ALC Project: Using a Sequential Convolutional Neural Network Model for Opinion Target Extraction

Victor Martinez Morant

May 2016

Contents

1 Introduction

Sentiment Analysis is a really interesting topic within Natural Language Processing. Nowadays, there is a huge amount of digital information coming from reviews, social networks, blog posts, etc that has enabled the posibility of applying several opinion mining techniques. In particular, Aspect-Based Sentiment Analysis consists mainly of recognizing which are the opinionated entities or properties within a document of a certain topic (product review, tweet about a polititian, ...) and determining which is the polarity associated to them.

2 Task Description

In this project, we focus in the task of extracting the Opinion Target Expresion as it is exposed in the SemEval-2015 Task 12: slot 2. Therefore, given a document about one concrete topic, the aim is to find out the linguistic expression used in the given text to refer to the reviewed entity E. In the following example, the Opinion Target Expresion is "food".

> *The food was delicious but do not come here on an empty stomach.* {category= "FOOD#QUALITY", target= "food", from: "4", to: "8", polarity= "positive"}, {category= "FOOD#STYLE_OPTIONS", target = "food", from: "4", to: "8", polarity= "negative"}

2.1 Dataset

In this project, the dataset provided by SemEval-15 Competition is used, specifically, the restaurant domain. It consists of a set of reviews from several restaurants as specified in the following table:

Restaurant Domain	Training	Test
Reviews	254	96
Sentences	1315	685
{ E#A , OTE, Polarity}	1654	845

3 System Description

3.1 Preprocessing

The dataset is annotated using a XML format. All the reviews are splitted in sentences and every sentence has associated a set of opinions. However, we would like to process the data sequentially, for instance, mixing the end of a sentence with the beginning of another or selecting just a part of a sentence. That is the reason why we apply IOB2.

3.1.1 Data Formatting: Applying IOB-2 tagging

We adopt a new data format which basically consists of a list of reviews where each review contains a list of tuples (word,tag) indicating if the word is a Beginning, Inside or Outside Tag as exposed by the IOB-2 tagging.

[
[(The, O) (food, B) (was, O) (delicious, O)]
[(The, O) (live, B) (music, I) (was, O) (awesome, O) (too, O)]
]

3.1.2 Preparing the input: Sliding Window

In the experimentation, we would like to try out if the past and future context of a word plays an important role for predicting the tag of the word. Therefore, we use as an input of the Convolutional Neural Network a matrix $m \in R^{2*context+1}$ where context is a hyperparameter of the model. This matrix is the sliding window that we will move throught the reviews to generate all the inputs. In addition, an empty word is used for padding purposes in the beggining and end of every review. For instance with context 1 our previous example will be:

[
[(", O) (The, O) (food, B)]
[(The, O) (food, B) (was, O)]
[(food, B) (was, O) (delicious, O)]
[(was, O) (delicious, O) (The, O)]
[(delicious, O) (The, O) (live, B)]
[(The, O) (live, B) (music, I)]
[(live, B) (music, I) (was, O)]
[(music, I) (was, O) (awesome, O)]
[(was, O) (awesome, O) (too, O)]
[(awesome, O) (too, O) (", O)]
]

3.2 Model

As mentioned previously, a Convolutional Neural Network is utilized as the classification approach. This network is composed by several layers:

- **Word Embeddings**. This layer transforms all the words into representative vectors using word2vec algorithm. The dimension of these vector is defined as a hyperparameter.
- **Convolution**. Once a several filters have been defined as hyperparameters, this layer is responsible for applying the convolutions.
- **Max-pooling.** This layer selects the maximum value that comes from every convolution in order to reduce the amount of information to be treated.
- **Dropout.** This layer prevents cells to co-adapt by discarding the information of some of them according to a dropout probability defined as hyperparameter.
- **Output.** This layer calculates the score for every class and makes a prediction of the most appropiated class for the input.

3.2.1 Hyperparameters

Hyperparameter	Description
Embedding dimension	The dimension of the vectors coming from word2vec
Filter sizes	Defines the dimension of the filters to be applied. For instance, [3x Embedding dimension]
Filter Amount	Number of filters per filter size
Dropout Probability	Percentage of cells to be kept for the classification
Word Context	Number of past and future words to be taken as the context word

3.2.2 Implementation

The model beforementioned has been implemented using tensorflow framework which provides us with several key operations such as calculating the embeddings from the training set or applying convolutions of a certain filter size.

4 Experiments

In the following table, we can appreciate different configurations of the proposed model. In all of them the dropout probability is set to 0.5 and the amount of filters per filter size is 50.

Id	Word Context	Filters	Filter Amount	Precision	Recall	F1
#1	1	[1,2,3]	50	0.607	0.570	0.588
#2	1	[1,2]	50	0.535	0.549	0.542
#3	1	[2,3]	50	0.549	0.562	0.556
#4	1	[1,2,3]	100	0.633	0.539	0.582
#5	1	[1,2]	100	0.562	0.550	0.556
#6	1	[2,3]	100	0.613	0.57	**0.595**
#7	2	[1,2,3,4,5]	50	0.604	0.555	0.579
#8	2	[1,2,3]	50	0.516	0.457	0.485
#9	2	[3,4,5]	50	0.586	0.580	0.583
#10	2	[3,4]	50	0.582	0.562	0.572
#11	2	[1,2,3,4,5]	100	0.602	0.570	0.586
#12	2	[1,2,3]	100	0.527	0.487	0.506
#13	2	[3,4,5]	100	0.586	0.582	0.584
#14	2	[3,4]	100	0.656	0.542	**0.594**
#15	3	[1,2,3,4,5,6,7]	50	0.626	0.512	0.563
#16	3	[1,2,3,4]	50	0.519	0.435	0.474
#17	3	[3,4,5,6,7]	50	0.596	0.584	0.590
#18	3	[4,5]	50	0.558	0.502	0.528
#19	3	[6,7]	50	0.587	0.535	0.560
#20	3	[1,2,3,4,5,6,7]	100	0.595	0.539	0.566
#21	3	[1,2,3,4]	100	0.505	0.432	0.466
#22	3	[3,4,5,6,7]	100	0.590	0.562	0.576
#23	3	[4,5]	100	0.580	0.512	0.544
#24	3	[6,7]	100	0.590	0.555	0.572

As it can be seen the best results are achieved by configurations #6 and #14 for this experimentation.

5 Conclusion

In this project, a sequential Convolutional Neural Network has been created to detect Opinion Target Expresion. In this section, we compare the results achieved with the ones in the competition and propose several directions for future work.

5.1 Comparison to the official results

Team	F1
IHS-RD.	63.12
Lsislif	62.22
NLANGP	61.49
Our Model	**0.595**
wnlp	57.63
UMDuluthC	50.36
UMDuluthT	50.36
CU-BDDA	36.01

5.2 Future Work

- **Add more features to the input**. In this approximation, we only use as our input data the embeddings coming from the training set. Nevertheless, features such as frequency name lists, word clustering, name lists extracted from external resources, etc are crutial to identify whether an expresion is a target for an opinion.

- **Input overlapping**. As we formatted our input data to care for the context sometimes a too large overlap between different words makes the classification problem harder.

- **Used pre-trained word embeddings**. There are several resources such as GloVe and Google word2vec wich provide us with word embeddings trained with millions of words.