Target Extraction and Polarity Classification
An experimental implementation aiming smaller

Giovana Meloni Craveiro (225249)
University of Trento
Via Sommarive, 9, 38123 Povo,Trento TN
g.melonicraveiro@studenti.unitn.it

Abstract

This work is an experimental implementation of fine-tuning pre-trained models for two natural language understanding tasks: target extraction and polarity classification. Each task was performed using a distilBERT model and smart batching, with the goal to increase efficiency.

1 Introduction

An important part of artificial intelligence relies on natural language understanding (NLU), as properly supervising machine learning is fundamental to ensure that progress on the field is respectful and positive to everyone. Among common NLU tasks in the subfield of sentiment analysis are target extraction and polarity classification. The first one refers to identifying determined words inside a text sample. And the latter one refers to categorizing the input as positive, negative or neutral. Pre-trained models that perform these tasks with state-of-the-art results have been released and made publicly available over the past years. As machines learn based on the data that is fed to them, a relevant factor that affects performance is the similarity between the data they were trained on and the data they should act on. As such, the proposition of this work is to fine-tune a pre-trained model for target extraction and polarity classification on a dataset that contains laptop reviews. Classification tasks can be performed by analysing each token separately or the sentence as a whole, and by limiting the targets to one per sentence or more. The initial intent was to perform both tasks span-based, analysing the whole sequence at once, with multi-target extraction followed by polarity classification in a pipeline, as it has been shown by previous research that it provides best results (Hu et al. 2019). However, as it is also quite hard to find information on how to perform span classification and multi-target extraction, the path chosen was to perform span single target extraction and polarity classification of the single target labeled, with a sequence classification pre-trained model. Additionally, efficiency was aimed at this project, as it is an ethical guideline for the field (Bender et al, 2021). To achieve that, distilBERT, a smaller, lighter version of BERT was used, and smart batching was performed at the dataset. The following sections detail the problem, model, dataset, approaches, limitations and results of this project.

2 Problem Statement

The whole algorithm can be simply described in a few steps. First, the dataset is read, tokenized and formatted. Then, the model is configured. The model must train with a split of the dataset and subsequently, evaluate the unseen group. As a final step, some measures are calculated and reported. Each of the tasks must go individually through

configuration, training, evaluation and results report. The dataset manipulation, contrarily, is common for both. But as at each epoch of training, it is necessary to select a new random order for the batches and smart batching was adopted, the input vectors were recalculated at every training and evaluation epoch. Smart batching consists of excluding unnecessary padding tokens, by padding each sentence only until the length of the longest sentence in its batch, as opposed to until the length of the longest sentence in the dataset. This significantly reduces memory and time without decreasing performance.

For the target extraction task, the input consists of a set of sentences. Each word of the sentence is labeled as part of a target if it is a component or characteristic in the technological domain, such as battery life. The dataset contained a certain number of targets per sentence, which varied from 0 to many, but was formatted to include only the first target of the sentence. As it is a span extraction, the model intends to predict the start and the end position of the target in the sentence. To perform span extraction, the most popular model is question and answering model. Therefore, the input given consists of the input sentence, an attention mask, the start positions and the end positions of the first target, if present, of the sentence. The model trains and afterwards performs the evaluation at a different file containing sentences with the same style. The output of each sequence includes two vectors with the length of the sequence, in which each position represents the probability of the token being, respectively, the starting and ending position of the targeted span.

The labels given for the polarity classification machine learning are simply a single polarity for each sentence, which corresponds to the polarity of the first target of that sentence. An issue was found at this step, as not all sentences have targets, and thus, polarities. This same problem was simply overcome at target extraction by the model itself, as it handles out-of-sequence labels by ignoring them. To work around it in polarity classification, however, in addition to *positive(0), negative(1)* and *neutral (2)* labels, *absent(3)* label was given to all sentences without targeted spans when formatting input data. This approach is experimental, it might not be advisable. The sequence classification model outputs the probability of the sequence belonging to each of the labels. All predictions to the *absent* category were ignored in the results. Individual and total accuracy, precision, recall and F1-scores were calculated at the end for both tasks.

3 Data Analysis

The dataset used is SemEval2014 (Li et al, 2019). It includes a training file with 3045 sentences and a test file with 800 sentences. Each sentence is associated with a label section, that contains one label for each token. The labels are “O” for tokens that are not identified as anything, “T-POS” for positive targeted entities, “T-NEG” for negative targeted entities and “T-NEU” for neutral targeted entities. All of the entries are reviews related to laptop domain. The sentence labels were manipulated to include only the first target of each sentence, in order to fit the model input.

4 Approaches and limitations

The initial proposition was inspired by the paper given as further orientation on the project presentation (Hu et al, 2019) and as such included the implementation of a span-based multi-target extraction, followed by a polarity classification in a pipeline format, since it was the architecture that provided the best results. First efforts went in the direction of understanding the code, running it, and adapting its functions. However, the memory requirement for the code

execution became a barrier, softened by decrease of the batch size from 32 to 8. The unstable success of the machine used to run the algorithm was also reason for frustration. The decisions to use the smaller version of BERT, distilBERT, and performing smart batching on the samples were taken and efforts were put into reorganizing and formatting the dataset. Furthermore, lack of clarity and further resources for replicability motivated a shift towards span-based single target extraction and polarity classification separately. A functional algorithm was then created, feeding the reorganized dataset into a distilBERT question answering model, configured with a mix-sourced set of recommendations. The model was able to complete training and evaluation. At its example, the polarity classification task was then elaborated. Functions that calculate accuracy, precision, recall and F1-score of the outputs were created as a final step. Results, however, were peculiar and poor, leading suspicions to a flawed approach, that results in bad machine learning, when all these steps in these specific circumstances were put together.

5 Models

The model used for target extraction was distilBERT question and answering, while the model used for polarity classification was distilBERT for sequence classification.
The tokenizer used at both was distilBERT tokenizer.
DistilBERT default configurations modifications were also experimented to decrease model size. The changed configurations, based on Pertschuk (2020), included number of layers from 12 to 4, number of dimensions from 768 to 512 and intermediate dimensions number from 3200 to 1200. These changes caused significant reduction of execution time of the polarity classification from approximately 20 minutes to less than 5. However, they also showed results weaker enough (from 0.4 accuracy to 0.2) to motivate the shift back to the original distilBERT configurations. As for the target extraction model, the modifications were maintained since the execution time was already around 17 minutes.
Execution time is around 17-19 minutes for both of the tasks, which results around 35 minutes for the whole program to run. It is configured, nevertheless, to run each at a time, by the user's choice.
The learning rate used for both tasks was $2e^{-5}$, as used in Hu et al. (2019). McCormick (2020) used a learning rate of $5e^{-5}$ for sequence classification task, which was also experimented at the polarity classification task due to poor results, but did not provide relevant improvement.
The optimizer used was AdamW from the transformers python library, also used by McCormick (2020).
The number of warmup steps for the scheduler and optimizer was 0, following McCormick (2020).
A clip grad function was also used to prevent the exploding gradients problem.
Batch size chosen was 8. Since this value supported a BERT run of (Hu et al. 2019) algorithm in the used machine, it was considered a good option for efficiency reasons.
The default number of epochs chosen was 3, following Hu et al. (2019). Significant worse performance is shown by polarity classification model, when the epochs number is changed to 1, as opposed to target extraction, which does not have its results intensely decreased.
No seed was set, so the results vary at each run and are not easily replicable, but as they are also not satisfying or stable, this aspect is minor at this phase.

6 Results and Conclusions

The measure of the results is unfortunately not reliable. The peculiar results open margin to suspicion that the outputs from the model were not correctly interpreted. The calculation of precision, recall and

F1-score are based on the quantities of true positives, false positives and false negatives to each category. Even though the formulas are correct, further investigations should be carried prior to any conclusions on the source of the poor results.
Nevertheless, the referred results are:

	Target Extraction	Polarity Classification
Accuracy	0.185	0.42
Precision	-	0.37
Recall	-	0.53
F1-Score	-	0.44

Additionally, the model never predicts *neutral* as the polarity.

7 Future Work

Enhancements to this repository would be highly beneficial. At first, a revision of the entire procedure and measure calculation should be conducted, followed by experiments of different parameter configurations on the models. An emphasis on the proper method to handle tokens, spans and sentences that do not have a class on classification tasks or are not/ do not have entities on target extraction classifications should also be given.
After both the separate models provide satisfying results, an implementation line could follow the path of multi-target extraction and span classification, as to cover the full set of original labels of the dataset.
Furthermore, the pipeline that uses the results of the target extraction to perform the polarity classification would be interesting.

8 Final Remarks

Although this work has failed on presenting decent performance, it has served its goal very well. The objectives of gaining practical experience manipulating bert and its variants, coding natural language understanding tasks, gaining deeper knowledge and confidence on related concepts were all reached. Paths on mitigating the problems and pursuing next steps could intuitively be followed at this point. Similar tasks can now also be welcomed with much more knowledge, confidence and speed. The code is publicly available at https://github.com/gicraveiro/Targeted_Sentiment_Analysis_Span-Based/

9 References

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019b. Open-domain targeted sentiment analysis via span-based extraction and classification. In ACL, pages 537–546.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . In Conference on Fairness, Accountability, and Transparency (FAccT ’21), March 3–10, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3442188.3445922

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In Proceedings of AAAI.

McCormick, Chris. 2020. Smart Batching Tutorial - Speed Up BERT Training. Available at: http://mccormickml.com/2020/07/29/smart-batching-tutorial/ (Accessed: 24 August 2021)

Pertschuk, Jack. 2020. TinyBERT for Search: 10x faster and 20x smaller than BERT. Available at: https://towardsdatascience.com/tinybert-for-search-10x-faster-and-20x-smaller-than-bert-74cd1b6b5aec (Accessed: 24 August 2021)

Alammar, Jay. 2019. A visual guide to using BERT for the first time Available at: http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/ (Accessed: 24 August 2021)

Borcan, Marius. 2020. BERT NLP: DistilBERT to build a Question Answering System. Available at: https://programmerbackpack.com/bert-nlp-using-distilbert-to-build-a-question-answering-system/ (Accessed: 24 August 2021)

Pogrebivsky, Steven. 2021. Precision, recall and F1 Explained (In Plain English). Available at: https://datagroomr.com/precision-recall-and-f1-explained-in-plain-english/ (Accessed: 24 August 2021)