Homework 2: Aspect-Based Sentiment Analysis

Giammarco D'Alessandro
Sapienza University of Rome, matricola 1753102
dalessandro.1753102@studenti.uniroma1.it

4^{th} September 2021

Abstract

This is the report for the Natural Language Processing course (a.y. 20/21) second homework on Aspect-Based Sentiment analysis.

1 Introduction

In NLP Aspect-Based Sentiment Analysis (ABSA) is a particular task where the goal is to predict the aspect terms regarding a given set of target entities, an the sentiment expressed towards each aspect. In this assignment we were also asked to classify the category of each aspect term and the overall sentiment polarity expressed towards the category. To deal with its complexity, the ABSA problem has been divided in 4 sub-tasks (A-B mandatory, C-D extra) :

- A) aspect term identification
- B) aspect term polarity classification
- C) aspect category classification
- D) aspect category polarity classification.

I have thus implemented four models to address each one of the sub-task, and I have then combined them to perform the higher level task. Respectively: *TaskATermExtractionModel*, *TaskBAspectSentimentModel*, *TaskCCategoryExtractionModel* and *TaskDCategorySentimentModel*.

2 Dataset and pre-processing

The dataset is organized in two parts, each one regarding one of the two given target entities: restaurants and laptops (Tab.1). Apart from the topic, the two parts are similar, and the main difference is that the task C and D can be performed only with the restaurants data, because only here ground truth information on categories, and categories sentiment, is present (and thus C and D model have been tested only on these data). Due to the differences between the four tasks, I have chosen to pre-process the data in various ways, according to the characteristics of each sub-problem. In general, as common to many NLP tasks, I have exploited the feature extraction power of transformers architectures such as BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019), to get a more meaningful contextual representation.

3 Task Models

The models presented are pretty similar, they differ mostly in the data pre-processing step and in the configuration of the learning task, and thus of the final output processing. The following sections explain each model in detail.

3.1 A: Aspect term indentification

Task A consist in identifying the terms in a sentence (i.e. the words), that represent the subject that a statement is describing, thus I have decided to set it up as a token classification tasks, exploiting the Named Entity Recognition BIO (Beginning-Inside-Outside) tagging format (Ramshaw and Marcus, 1995). To solve the sub-task I have tried two different approaches: a standard sequence encoding approach and a transformer based solution.

In the former I have tokenize the input sentences and build a vocabulary with the GloVE 6B-100 embeddings, as latent word representation (Pennington et al., 2014). Then the data are passed through a BiLSTM layer, and in the end through a small MLP classifier composed of two linear layers.

The second approach follows the same BIO tagging format, but the word representation rely on an implementation of a pre-trained BERT transformer for token classification, that I have fine tuned adding a custom MLP classifier head. The MLP is composed of two linear layers separated by a ReLU activation layer (*TaskATermExtractionModel* model implements the second and best performing solution) (Tab.2).

3.2 B: Aspect term polarity classification

Task B consist in classifying the the polarity ("positive", "negative", "neutral" or "conflict") expressed towards each target term. I have set up this task as a sequence classification task, trying to classify each $< sentence, aspect\_term >$ couple to the correct polarity. Moreover notice that the four polarity label provided were not enough to deal with sentences in the dataset with no target, and I have added an extra dummy label ("un-polarized") to classify those special cases (label that has not been considered in the metrics to not affect performances).

The *TaskBAspectSentimentModel* model implements a pre-trained BERT for sequence classification, with a custom MLP classifier on top of it. The MLP is composed of two linear layers separated by a GELU (Gaussian Error Linaer Unit) activation layer (Tab.3).

3.3 C: Aspect category classification

Task C consist in identifying the category ("anecdotes/miscellaneous", "price", "food", "ambience", "service") to which the aspect term belongs to. I have implemented this task as a sequence multi-label classification, as more categories may appear in a sentence (they are not mutually exclusive as in a typical multi-class classification problem). In this case I have chosen RoBERTa, an optimized version of BERT that modifies key hyperparameters, to better approach the higher complexity of the multi-labeling task.

The *TaskCCategoryExtractionModel* model, that address task C, implements a pre-trained RoBERTa transformer for sequence classification, with a custom MLP classifier on top of it (two linear layers with a GELU activation in between)(Tab.4).

3.4 D: Aspect category polarity classification

Task D is very similar to task B, and consists in classifying the sentiment polarity expressed towards each category. Again the classes are the same as in B ("positive", "negative", "neutral" or "conflict"), but in this case there is no need to add an extra label because every entry of the dataset has a category field, even when the targets are missing. Again I have chosen RoBERTa transformer architecture, in its sequence classification configuration, feeding as input the $< sentence, aspect\_category >$ pairs, to keep up with the more abstract notion of category, compared to the aspect term.

Thus the*TaskDCategorySentimentModel* model exploits a pre-trained RoBERTa for sequence classification transformer, fine-tuned with a custom classifier head, here composed of two linear layers with a GELU activation layer in the middle (Tab.5).

4 Training set up

Training and evaluation have been carried out mainly considering different metrics, to describes the performances of the different classification tasks. Mainly F1 score has been considered, applying both micro and macro reductions, to better analyzed the performances over the unbalanced dataset. Notice that for task A the F1 score was not computed over the correctly predicted tokens, but comparing the sets of ground truth aspect terms and the predicted ones, to avoid the bias of the O-outside tokens, representing the majority and thus tainting the measure. I have also taken into account the accuracy score, especially in task B and D, to analyze the training process and easily detect model overfitting.

To reduce overfitting I have also implemented an early stopping technique to regulate the training loop, with a patience level of 3 for all models, considering that fin-tuned BERT-based models tend to achieve the best performances with really few epoch training (3-4) (Devlin et al., 2019).

5 Experiments

The configuration of the models shown above are the parameters that allowed me to obtain the best performances with each of the models. (Figs.**??**,**??** and Tabs.**??**, **??**). Indeed I tried various configuration at each step of the two models before achieving those results (The values are reported in the corresponding tables below)

Pre-processing: For task A I have tried to build the vocabulary with different size of the pre-trained GloVe embeddings, but greater than 100 sizes (200,300) badly influenced the performances. This is probably due to the fact that those embeddings are trained on different data and tasks, and lower dimension representation are less task-specific, granting them a better generalization.

Activation: I have tested different activation functions (GELU, ReLU, tanh), though the GELU outperforms the other in almost all cases (for task A the ReLU works better), scoring even better than the tanh with RoBERTa (being tanh RoBERTa

classfier default).

Optimizer: I have trained all the model using the advised optimizer for BERT, AdamW (Adam decoupled weight decay version, (Loshchilov and Hutter, 2017)), with learning rate 3e-5. This has worked pretty well in all tasks, except for the task D, for which I had to reduce learning rate to 3e-6 and increase the dropout to avoid a huge overfitting (100% accuracy and F1 scores after one epoch).

6 Results

The following tables and plots reports the results of my experiments.

no.	**rest.**	**laptop**
entries	2500	2500
words	4000	3712
targets	1107	883

Table 1: *Train datasets compositions, restaurants (rest.) and laptop parts.*

TaskA model	prec.	recall	f1
BERT res2res	**0.6569**	**0.6751**	**0.6673**
BERT lap2res	0.6321	0.6249	0.5378
Base res2lap	0.6025	0.5738	0.4151
class precision	B	I	O
BERT res2res	**0.8707**	**0.8094**	**0.9866**
BERT lap2res	0.8476	0.7377	0.9399
Base res2lap	0.4642	0.4348	0.9630

Table 2: *Task A top 3 performing models. For this model also per-class precision was considered, for each tag as in BIO (Beginning Inside Outside) scheme.*

B models	*m*-prec.	*m*-f1	*M*-f1
BERT r2r gelu	**0.7727**	**0.7674**	**0.5387**
BERT r2l gelu	0.7518	0.7491	0.5357
BERT r2r relu	0.7025	0.6738	0.4951

Table 3: *The performances of top 3 task B models, analysing micro precision (m-prec), micro F1 (m-f1) and macro F1 (M-f1)*

C models	*m*-prec.	*m*-f1	*M*-f1
RoBERTa r2r gelu	0.8812	**0.8825**	**0.8744**
RoBERTa r2l gelu	**0.9418**	0.8683	0.8492
RoBERTa r2r tanh	0.9273	0.8525	0.8322

Table 4: *The performances of top 3 task C models, analysing micro precision (m-prec), micro F1 (m-f1) and macro F1 (M-f1)*

D models	*m*-prec.	*m*-f1	*M*-f1
RoBERTa r2r lr1e6	0.9532	**0.9532**	**0.7388**
RoBERTa r2l lr3e6	**0.9638**	0.9222	0.7124
RoBERTa r2r lr1e5	0.9431	0.8525	0.6929

Table 5: *The performances of top 3 task D models, analysing micro precision (m-prec), micro F1 (m-f1) and macro F1 (M-f1)*

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040.