Homework 2: Aspect-Based Sentiment Analysis

Leonardo Emili
Sapienza University of Rome, Italy
emili.1802989@studenti.uniroma1.it

1 Introduction

Aspect-Based Sentiment Analysis (ABSA) is a classification task that aims at identifying the aspects of a given target entity as well as the sentiment expressed toward each aspect. Moreover, in ABSA, we classify a sentence into a set of pre-identified categories with their polarities. Hence, we frame its formulation as a multi-step problem with the following phases: aspect terms extraction (A), aspect polarities identification (B), categories extraction (C), and category polarities identification (D), where task A+B identifies aspect polarities from the set of aspects extracted at step A, and similarly for task C+D. The main contributions of our work can be summarized as follows:

- Propose a multi-task transformer-based architecture to jointly learn all the subtasks.
- Draw a quantitative analysis of the results achieved by the experiments.
- Describe how to improve the drawbacks of proposed approaches with future work.

2 Related work

A large interest has been spent on ABSA in recent years, both from industry (e.g. opinion mining of consumer reviews) and academia (Wang et al., 2020; Jiang et al., 2019; Xue and Li, 2018). Neural networks, and more recently, transformer-based architectures, keep breaking state-of-the-art (SOTA) results in many Natural Language Processing (NLP) tasks. Along these lines, in (Tang et al., 2016) the authors propose a Target-Dependent LSTM architecture to encode the relatedness of a target word with its context words to infer the sentiment polarity towards the target term. In (Wang et al., 2016) the authors investigate the role played by attention in determining the part of a sentence relevant to a specific aspect term.

3 Dataset description

The dataset consists of English sentences with pre-identified entities that can be either laptops or restaurants. Each input instance is annotated with a collection of aspect terms of such entities and their polarities. It is worth noting that multi-word aspect terms should be treated as single aspect terms. Furthermore, only for the instances from the restaurant domain, we have a set of categories and their polarities.

4 Preprocessing

We extract lemmas and Part-Of-Speech (POS) tags using the NLTK library (Loper and Bird, 2002), and pre-compute WordPiece masks to get word-level representations [1]. Additionally, we apply data augmentation such that input sentences are duplicated by a factor of n, being n the number of aspect terms in the sentence, each time masking-out the other $n - 1$ terms to avoid data inconsistencies. Similar reasoning holds for aspect category terms. After this step, the size of the training set increases by more than 160%.

5 Contextualized Word Embeddings

In the past years, static word embeddings (Mikolov et al., 2013; Pennington et al., 2014) well served in many NLP tasks even though they were not suited for polysemous words (Hu et al., 2016). Thanks to the recent success of contextualized embeddings (Peters et al., 2018; Devlin et al., 2019), we are able to get context-specific representations of words depending on their surrounding terms. In this context, we employ BERT that conveniently uses WordPiece segmentation to avoid Out-Of-Vocabulary words.

[1] We define our pooling strategy, similar to AllenNLP's implementation (Gardner et al., 2018).

6 Conditional Random Field

Linear chain Conditional Random Fields (CRFs) (Lafferty et al., 2001) have proven their effectiveness in several sequence tagging tasks (Huang et al., 2015). As a structured learning model, the idea is to chain predictions Y such that the value of Y_i depends on Y_{i-1} and to condition them on a sequence of observations X.

7 Methodology

In this section, we formalize the tasks of aspect terms identification 7.1 and category identification 7.2. Furthermore, we propose a unified approach 7.3 that enables the model to learn the two tasks in parallel in a *multitask learning* (Caruana, 1996) fashion.

7.1 Aspect terms identification

Given an input sentence of tokens $x_1 x_2 \cdots x_n$, our objective goal is to determine the range of contiguous tokens $x_i \cdots x_{i+k}$ that belong to the same aspect term. For instance, in the sentence *The food is tasty and portion sizes are appropriate.* we recognize *food* and *portion sizes* as target aspect terms. We frame the problem as a Named Entity Recognition (NER) task, where aspect terms are annotated according to the Inside–outside–beginning (IOB/BIO) tagging scheme, and entity types (e.g. LOC, ORG) are dropped. Consequently, we model multi-word aspect terms using an initial *begin* symbol, followed by a collection of *inside* symbols. To extract aspect terms polarities, we can extend this reasoning and pair each symbol of the BIO tagging scheme to the desired polarity (Li et al., 2019). The *outside* tag does not change since it does not involve polarities. To extract the sentiment of a candidate aspect term, we apply an aggregation strategy over the predicted labels (e.g. first or most common polarity aspect-wise).

7.2 Category identification

The naïve approach is to infer the set of categories of a sentence by looking at the probabilities of each of them given the input sentence. The intuition is that if the word *food* occurs in a sentence, there will be a good chance of observing the homonymous class for that sentence. More generally, we expect a high probability for class x if the input tokens are related to the domain of x, and a low probability for class y if none of the input tokens are related to y. To identify polarities we apply the same reasoning as in 7.1, considering as target vocabulary the Cartesian product of the category class and the polarity class.

7.3 Unified approach

Thanks to the big success of multitask learning approaches (Conia et al., 2021; Raganato et al., 2017; Collobert and Weston, 2008), we propose a unified model that tries to solve both tasks in parallel. According to the hypothesis for which aspect terms can be informative for predicting aspect categories (Xue et al., 2017), we leverage both the input sentence and the recognized aspect terms for the extraction. We define a loss function for each subtask and optimize for them *jointly* to improve subtasks A+B and C+D.

8 Model architecture

In this section, we describe the architecture for the multitask approach. Solving only for task B, or D, can be achieved as a simplification of this model, where pre-identified aspect terms, or aspect categories, are fed as input, and we analyze their relatedness with respect to the input sentence.

8.1 Input representation

As the core representation of the input sequences, we stack the last four layers of BERT embeddings to have a context-aware representation of the tokens. Following (Alghanmi et al., 2020), we apply an average pooling layer and concatenate the resulting word-level BERT embeddings with static word embeddings (i.e. Word2Vec).

8.2 POS embeddings

To further enhance the quality of our input embeddings, we append a learnable k-dimensional POS embedding. We expect it to boost the performance since some parts of speech are less observed when annotating words as aspect terms 1. The value of k is experimentally found through hyperparameter tuning.

8.3 Sentence encoder

Leveraging the sequential nature of the input sentences, we employ a Bidirectional Long Short-Term Memory (BiLSTM) to extract latent representations of the input words. Their gated architecture enables to capture short-term dependencies while also retraining part of long-term dependencies. However, since hidden state h_i depends on the

computation of h_{i-1} for both directions, it heavily affects parallelism. To overcome this issue, we experiment with self-attention layers (Vaswani et al., 2017) and transformer encoders. Differently from Recurrent Neural Networks (RNNs), they attend to information from the whole input sequence rather than compressing it into single hidden vector states, with corresponding information loss (Bahdanau et al., 2016).

8.4 Task-specific decoders

At the end of the encoding pipeline, we train two linear decoders to map predictions to the corresponding output space. Their linear nature confirms the intuition that the subtasks should be informative to each other. The multitask model should benefit from it, learning a shared hidden representation.

9 Experimental setup

In this section, we define the settings and the tools (Biewald, 2020) used to evaluate the experiments.

9.1 Datasets

For the purpose of this project, we do not use any external training corpus. However, for some experiments, we find it beneficial to train our model on a subset of the available data. Experiments that show this behavior are reported with Φ_d, with $d \in \{laptop, restaurant\}$, referring to models trained only on that subset of data.

9.2 Evaluation metrics

As an evaluation framework, we consider multiple metrics in order to assess the quality of a particular model. Namely, in Table 1 we depict the highest F1-macro and F1-micro scores achieved by each architecture. However, for the testing phase, we pick the model with the highest F1-macro score.

9.3 Hyperparameters

In Table 3, we present a subset of the hyperparameters used in our models. However, we do not intend it to be an exhaustive list of all the hyperparameters and configurations used. They can be reached here.

9.4 Training details

In order to alleviate the memory requirements of the transformer-based experiments and speed up training, we use axial positional encodings (Kitaev et al., 2020; Ho et al., 2019) to reduce the number of parameters of the network, as well as APEX mixed-precision training and frozen word embeddings. We include weighted loss functions according to the distribution of polarity labels in the training set.

10 Experimental results

Table 1 depicts the ablation study derived from different experiments of the multitask learner. It is worth observing how the bidirectional LSTM model achieves the best performance when exploiting all the core features. In Figures 3 and 5, we can observe the relative confusion matrices. It confirms our intuition that recurrent neural models overcome the difficulties faced by simpler multi-layer perceptron architectures. Moreover, we can notice a general performance degradation when adding a self-attention layer on top of the recurrent layer. The CRF-BiLSTM model achieves similar results without significant performance gain. Please refer to Figure 4 for the complete architecture. Furthermore, training on a subset of the available data has proven beneficial only in the category identification task, with improvements up to 3%. Additionally, word-level tokens have proven useful, and removing them lead the scores down to 48.56% on task A+B and 52.92% on task C+D.

10.1 Cross-domain evaluation

We additionally try the following experiment masking out labels for aspect categories: first train model $\Phi_{restaurant}$ on restaurant data and evaluate on laptop data, and vice versa for Φ_{laptop}. The resulting macro F1 scores on task A+B are respectively 19.76% and 16.68%. We observe that both the models overfit the training data and lack generalization capabilities. It is worth noting how, without masking, the former improves up to 21.76%. It gives us a reason to believe that multitask learning is a good choice even in such transfer learning settings.

11 Conclusion and future work

We developed different strategies to tackle the ABSA pipeline and eventually discovered the validity of the multitask learning approach against individual learners. We observed through extensive experimentation that recurrent neural models paired with contextual embeddings lead to good performance. As a crucial drawback of the system, we believe that frozen word embeddings heavily limit the performance. Future work would include external training corpora, train with learnable word embeddings, and apply more sophisticated data augmentation techniques (Liesting et al., 2021).

References

Israa Alghanmi, Luis Espinosa-Anke, and Steven Schockaert. 2020. Combining bert with static word embeddings for categorizing social media. pages 28–33.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Rich Caruana. 1996. Algorithms and applications for multitask learning. In *In Proceedings of the Thirteenth International Conference on Machine Learning*, pages 87–95. Morgan Kaufmann.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Simone Conia, Andrea Bacciu, and Roberto Navigli. 2021. Unifying cross-lingual semantic role labeling with heterogeneous linguistic resources. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–351, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. 2019. Axial attention in multidimensional transformers.

Wenpeng Hu, Jiajun Zhang, and Nan Zheng. 2016. Different contexts lead to different word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 762–771.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. pages 6281–6286.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting bert for end-to-end aspect-based sentiment analysis.

Tomas Liesting, Flavius Frasincar, and Maria Mihaela Truşcă. 2021. Data augmentation in a hybrid approach for aspect-based sentiment analysis. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 828–835, New York, NY, USA. Association for Computing Machinery.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. Cite arxiv:1301.3781.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective lstms for target-dependent sentiment classification.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. pages 606–615.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks.

Wei Xue, Wubai Zhou, Tao Li, and Qing Wang. 2017. Mtna: A neural multi-task model for aspect category classification and aspect term extraction on restaurant reviews.

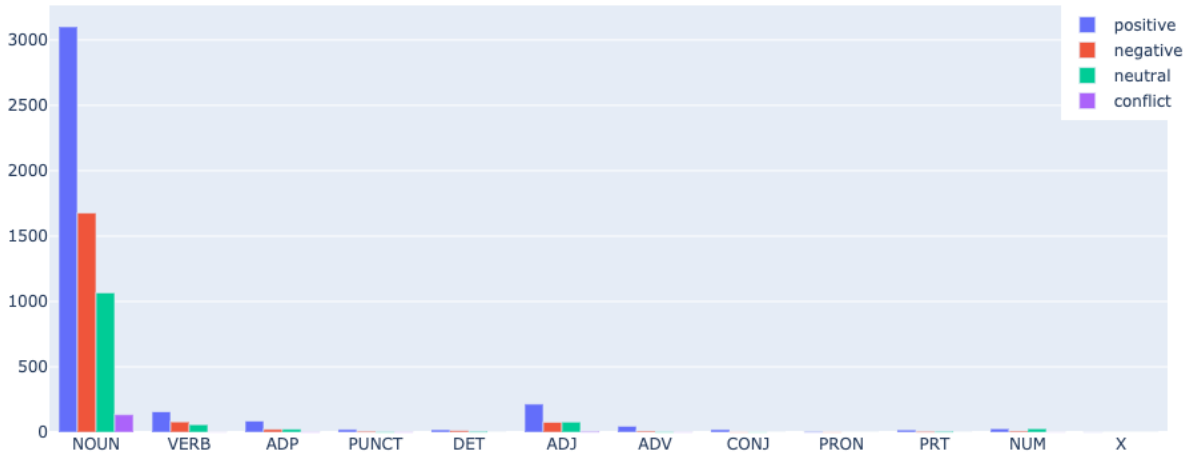

Figure 1: Distribution of POS tags over the tokens labeled as aspect terms.

Architecture	Core features	Aspects		Categories $^{(\Phi_{restaurant})}$	
		F_1^{macro}	F_1^{micro}	F_1^{macro}	F_1^{micro}
MLP	BERT	35.82	47.73	34.59	41.97
	+ Word2Vec	36.95	50.09	33.76	42.51
	+ POS	38.10	48.58	36.42	42.86
BiLSTM	BERT	49.54	59.91	50.69	62.34
	+ Word2Vec	49.20	61.50	51.63	64.50
	+ POS	**50.04**	**65.02**	**55.00**	**66.47**
BiLSTM (attention)	BERT	47.79	58.47	49.05	60.18
	+ Word2Vec	47.98	59.22	49.34	60.79
	+ POS	49.46	60.18	53.88	64.05
CRF-BiLSTM	BERT	47.57	57.17	48.95	59.48
	+ Word2Vec	48.11	60.10	48.03	59.22
	+ POS	49.85	62.93	51.13	62.88
Transformer (encoder)	BERT	35.92	48.19	49.44	61.07
	+ Word2Vec	36.79	50.10	49.93	61.86
	+ POS	37.33	50.53	51.09	63.01

Table 1: Experiments of the multitask model.

Model	Aspects		Categories $^{(\Phi_{restaurant})}$	
	F_1^{macro}	F_1^{micro}	F_1^{macro}	F_1^{micro}
Aspect classifier	41.25	60.16	-	-
Category classifier	-	-	38.23	49.12
Multistep classifier	**50.04**	**65.02**	**55.00**	**66.47**

Table 2: Performance of the multistep classifier (*multitask learning*) vs task-specific classifiers.

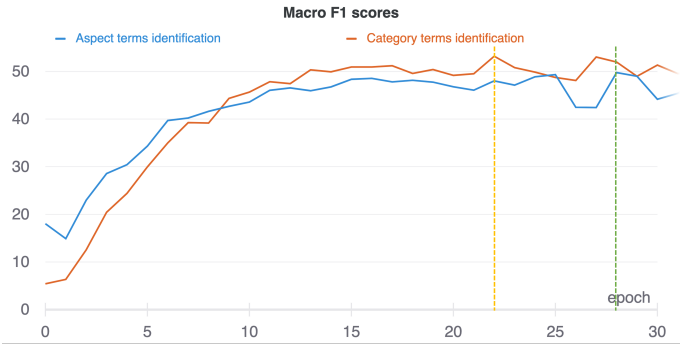

Figure 2: Diagram showing the macro F1 score over the epochs. Dashed lines denote the highest values for the category terms identification task and the aspect terms identification task, respectively yellow and green line.

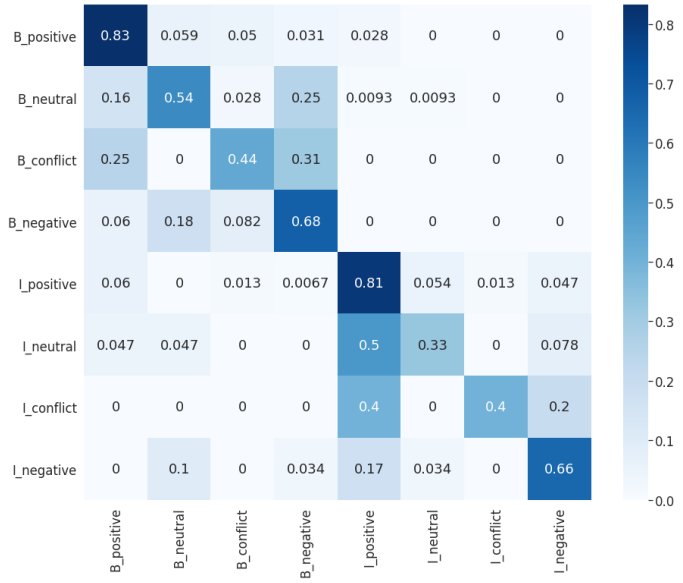

Figure 3: Normalized confusion matrix for the aspects identification task. It is worth observing that the model performs much better on positive aspect terms since the dataset is highly unbalanced towards the majority class 1.

Hyperparameter	Value
Learning rate	0.05
Optimizer	Adam
Loss function	Cross entropy
POS embedding size	30
Hidden size	512
Dropout	0.65
Epochs	50
Word encoder	bert-base-uncased

Table 3: List of hyperparameters.

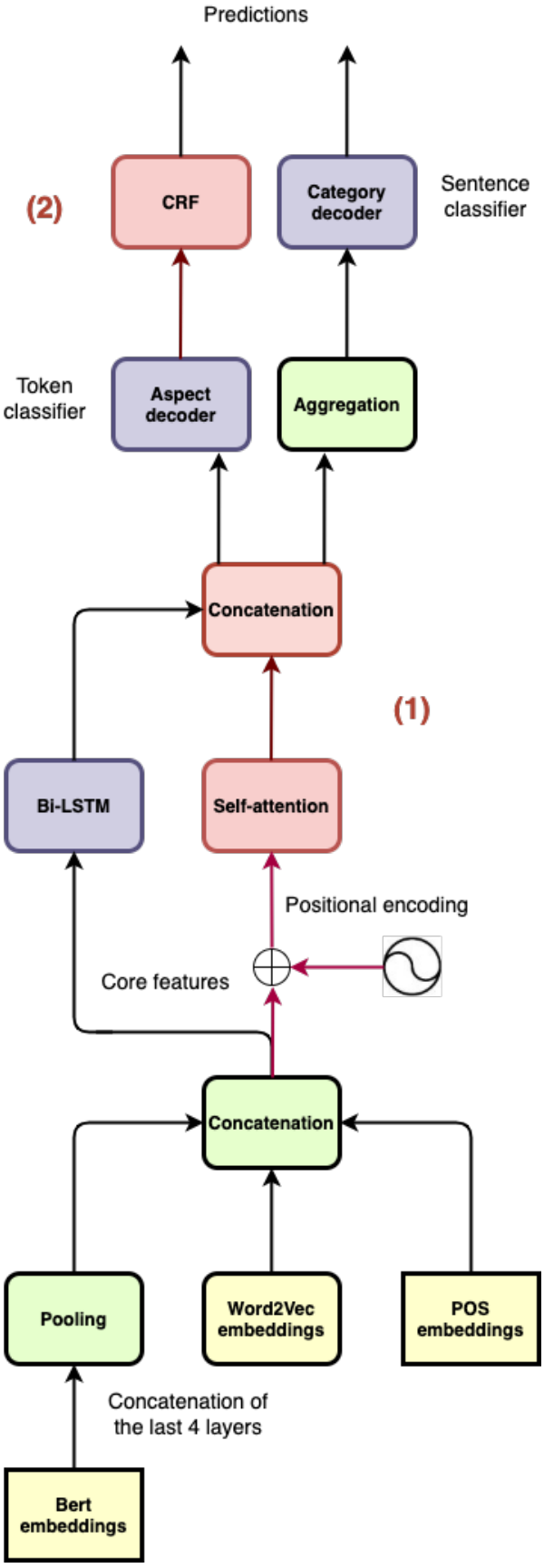

Figure 4: Diagram showing the architecture of the multi-step model for multitask learning. Differences from the base Bi-LSTM architecture are highlighted in red: *(1)* refers to the BiLSTM (attention) architecture with positional embeddings, *(2)* highlights the CRF layer used to decode emission scores using the Viterbi algorithm.

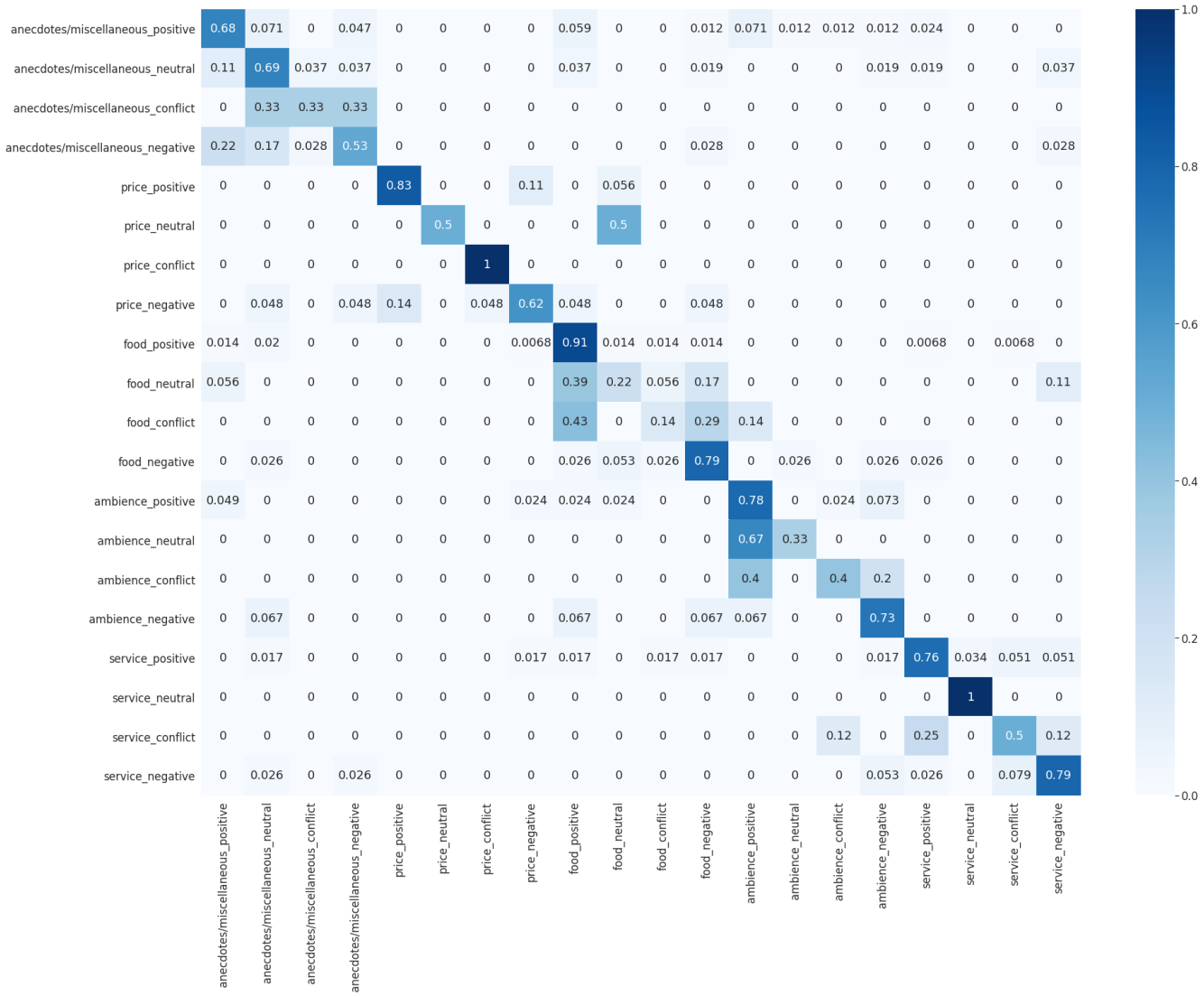

Figure 5: Normalized confusion matrix for the category identification task. Even in this confusion matrix, it is possible to see that the model overfits the majority polarity class. However, it is the case that for some minority classes we have good performance (e.g. price_conflict, service_neutral).