Aspect-Based Sentiment Analysis

Francesco Montano 1744183
montano.1744183@studenti.uniroma1.it

Abstract

Aspect-based sentiment analysis is the task of identifying the aspect terms of a given sentence and the related opinion polarity.

1 Task A: Aspect Term Identification

In this section I am going to approach the problem of the aspect term identification. In order to tackle this problem I have developed different approaches. The first and simpler approach is based on a LSTM (Hochreiter and Schmidhuber, 1997) (I am going to call it baseline model), while the other ones are based on BERT (Devlin et al., 2018).

1.1 Preprocessing

For what concerns the baseline model as preprocessing I have only lower the tokens of the sentence, while for the models based on BERT I have done nothing. This simple preprocessing has been chosen in order to simplify the recostruction of the predicted aspect terms.

1.2 Classification

For all the following models I tried to both use a binary classification and a multi class classification in order to classify aspect terms. In the binary classification each term classified as 1 is considered an aspect term and, in order to deal with aspect terms formed by multiple tokens, I decided to aggregate consecutive terms classified as 1 into a single aspect term.

Instead with the multi class classification I used three labels: *B*, *I* and *O*. *B* is the class for the token that are at the beginning of an aspect term, *I* is the class for the tokens that are inside an aspect term but that are not the first ones and *O* is the class for the tokens that are not inside an aspect term. With the latter approach I am able to handle sentences where there are consecutive aspect terms.

1.3 BiLSTM Model

With this approach I used the *Glove 840B 300d* (Pennington et al., 2014) pretrained embeddings and I have managed the OOV terms with a special token `UNK` associated with a random vector as embedding. This model as said before is based on a LSTM and in particular is based on a BiLSTM in order to obtain contextualized embeddings that depend on both the precedent and subsequent terms. The BiLSTM takes as input the embeddings of the terms of the considered sentence and outputs the contextualized embeddings of each term. Each contextualized embedding than is passed through a multi layer perceptron (MLP) in order to obtain a classification for each term.

1.4 BERT Models

The following approaches are all similar and change in the way of handling terms associated to more than one word piece. All this models use the BERT tokenizer in order to obtain the word pieces of the sentence and their encoding. The encoded sentence than is passed through the pretrained *BERT-large-cased* model in order to obtain the contextualized embedding for each word piece. The contextualized embeddings of the word pieces are used in order to obtain the contextualized embeddings of the terms though the combination techniques that I will explain below. Each term contextualized embedding than passes through a MLP in order to classify it.

The techniques used in order to combine the contextualized embeddings of the word pieces into the contextualized embedding of the associated term are the following ones:

- *SUM*: take the sum of the contextualized embeddings of the word pieces that form the considered term. See Figure 3

- *AVG*: take the average of the contextualized embeddings of the word pieces that form the considered term. See Figure 2
- *MAX*: take the max element wise of the contextualized embeddings of the word pieces that form the considered term. See Figure 1
- *FIRST*: consider only the first word piece of the considered term. See Figure 4

1.5 Task A Results

All the results have been obtained on the concatenation of the two development sets. In Table 1 we can see the results of the different approaches on the task A while using binary classification. As we can see for each BERT approach there is a row for the model that takes only the last hidden layer as output and a row for the model that takes the last four hidden layers and concatenates them. As we can see all the model based on BERT are far better than the baseline model based on the BiLSTM and also that we obtain an improvement in performance when using the last four layers rather than only the last one in BERT. The best overall method is the one that uses the *MAX* combination technique indeed this method achieves the higher F1 score and precision values and a slightly worse recall than the *SUM* method.

In Table 2 we can see the comparison of the baseline model and the BERT model (the ones that take the last four hidden layers) when using binary (rows with B) and multiclass classification (rows with M). As we can see the binary classification goes slightly better than the multiclass classification for all the models and this could be due to the fact that the binary classification is an easier task and that in the development set we do not have any sentence with two consecutive aspect terms (that is the only thing that the multiclass classification model can handle that the binary ones cannot). So the overall best method is the one using the *MAX* combination technique with binary classification and that takes the last four hidden layers as BERT output.

1.6 Train and Hyperparameters

All the models have been trained over the concatenation of the train sets of both the restaurants and laptops datasets. The baseline model has been trained by using *Glove 840B 300d* embeddings, with a 2 layers bidirectiona LSTM with hidden dimension of 256 for 9 epochs with patience 3 and adam optimizer with learning rate of 0.001. In both the BERT models and the baseline model I used dropout of 0.3. The BERT models have been trained with a batch size of 32 and as optimizer the Adam optimizer with a learning rate of $5e-5$. The train lasted a few epochs, indeed the loss minimum has been achieved at the end of the first epoch for all the BERT models. The train parameters of the models based on BERT (discussed in the precedent lines) have been chosen from the ones suggested in (Devlin et al., 2018) for BERT finetuning.

2 Task B: Aspect Term Polarity Classification

In this section I am going to approach the problem of the aspect term polarity classification. In order to tackle this problem I have developed two approaches based on BERT. The first one makes use of the combination techniques explained in the precedent task, while the second one uses a special token in order to represent the aspect term.

2.1 Special Token Approach

With this approach we have that in each sentence we replace the aspect term with a new token <*target-term*> that is initially associated to a random embedding that the BERT model will learn during the train. Each sentence is associated through the BERT tokenizer to the ids of the word pieces in the sentence.Than with the *BERT-large-uncased* model we obtain the contextualized embeddings of each word piece. The contextualized embedding of the <*target-term*> token is passed through a MLP and classified over the four sentiment classes.

2.2 Without Special Token

In this case, rather than replacing the aspect term with a special token I decided to let the model take as input the entire original sentence and than combine the word pieces of the aspect term with two of the precedent explained combination methods: *MAX* and *AVG* that where the most efficient ones in the precedent task. After this also in this case, the contextualized embedding of the aspect term is passed through an MLP and classified over the four sentiment classes.

2.3 Train and Hyperparameters

The different models have been trained over the concatenation of the train set of both restaurants

and laptops dataset with Adam optimizer with learning rate of 0.000_01, dropout of 0.3, by using a weighted loss and patience of 3. In this case I directly tested the model by considering as BERT output the concatenation of the last four hidden layers cause of the increase in performance in the precedent task. In this task I decided to weight the Categorical Cross Entropy loss function in order to tackle the fact that the dataset is unbalanced as we can see from Table 3. The weights that I have used are defined for a class i as follows: $\frac{aspect\_term - aspect\_term\_class\_i}{aspect\_term}$ where $aspect\_term$ is the total number of aspect terms in the trainset and $aspect\_term\_class\_i$ is the number of aspect term with sentiment i in the trainset. With these weights I give more importance to the class with lower examples (the conflict class) than the class with a larger number of examples like for instance the positive class.

In this task I decided also to not take the model with lower loss but the model returned by the last epoch of the train in order to let the model fit better the trainset since I noticed an increase in the scores obtained over the classes with a lower number of instance as we can see in Table 4 where for each model we have a row with the last version in the train (row with *Last*) and the ones with lower loss (the rows with *Min Loss*). From Table 4 we can see that there is always an increase in performances over the last two classes (*Neutral* and *Conflict*) when taking the last version.

2.4 Task B Results

We can see in Table 5 and 6 the results obtained by the different approaches used for this task. As we can see from Table 5 the approach with the special token has worse F1 score over the classes with higher number of examples that the other two methods, but the opposite happens for the two classes with less examples (conflict and neutral). Overall from Table 6 we can see that the approach with the special token obtains the highest Macro F1, and micro F1 (the latter is equal to the case with the method without special token and with *AVG* as combination technique).

3 Task A+B Results

In order to test both the task A and B i decided to test over the concatenation of the development set of the restaurants and laptops dataset. The tested model has been obtained by the concatenation of the model with the *MAX* technique for combining the word pieces for obtaining the aspect terms and the model with the special token in order to classify the polarity of the predicted aspect terms. This model obtained a Macro F1 of 0.538 and a micro F1 of 0.639.

4 Extra

As extra I decided to try do use a new dataset (Jiang et al., 2019) that has 4.297, 500 and 500 train, development and test sentences. In this dataset, rather than having four possible sentiments, only has three of them: *positive, negative* and *neutral*. Each sentence of the dataset is characterized by the fact that there are at least two aspect terms with different sentiment associated.

Over this dataset I trained and tested different models for the aspect term identification task: the ones based on BERT with *MAX*, *AVG* and *SUM* combination techniques that take the last four outputs layer, while for the aspect term polarity classification I trained and tested the model with the special token.

We can see from Table 7 the results of the task A of the methods described above, as we can see in this case the best combination technique has been the *AVG* method with a small margin over the F1 score of the *MAX* method. We can also notice that the model based on the *SUM* combination techniques also over this dataset tends to return high recall and low precision, like over the homework dataset.

From Table 8 we can see the results over the different classes obtained by the model with special token. As we can see, differently from the dataset provided for this homework, the classes are much more balanced and this surely helped the model to achieve better performances. Overall the model with special token has obtained a Macro and micro F1 of 0.82.

	Precision	Recall	F1
Baseline	0.765	0.713	0.738
Max	0.804	0.839	0.821
Max last 4	**0.816**	0.841	**0.828**
Avg	0.790	0.845	0.817
Avg last 4	0.810	0.834	0.822
Sum	0.780	**0.846**	0.812
Sum last 4	0.798	0.829	0.814
First	0.797	0.825	0.811
First last 4	0.803	0.815	0.809

Table 1: Results on the task A of the models with binary classification

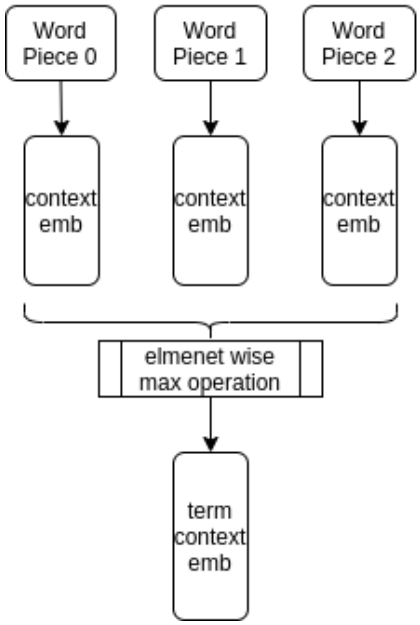

Figure 1: Max combination technique representation: a term formed by three word pieces.

	Precision	Recall	F1
Baseline B	0.765	0.713	0.738
Baseline M	0.753	0.716	0.734
Max B	0.816	**0.841**	**0.828**
Max M	**0.830**	0.808	0.819
Avg B	0.810	0.834	0.822
Avg M	0.817	0.822	0.819
Sum B	0.798	0.829	0.814
Sum M	0.774	0.834	0.803

Table 2: Results on the task A of the models with binary B and multiclass classification M

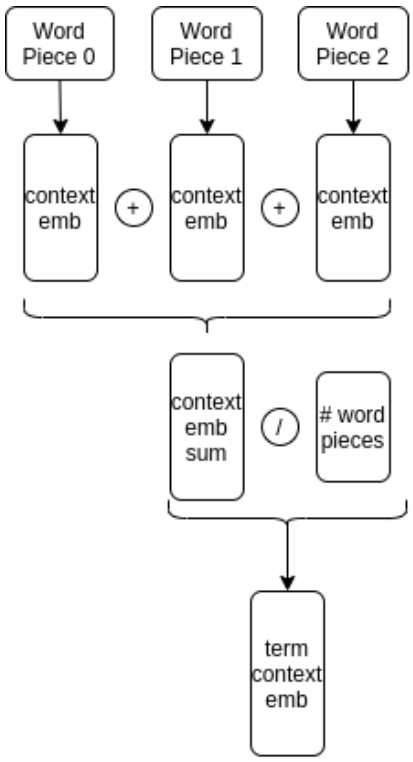

Figure 2: Average combination technique representation: a term formed by three word pieces.

	Train	Development
Posive	2605	546
Negative	1364	307
Neutral	877	216
Conflict	111	25

Table 3: Number of examples per class in both Train and Development set

	Positive	Negative	Neutral	Conflict
Special Token Last	0.87	0.76	**0.60**	**0.37**
Special Token Min Loss	0.86	0.76	0.55	0.00
Max Last	0.86	**0.79**	0.51	0.21
Max Min Loss	0.87	0.73	0.58	0.17
Avg Last	**0.88**	**0.79**	0.59	0.20
Avg Min Loss	0.87	0.78	0.52	0.00

Table 4: F1 score on the task B over the different classes comparing the model with the min loss and the last model of the training

	Special Token	Avg	Max
Positive F1	0.87	**0.88**	0.86
Negative F1	0.76	**0.79**	**0.79**
Neutral F1	**0.60**	059	0.51
Conflict F1	**0.37**	0.20	0.21

Table 5: Results on the task B with respect to the different classes

	Special Token	Avg	Max
Micro F1	**0.78**	**0.78**	0.76
Macro F1	**0.65**	0.61	0.60

Table 6: Results on the task B in terms of Macro and Micro F1

	Precision	Recall	F1
Avg	**0.725**	0.783	**0.753**
Sum	0.681	**0.830**	0.748
Max	0.722	0.779	0.750

Table 7: Results on the task A over the new dataset

	Precision	Recall	F1	Supp
Positive	0.83	0.81	0.82	400
Negative	0.76	0.82	0.79	329
Neutral	0.84	0.84	0.84	607

Table 8: Results on the task B over the new dataset with respect to the different classes with the model with special token

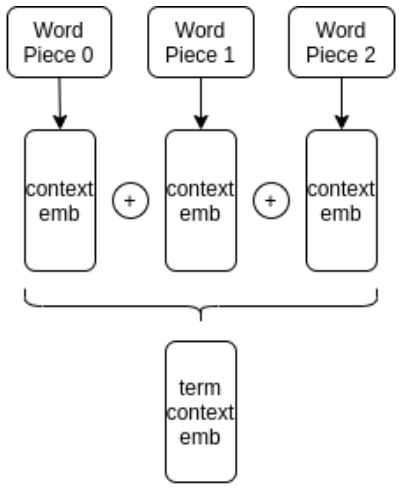

Figure 3: Sum combination technique representation: a term formed by three word pieces.

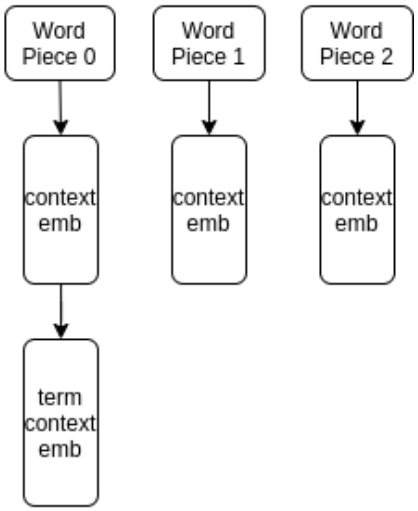

Figure 4: First combination technique representation: a term formed by three word pieces.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6280–6285, Hong Kong, China. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.