Homework 2 for Natural Language Processing 2021: Aspect-Based Sentiment Analysis

Salvatore Cognetta
Sapienza University, Rome
cognetta.1874383@studenti.uniroma1.it

1 Introduction

Aspect-Based Sentiment Analysis (ABSA) consists in the resolution of two major problems: aspect term extraction and term polarity classification. The first one extracts the aspect terms or features of an entity (in our case restaurants and laptops) from a given sentence. The second one determines the sentiment polarity (positive, negative, neutral or conflict) of each aspect of that entity.
These tasks can be very useful, for example companies are interested in the opinion of customers towards their products. This can help the company implementing marketing solutions or correcting products. To this end, performing an automated analysis of the user opinions becomes a crucial issue.
This is a challenging task for NLP and it's the main scope of this work, which addresses it using two different approaches: one with LSTM (Long Short-Term Memory) (1) and one with Bert (Bidirectional Encoder Representations from Transformers) (2).

2 Data preprocessing

The dataset presents a set of sentences, associated with one or more target words that identifies the aspect terms.
Each of these sentences is decomposed and tokenized in a vector. After that, depending on the subsequent used method for the ABSA task, there are used two different approaches to encode the vector of words in data that can be given as in input to a neural network.
In particular there can be described two techniques for the word encoding:

- a vocabulary of unique words, created from all the sentences inside the complete dataset, is created for the LSTM approach, including special token for the unknown words. Each word inside the vocabulary has a unique index and for each sentence there will be the encoding of words into indices;
- a pretrained tokenizer from HuggingFace is used to encode token strings to ids.

No words or puctuation are removed from the sentences.

3 Aspect Term Extraction

The first step in ABSA task can be identified in aspect term extraction (ATE). Starting from the work of Zhiqiang Toh and Wenting Wang. 2014 (3), this problem can be seen as a Named-Entity Recognition (NER) (4) task, in which each word is labelled following the traditional IOB format (short for Inside, Outside, Beginning).
For this reason each sentence is decomposed in a list of words, each word, that can be called token, are labelled with the corresponding IOB tag. The strategy is the following:

- **B-tag**: singular tokens that represent the aspect term are labelled with the B tag;
- **I-tag**: if an aspect term consists of multiple tokens, the first token receives the B label and the rest receive the I label;
- **O-tag**: tokens that are not aspect terms are labelled with O.

After preprocessing data and labels, the training is done with two different architectures: LSTM and Bert.

3.1 LSTM method

For the LSTM technique this work take inspiration from Athanasios Giannakopoulos et al. 2017 (5), in which the authors used a BiLSTM architecture with fastText pretrained word embeddings. On top of the LSTM network is placed a linear classifier with 3 output labels, one for each IOB label.

In this work the LSTM architecture is used with no pretrained word embeddings , moreover a dropout layer used during training is placed right before the last linear layer.

After the training some data post-processing is necessary, to decode the vocabulary indices in words and for the aspect term generation. In particular, considering that the aspect term can consists of multiple words, for each sentence consecutives B and I labels are seen as a unique term.

Different experiments are done on this architecture, but the results are unsatisfactory with respect to similar works using other architectures.

A summary of the results for Aspect Term Extraction (ATE) using the LSTM is listed in table [1] alongside with confusion matrix [3].

3.2 Bert method

Bert is a transformer-based machine learning technique developed by Google, really powerful for NLP tasks like NER. Finetuning Bert for this task, however, is really computationally heavy; for this reason a faster (*distilled*) version of Bert is used in this work: DistilBert, from Sanh et al. 2020 (6).

First of all the sentences, as explained in the data preprocessing paragraph, are tokenized with a pretrained tokenizer, instead of an home-made vocabulary. This tokenizer, as showed in the figure [1], adds specials tokens to the sentence:

- $[CLS]$: a special classification token representing the class of the input, it's the first token of every sequence. The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks.;
- $[SEP]$: is a special separator token (e.g. separating questions/answers).

After the tokenization part each token is associated to the corresponding IOB tag, in this case two different tests are done: one using the classic IOB tags and another one dropping the I tag and using only the OB tags as explained in the next section.

3.3 Experiments

3.3.1 DistilBert - IOB format

In the first experiment the classical IOB tagging is used, therefore each token is associated to the corresponding tag, as explained in the beginning of the section.

After this, a simple process of finetuning of *DistilBertForSequenceClassification* is done and the results were really exciting, outperforming the previous architecture.

Using the Adam optimizer with a learning rate od $2e - 5$ and a weight decay of 0.01 this method reached an F1-score of 81.53 and 69.22 respectively on the restaurant validation set and laptop validation set.

A summary of the results for Aspect Term Extraction (ATE) using the IOB format is listed in table [2] and [3] alongside with confusion matrices [4] and [5].

3.3.2 DistilBert - OB format

To improve the overall F1-score, instead of encoding the labels with the IOB format, only the O and B label are considered. Therefore the work is transformed from a multilabel classification problem to a binary classification problem.

Also in this case finetuning of *DistilBertForSequenceClassification* is done and using the same hyperparameters for the optimizer the F1-score reached values of 83.42 and 71.81 respectively on the restaurant validation set and laptop validation set.

A summary of the results for Aspect Term Extraction (ATE) using the OB format is listed in table [4] and [5] alongside with confusion matrices [6] and [7].

4 Sentiment Polarity

After the aspect terms identification the subsequent task consists in sentiment polarity identification. In fact each term can be either positive, negative or neutral, there is however a special case, the conflict label. The last one, in fact, is used when there is more then one constrasting polarity.

In this case after the preprocessing, the Bert tokenizer used is different, it takes the sentence and the aspect term concatenated and the resulting tokenized sentence is

$$[CLS] sentence [SEP] aspect\ term [SEP]$$

Because each sentence can contain more than one aspect term, for each of it the pair sentence-term is treated uniquely, doing post processing on the data to assemble back the sentences with it's aspect term polarities.

For sentiment polarity is used another version of DistilBert, DistilBertForSequenceClassification which has a linear classifier on top of it.
After some finetuning the results with this architecture were quite good, except for the conflict label, because in the dataset are present very fews instances of it. The F1-score is $77.59(micro)$ and $77.35(micro)$ respectively on the restaurant validation set and laptop validation set.
A summary of the results for sentiment analysis, for each polarity class, is listed in table [6] and [7].

4.1 Experiments

4.1.1 Threshold value for conflict label

After some finetuning the results with this architecture were quite good, except for the conflict label, because in the dataset are present very fews instances of it. For this reason, taking into account the definition of the conflict label, during inference is measured the 'uncertainty' of the prediction and if the first two logits of the prediction are below a certain threshold value, the polarity is setted to conflict.
In this case the performance are similar overall with an F1-score of $77.35(micro)$ and $77.50(micro)$ respectively on the restaurant validation set and laptop validation set, but the performance on the conflict label are improved from 0 to 32.26 for F1-score for restuarant and from 25 to 28.57 for F1-score.
A summary of the results for sentiment analysis, for each polarity class, is listed in table [8] and [9] alongside with confusion matrices [6] and [7].

5 Category Extraction

In opposition to aspect term extraction, where the target terms are not predefined, for category extraction there are five different categories which each sentence can belong: anecdotes/miscellaneous, price, food, ambience and service.
This task can be seen as a multilabel-multiclass classification problem, because each sentence can be associated to 0 or more than one category.
In this case a multilabel classification head is placed on top of the pooled output, given as output by the Bert transformer. The classification head has 5 different Linear Layer as output, one for each category, preceeded by another Linear Layer concatenated by a ReLU activation function and a Dropout layer. A representation of the architecture can be seen at the figure [2]
A Binary Cross-Entropy Loss for each Linear layer with sigmoid activation function is used to compute each one of the 5 different losses, collected togheter with a simple mean.
Using the Adam optimizer with a learning rate of $2e - 5$ and a dropout probability of 0.4 the results are quite good, with an F1-score of 70.13 for the restaurant dataset.
A summary of the results for category extraction analysis, for each category class, is listed in table [10] alongside with confusion matrices [10], [11], [12], [13] and [14].

6 Category Polarity

This task is almost identical to the aspect term sentiment polarity one. In fact it is treated as before, using a toknenizer that takes the sentence and the category concatenated. There can be multiple categories-polarity pair associated to each sentence, for this reason, as for the previous case, each sentence-category pair is treated as unique instance of the dataset.

$$[CLS]sentence[SEP]category[SEP]$$

Also in this case, considering the particular nature of the conflict label, a treshold value is setted, below wich the polarity is considered in 'conflict'.
Using DistilBertForSequenceClassification the results are listed in the table [11].

7 Conclusion

Each task is kept separetly, using different networks for each one, in order to keep the project simple and athomic.
The frameworks and libraries used in this work are different: for the model architectures PyTorch is used alongside with PyTorch Lightning used to simplify the training process and model savings. For Bert transformer and tokenizers the transformers library is used, made available by the HuggingFace community.

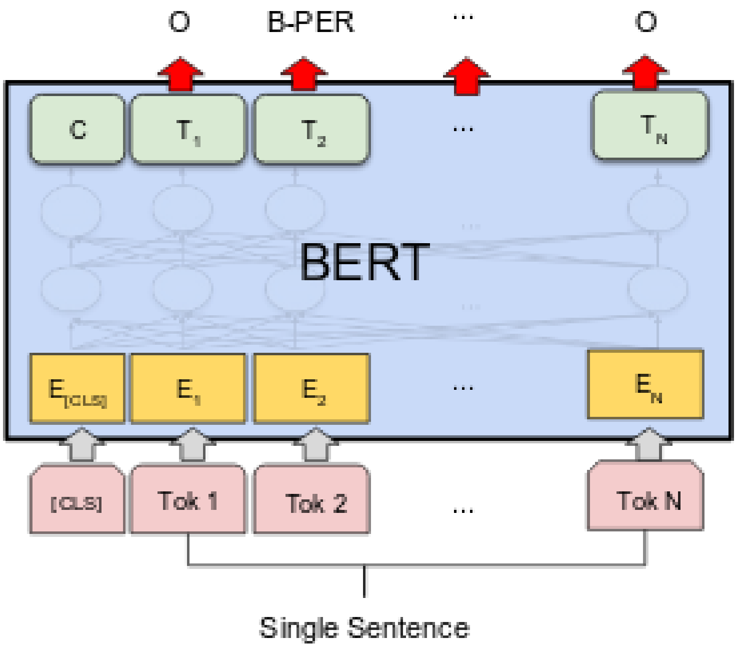

Figure 1: Bert for Token Classification, https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/

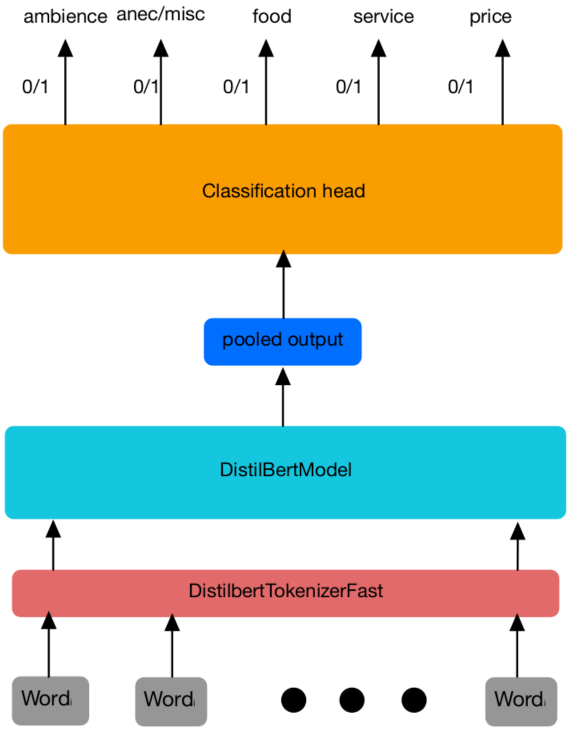

Figure 2: Model architecture for Category Extraction

Metric	Value
Precision	52.03
F1	15.66

Table 1: ATE: LSTM method.
Evaluation on complete dataset

Metric	Value
Precision	80.68
Recall	82.39
F1	81.53

Table 2: ATE: BERT-IOB method.
Evaluation on Restaurant dataset

Metric	Value
Precision	68.03
Recall	70.44
F1	69.22

Table 3: ATE: BERT-IOB method.
Evaluation on Laptop dataset

Metric	Value
Precision	81.78
Recall	85.13
F1	83.42

Table 4: ATE: BERT-OB method.
Evaluation on Restaurant dataset

Metric	Value
Precision	71.06
Recall	72.58
F1	71.81

Table 5: ATE: BERT-OB method.
Evaluation on Laptop dataset

Polarity	Precision	Recall	F1
positive	84.68	77.21	80.77
negative	76.38	88.99	82.20
neutral	75.38	62.03	68.06
conflict	0.00	0.00	0.00

Table 6: Sentiment polarity: BERT method.
Evaluation on Laptop dataset

Polarity	Precision	Recall	F1
positive	83.89	91.24	87.41
negative	73.73	82.08	77.68
neutral	71.19	48.28	57.53
conflict	50.00	16.67	25.00

Table 7: Sentiment polarity: BERT method.
Evaluation on Restaurant dataset

Polarity	Precision	Recall	F1
positive	84.55	76.47	80.31
negative	78.23	88.99	83.26
neutral	75.00	60.76	67.13
conflict	22.22	40.00	28.57

Table 8: Sentiment polarity: BERT method usign threshold value.
Evaluation on Laptop dataset

Polarity	Precision	Recall	F1
positive	84.69	91.24	87.84
negative	75.00	82.08	78.38
neutral	71.93	47.13	56.94
conflict	38.46	27.78	32.26

Table 9: Sentiment polarity: BERT method usign threshold value.
Evaluation on Restaurant dataset

Category	Precision	Recall	F1
anecdotes/miscellaneous	74.05	71.73	72.87
price:	73.33	41.51	53.01
food	82.67	74.55	78.40
ambience	38.02	60.53	46.70
service	88.04	68.07	76.78

Table 10: Category extraction: evaluation on Restaurant dataset

Polarity	Precision	Recall	F1
positive	65.00	62.23	63.59
negative	46.86	49.10	47.95
neutral	45.71	35.96	40.25
conflict	28.00	22.58	25.00

Table 11: Category polarity: evaluation on Restaurant dataset

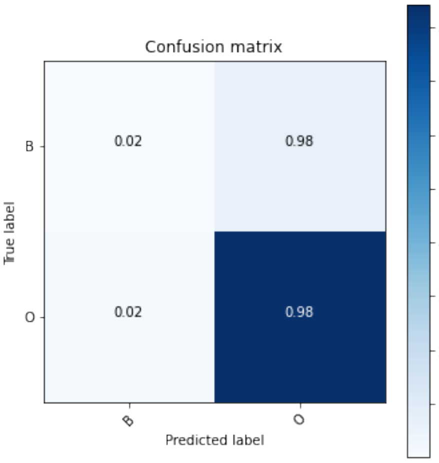

Figure 3: Confusion matrix of ATE on full dataset using LSTM method

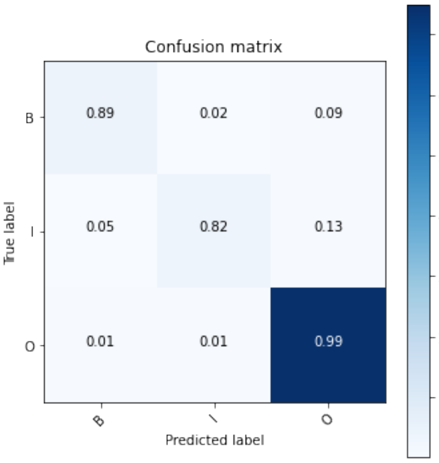

Figure 4: Confusion matrix of ATE on Restaurant dataset using IOB format

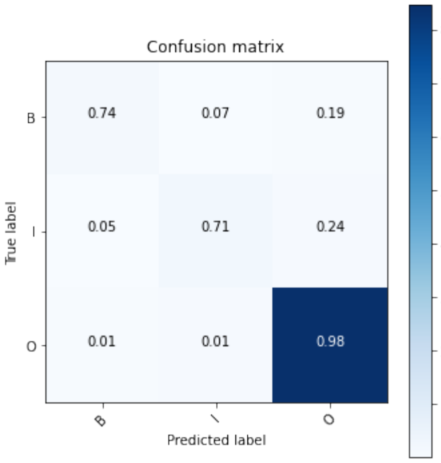

Figure 5: Confusion matrix of ATE on Laptop dataset using IOB format

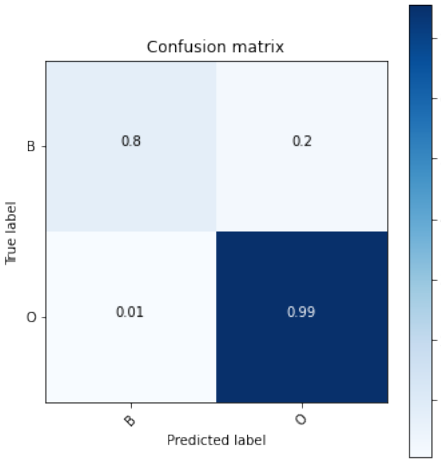

Figure 6: Confusion matrix of ATE on Restaurant dataset using OB format

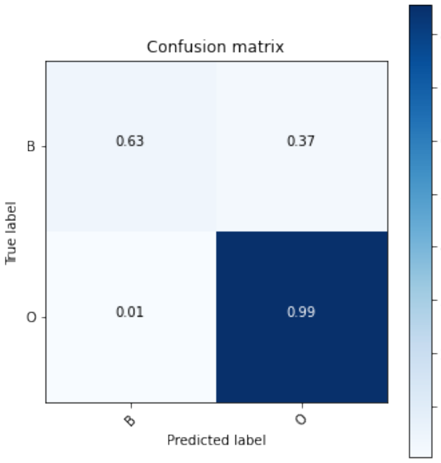

Figure 7: Confusion matrix of ATE on Laptop dataset using OB format

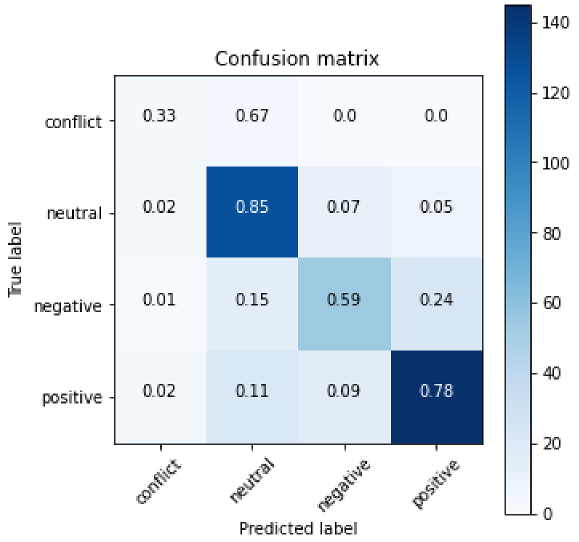

Figure 8: Confusion matrix of sentiment polarity on Laptop dataset using conflict threshold

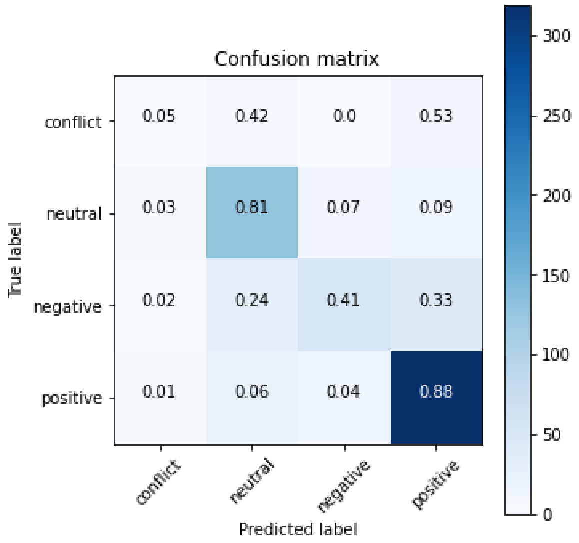

Figure 9: RConfusion matrix of sentiment polarity on Laptop dataset using conflict threshold

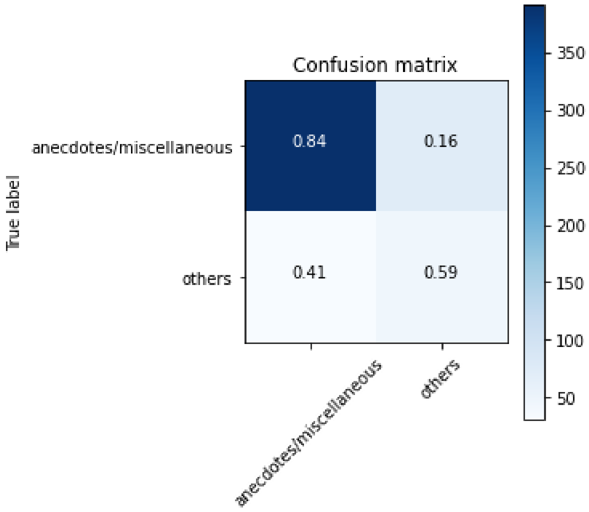

Figure 10: Confusion matrix of category extraction for anecdotes/miscellaneous

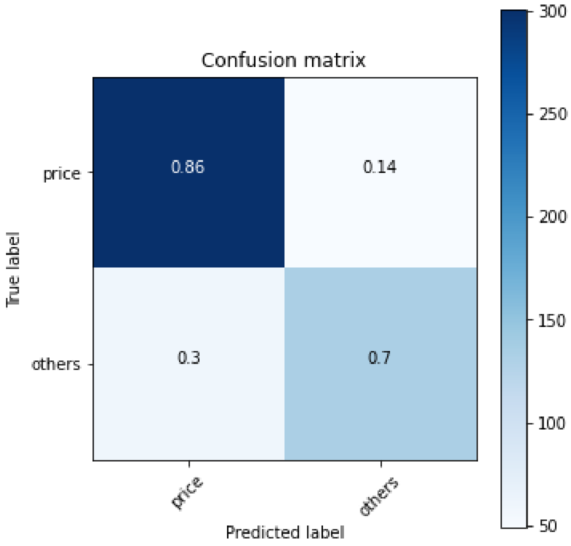

Figure 11: Confusion matrix of category extraction for price

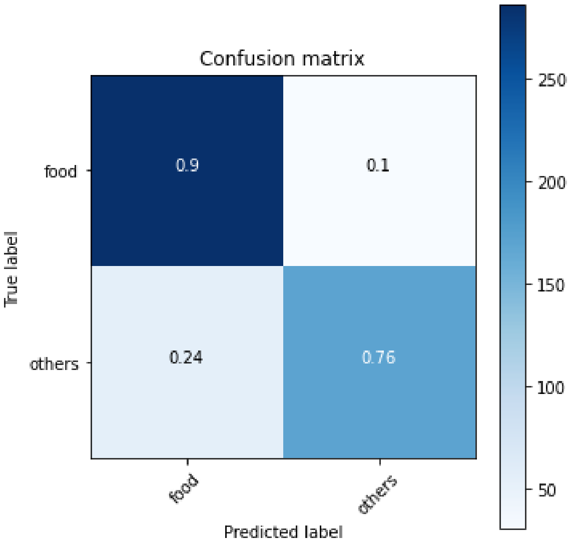

Figure 12: Confusion matrix of category extraction for food

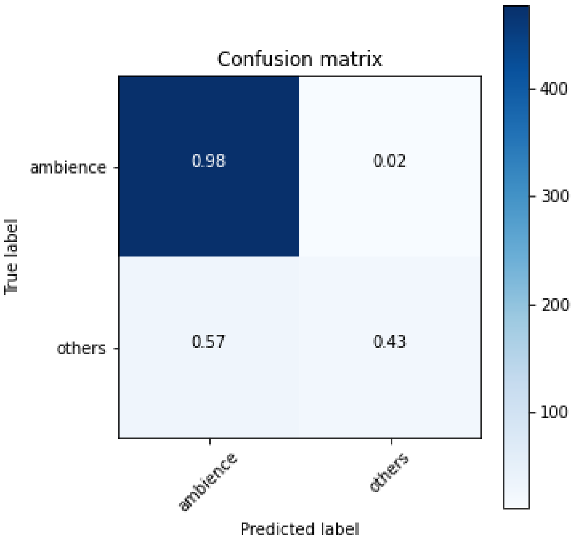

Figure 13: Confusion matrix of category extraction for ambience

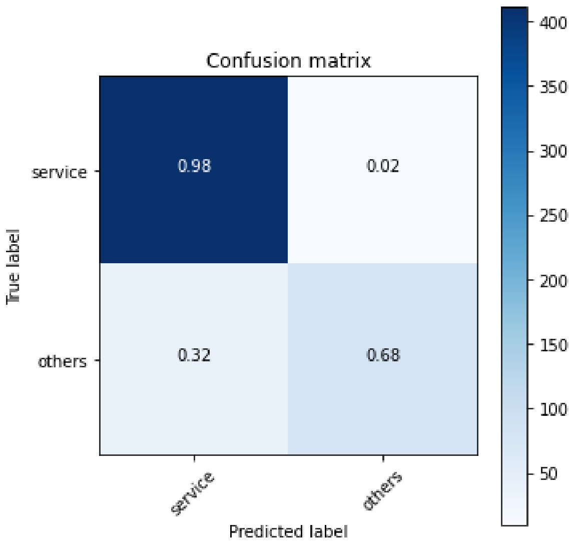

Figure 14: Confusion matrix of category extraction for service

Task	Epochs	Batch Size	Learning Rate	WeightDecay
ATE	5	16	2e-5	0.01
SP	5	16	2e-5	0.01
CE	10	64	2e-5	-1
SC	5	16	2e-5	0.01

Table 12: Hyperparameters

References

[1] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. Neural Computation, 9(8):1735–1780.

[2] Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805

[3] Zhiqiang Toh and Wenting Wang. 2014. DLIREC: aspect term extraction and term polarity classification system. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014).

[4] Maksim Tkachenko and Andrey Simanovsky. 2012.Named entity recognition: Exploring features

[5] Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmannand Michael Baeriswyl. Unsupervised Aspect Term Extraction with B-LSTM CRF using Automatically Labelled Datasets

[6] Victor Sanh and Lysandre Debut and Julien Chaumond and Thomas Wolf. 2020 DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108