze the per-class performance of the three tasks. Table 7 reports Precision, Recall and F1 scores of each class. In **Sentiment** task, we observed that the negative class contributes the largest error in both of the domains. In fact, negative comments in the dataset take the smallest proportion and probably contains complicated grammars, so it is more difficult to learn an efficient classifier for the negative class compared with the positive class and the uninformative class. We discuss the difficulty of detecting negative sentiment in Section 6.5. In **Type** task, the uninformative class is considerably more difficult than the other classes. An uninformative comment could be a spam or an off-topic comment or even a comment about unrelated products, so it is intuitively quite hard to classify it. In **Full** task, the Product-negative and Video-negative classes have the worst performance. This confirms with the result from **Sentiment** task.

6.5. Quality analysis

To obtain a better sense of the improvement and limitation of the proposed model, we manually inspect some typical cases, which are shown in Table 8. These samples are the good examples of the proposed model's advantages and disadvantages.

The first source of the improvement is to capture better long distance dependencies compared with BiLSTM. In sample #1, the comment gives two polarity sentiments: one negative sentiment toward the product (*Ferrari*) mentioned in the title and one positive sentiment toward the other (*Lamborghini*). Analyzing the main sentiment of this comment requires capturing a long distance relationship between the title and its comment. The second advantage of our model is to build a better word embedding, which captures contextual information and the part of speech. For example, word "*pretty*" in sample #2 is an adverb and almost contributes nothing to the meaning of this comment. However, BiLSTM considered this word as an adjective word bearing a positive sentiment. Consequently, BiLSTM made a wrong prediction for this sample. Another example is the word "*cheaper*". This word has two opposite

Table 8
Some typical samples for quality analysis. The labels are 0:product-positive, 1:product-neutral, 2:product-negative, 3:video-positive, 4:video-neutral, 5:video-negative and 6:uninformative.

No.	Title	Comment	BiLSTM	CoNBiLSTM	True
1	ferrari 430 review...	ferrari look so dull (spelling) and boring! lamborghini is so much more awsome! they look so mean! and just evil!	0	2	2
2	bugatti veyron vitesse video review	is it just me? or does the dash look pretty dull...	0	2	2
3	ferrari 430 review...	this is my dream car, and its getting cheaper and cheaper o yah!	2	0	0
4	2012 range rover evoque coupe hd video review	so it does have a rear wiper as in the new lexus rx eh?...	1	0	2
5	ferrari 430 review...	why did they change the music... the original was way more dramatic	4	4	5
6	2012 fiat 500 test drive	review but toyotas and hondas are still the most reliable cars on the planet. my 1997 honda civic still has the original engine even though it has 400,000 miles on it.	0	0	2

sentiments depending on context. In sample #3, our model correctly assigned the positive sentiment for "*cheaper*" while BiLSTM did not.

As we mentioned in Section 6.4, our model has a quite low performance in the negative class. To analyze the difficulty of negative comments, we also inspect some typical negative comments, which are reported in Table 8. Comment #4 actually is a rhetorical question and carries a negation meaning. In sample #5 and #6, these comments do not directly mention the products in the titles. However, these comments implicitly give the negative sentiments toward those products by giving good reviews for other products. These facts make the negative class more challenging than the others in sentiment analysis.

7. Conclusion

In this work, we introduced a convolutional N-gram BiLSTM word embedding model. Our approach enhances the conventional word embedding by using i) multiple convolutional filters with variant sizes for capturing contextual information; ii) BiLSTM for encoding long distance contextual dependencies. Through the multilingual and cross domain experiments on the SenTube dataset, our model shows the more robust and better performance compared with the previous work STRUCT – the state-of-the-art method on the SenTube dataset. While the previous work requires a pre-defined sentiment dictionary and some linguistic preprocessing tools, our model only requires a pre-trained word embedding which is trained in unsupervised learning scheme. Therefore, our model has a larger applicability to multilingual environments as YouTube.

For future work, we plan to improve the ability to interpret implication, where main subjects are not mentioned explicitly as we analyzed in Section 6.5. In addition, we also plan to build a model for extracting helpful comments, which give polarity sentiments and explanation for those sentiments.

Acknowledgments

This work was supported by JSPS KAKENHI Grant number JP15K16048, JSPS KAKENHI Grant Number JP15K12094, and JST CREST Grant Number JPMJCR1513, Japan.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2018.02.001.

References

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research, 3*(February), 1137–1155.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135–146.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). *Transition-based dependency parsing with stack long short-term memory. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing*. Association for Computational Linguistics334–343. http://dx.doi.org/10.3115/v1/P15-1033.

Fersini, E., Messina, E., & Pozzi, F. A. (2016). Expressive signals in social media languages to improve polarity detection. *Information Processing & Management, 52*(1), 20–35.

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications, 69*, 214–224.

Goller, C., & Kuchler, A. (1996). *Learning task-dependent distributed representations by backpropagation through structure. Neural networks, 1996, IEEE international conference on1. Neural networks, 1996, IEEE international conference on* IEEE347–352. http://dx.doi.org/10.1109/ICNN.1996.548916.

Gupta, D. K., Reddy, K. S., Ekbal, A., et al. (2015). *Pso-asent: Feature selection using particle swarm optimization for aspect based sentiment analysis. International conference on applications of natural language to information systems*. Springer220–233.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Huy Tien, N., & Minh Le, N. (2017). *An ensemble method with sentiment features and clustering support. Proceedings of the eighth international joint conference on natural language processing (volume 1: Long papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing644–653.

Irsoy, O., & Cardie, C. (2014). Deep recursive neural networks for compositionality in language. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.). *Advances in neural information processing systems 27* (pp. 2096–2104). Curran Associates, Inc.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). *A convolutional neural network for modelling sentences. Proceedings of the 52nd annual meeting of the association for computational linguistics*. Association for Computational Linguistics655–665. http://dx.doi.org/10.3115/v1/P14-1062.

Kim, Y. (2014). *Convolutional neural networks for sentence classification. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Association for Computational Linguistics1746–1751. http://dx.doi.org/10.3115/v1/D14-1181.

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research, 50*, 723–762.

Le, Q., & Mikolov, T. (2014). *Distributed representations of sentences and documents. Proceedings of the 31st international conference on machine learning32. Proceedings of the 31st international conference on machine learning* PMLR1188–1196.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, 5*(1), 1–167.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). *Learning word vectors for sentiment analysis. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics142–150.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space. Proceedings of workshop at the international conference on learning representations*.

Mohammad, S., Kiritchenko, S., & Zhu, X. (2013). *Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. Second joint conference on lexical and computational semantics (*sem), volume 2: Proceedings of the seventh international workshop on semantic evaluation (semeval 2013)*. Association for Computational Linguistics321–327.

Pandey, A. C., Rajpoot, D. S., & Saraswat, M. (2017). Twitter sentiment analysis using hybrid cuckoo search method. *Information Processing & Management, 53*(4), 764–779.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval, 2*(1–2), 1–135.

Phung, V., & De Vine, L. (2015). *A study on the use of word embeddings and pagerank for vietnamese text summarization. Proceedings of the 20th Australasian document computing symposiumADCS '15*New York, NY, USA: ACMhttp://dx.doi.org/10.1145/2838931.2838935 7:1–7:8

Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of twitter. *Information Processing & Management, 52*(1), 5–19.

Severyn, A., Moschitti, A., Uryupina, O., Plank, B., & Filippova, K. (2016). Multi-lingual opinion mining on youtube. *Information Processing & Management, 52*(1), 46–60.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). *Semi-supervised recursive autoencoders for predicting sentiment distributions. Proceedings of the conference on empirical methods in natural language processing.* Association for Computational Linguistics151–161.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). *Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the conference on empirical methods in natural language processing (EMNLP)1631. Proceedings of the conference on empirical methods in natural language processing (EMNLP)* Citeseer1642.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*(2), 267–307.

Tai, K. S., Socher, R., & Manning, C. D. (2015). *Improved semantic representations from tree-structured long short-term memory networks. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing.* Association for Computational Linguistics1556–1566. http://dx.doi.org/10.3115/v1/P15-1150.

Tang, D., Qin, B., & Liu, T. (2015). *Document modeling with gated recurrent neural network for sentiment classification. Proceedings of the 2015 conference on empirical methods in natural language processing*. Association for Computational Linguistics1422–1432. http://dx.doi.org/10.18653/v1/D15-1167.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). *Learning sentiment-specific word embedding for twitter sentiment classification. Proceedings of the 52nd annual meeting of the association for computational linguistics*. Association for Computational Linguistics1555–1565. http://dx.doi.org/10.3115/v1/P14-1146.

Uryupina, O., Plank, B., Severyn, A., Rotondi, A., & Moschitti, A. (2014). *Sentube: A corpus for sentiment analysis on youtube social media. Proceedings of the ninth international conference on language resources and evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Vechtomova, O. (2017). Disambiguating context-dependent polarity of words: An information retrieval approach. *Information Processing & Management, 53*(5), 1062–1079.

Vilares, D., Alonso, M. A., & Gómez-Rodríguez, C. (2017). Supervised sentiment analysis in multilingual environments. *Information Processing & Management, 53*(3), 595–607.

Vo, Q. H., Nguyen, H. T., Le, B., & Nguyen, M. L. (2017). *Multi-channel lstm-cnn model for vietnamese sentiment analysis. 2017 9th international conference on knowledge and systems engineering (KSE)*24–29. http://dx.doi.org/10.1109/KSE.2017.8119429.

Wang, S., & Manning, C. D. (2012). *Baselines and bigrams: Simple, good sentiment and topic classification. Proceedings of the 50th annual meeting of the association for computational linguistics*. Association for Computational Linguistics90–94.

Yessenalina, A., & Cardie, C. (2011). *Compositional matrix-space models for sentiment analysis. Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics172–182.

Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. *CoRR, abs/1212.5701*.

Zhang, R., Lee, H., & Radev, D. R. (2016). *Dependency sensitive convolutional neural networks for modeling sentences and documents. Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics1512–1521. http://dx.doi.org/10.18653/v1/N16-1177.

Zhao, J., Dong, L., Wu, J., & Xu, K. (2012). *Moodlens: An emoticon-based sentiment analysis system for chinese tweets. Proceedings of the 18th ACM sigkdd international conference on knowledge discovery and data mining.* ACM1528–1531.