Aspect-Based Sentiment Analysis

Giuseppe Masi
Sapienza Univesity of Rome
Matricola 1962771
masi.1962771@studenti.uniroma1.it

1 Introduction

The goal of this homework is to identify the aspect terms of given target entities (in our case restaurants and laptops) and the sentiment expressed towards each aspect.

In particular, I address the following subtasks:
A. aspect term identification;
B. aspect term polarity classification;
C. aspect category identification;
D. aspect category polarity classification.

2 Pre-processing

Punctuation symbols are removed and the texts are lower-cased.

3 Task A

3.1 Dataset

Starting from the restaurants and the laptops datasets, for each text I associate a sequence of BIO tags with the following criteria: B tag corresponds to a word that is Beginning of an aspect term; I tag corresponds to a word that is Inner an aspect term; O tag corresponds to all the Other words of the text. An example is shown in Figure [1].

Moreover, I build a vocabulary with the words of the training data mapping words to integer and vice-versa. Words in the development datasets that are not in the vocabulary are mapped to a fixed index (representing an unknown token).

3.2 Model Architecture

The first layer of the model is the Embedding layer, that associate a vector (of embedding_dim dimension) to each word in the vocabulary.

Then there is an LSTM layer implementing a many-to-many LSTM neural network.

On top of it, there is a linear classifier layer to assign each token representations (that is the output of the LSTM) a BIO tag (i.e. the classifier has 3 output classes).

After the first two layers, DROPOUT [2] is applied.

The overall architecture is shown in Figure [2].

3.3 Training Time

Given a text, as ground truth, I use the BIO tags sequence of the whole text obtained from the gold target aspects.

The loss function used is the CROSS ENTROPY function.

3.4 Validation Time

At validation time, for each word in the input text I use the trained model to get the probabilities that the word belongs to each BIO tags. In order to get a prediction, I take the tag corresponding to the highest probability.

3.5 Experiments

I perform various experiments involving different hyper-parameters.

About the embedding layer, I try both to learn it during training (by random initialization and back-propagation) and to use a pre-trained FASTTEXT [1] word embeddings. When using it, the missing words are represented by a (fixed) random vector.

All the test performed are summarized in the table [1].

3.6 Results

The plots [3] and [4] compare respectively the trend of the loss on the training set and the macro $F1$-score on the validation set of the various models over the epochs. Table [2] summarizes the best performances of the models. The best model is that one of Test n.2 reaching 80.3% of macro F1-score. Table [3] reports the metrics of it and Figure [5] shows the confusion matrix.

4 Task B

4.1 Dataset

For each record in the provided datasets, I consider the text and the respective aspect term(s) one by one. In particular, given a text and an aspect term in it, I generate the following two texts for each sentiment (positive, negative, neutral, conflict): the first one is just the original text in which the aspect term is highlighted by surrounding it with '"'; the second one is an auxiliary sentence of the shape: `aspect_term:sentiment`. Each of these pairs is associated with the binary label *True/False* if the `sentiment` in the auxiliary sentence corresponds to the gold sentiment or not.

An example is shown in Figure [6].

4.2 Model Architecture

Given the pair of the previous section, I use the pre-trained uncased BERTTOKENIZER to obtain a sequence of tokens to give in input to the pre-trained uncased BERT$_{BASE}$, that I fine-tuned on the addressed task during training backpropagating the error on the BERT layer. It is composed of 12 Transformer blocks, the dimension of the hidden layers is 768, and 12 self-attention heads.

I consider the final hidden state of BERT corresponding to the first token `[CLS]` as the latent representation of the whole input sequence.

Then DROPOUT is applied and on top of the model there is a classifier (MLP) and a SIGMOID function is computed to obtain values between 0 and 1.

The overall architecture is shown in Figure [7].

4.3 Training Time

Given a *text-auxiliary sentence* pair, as ground truth I use the *True/False* label if the gold sentiment towards the target aspect term in the *text* is that one reported in the *auxiliary sentence*. So, for each pair, the model provides a probability that the above label is *True* or *False*.

The loss function used is the BINARY CROSS ENTROPY function.

4.4 Validation Time

At validation time, I use the trained model to get a prediction *True/False* for each pair in the dev set. So, given a *text* and an *aspect term*, I obtain a list of probability for each sentiment towards the target *aspect term* in the *text*. In order to get a prediction, I take the sentiment corresponding to the highest probability.

4.5 Experiments

The performed experiments involve different hyperparameters.

About the classifier on top of the model, I try two architectures: the first one presents two linear layers with a DROPOUT layer in between; the second one is just one linear layer.

All the test peformed are summarized in the table [4].

4.6 Results

The plots [8] and [9] compare respectively the trend of the loss on the training set and the macro $F1$-score on the validation set of the various models over the epochs. Table [5] summarizes the best performances of the models. The best model is that one of Test n.3 reaching 64.0% of macro F1-score. Table [6] reports the metrics of it and Figure [10] shows the confusion matrix.

5 Task A+B

I perform task A+B by stacking the best model B on top of the best model A, so by using the outputs of model A to input model B. I reach 46.0% of macro F1-score, 55.5% of micro F1-score, and table [7] reports all the measures.

As we can see, the hardest classes to predict correspond to the ones with fewer examples in the training datasets.

6 Task C

6.1 Dataset

For each record in the restaurants' dataset, I consider the text and the respective categories altogether. In fact, I represent the categories by using a binary $0/1$ vector of length equal to the number of different categories present in the dataset (in a fixed order). Given an instance of the dataset, I associate to it the above vector in which each entry indicates whether the corresponding category is involved or not. This also represents the ground truth that the model is meant to learn.

6.2 Model Architecture

As for Task B, there is the BERTTOKENIZER to obtain a sequence of tokens to give in input to the pre-trained uncased BERT$_{BASE}$. I consider the final hidden state of BERT corresponding to the first token [CLS] as the latent representation of the whole input sequence.

Then DROPOUT is applied and on top of the model there is a classifier (MLP) and a SIGMOID function. This time, the number of output nodes of the classifier is equal to the number of different categories.

6.3 Training Time

Given an input instance, as ground truth, I use the above binary vector representing the categories.

The loss function used is the BINARY CROSS ENTROPY function.

6.4 Validation Time

At validation time, I use the trained model to get a predicted vector of length equal to the number of different categories, in which each entry represent the probability that the corresponding category is involved in the text or not. In order to get a prediction, I round the entry of the vector obtaining $0/1$ values. Since each input text involves at least one category, I always take the category corresponding to the highest probability.

6.5 Experiments

The first experiment considers just the input text to input the model.

In the second experiment, for each instance, I generate a *text-auxiliary sentence* pair in which: the *text* is just the original text; the *auxiliary sentence* is the comma-separated list of the aspect terms included in the *text*, obtained recalling the model A. An example is shown in Figure [11].

The test peformed are summarized in the table [8].

6.6 Results

The plots [12] and [13] compare respectively the trend of the loss on the training set and the macro $F1$-score on the validation set of the various models over the epochs. Table [9] summarizes the best performances of the models. The best model is that one of Test n.1 reaching 86.2% of macro F1-score. Table [10] reports the metrics of it and Figure [14] shows the confusion matrix.

7 Task D

7.1 Dataset

For each record in the provided datasets, I consider the text and the respective category(ies) one by one. In particular, given a text and a category, I generate the following two texts: the first one is just the original text; the second one is an auxiliary sentence reporting the category whose sentiment we want to predict.

I represent the sentiments by using a binary vector in which the (only) value 1 corresponds to the gold sentiment towards the category. So each *text-category* pair is associated to the vector of the sentiments towards the *category*.

7.2 Model Architecture

The model architecture is the same as in Task C, but the number of output nodes of the classifier is equal to the number of different sentiments.

7.3 Training Time

Given an input instance, as ground truth, I use the above binary vector representing the sentiments.

The loss function used is the BINARY CROSS ENTROPY function.

7.4 Validation Time

At validation time, I use the trained model to get a predicted vector of length equal to the number of different sentiments, in which each entry represent the probability that the corresponding sentiment is expressed towards the relative category. In order to get a prediction, I take the sentiment corresponding to the highest probability.

7.5 Experiments

The first experiment considers just the pair *text-category* to input the model.

In the second experiment, for each instance, the auxiliary sentence is augmented with a comma-separated list of the aspect terms and relative sentiments, obtained recalling model A and model B. An example is shown in Figure [15].

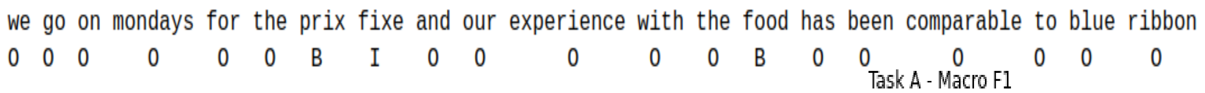
The test peformed are summarized in the table [11].

7.6 Results

The plots [16] and [17] compare respectively the trend of the loss on the training set and the macro $F1$-score on the validation set of the various models over the epochs. Table [12] summarizes the best performances of the models. The best model is that one of Test n.1 reaching 63.1% of macro F1-score. Table [13] reports the metrics of it.

8 Figures

8.1 Task A



```
we go on mondays for the prix fixe and our experience with the food has been comparable to blue ribbon
O  O  O     O     O   O   B    I   O   O      O       O    O   B    O   O      O        O   O     O
```

Figure 1: Example of generating BIO tags starting from a text and the target aspects.

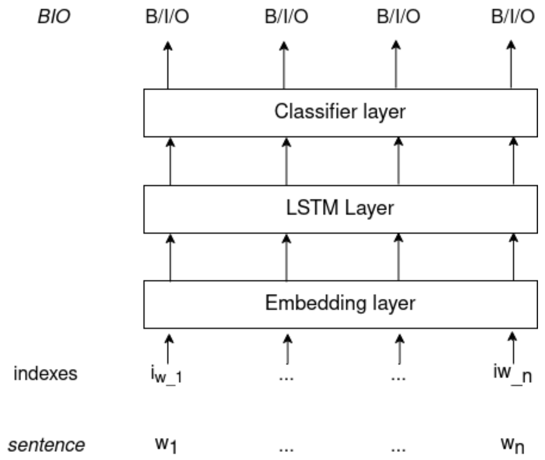


Figure 2: Model A overall architecture.

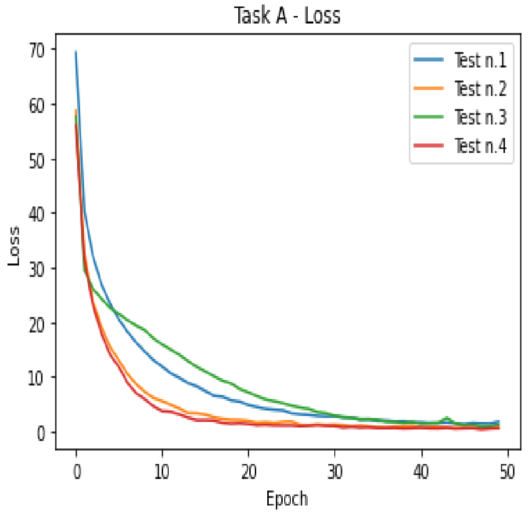


Figure 3: The trend of the loss on the training set (both restaurants and laptops) of the models A over the epochs.

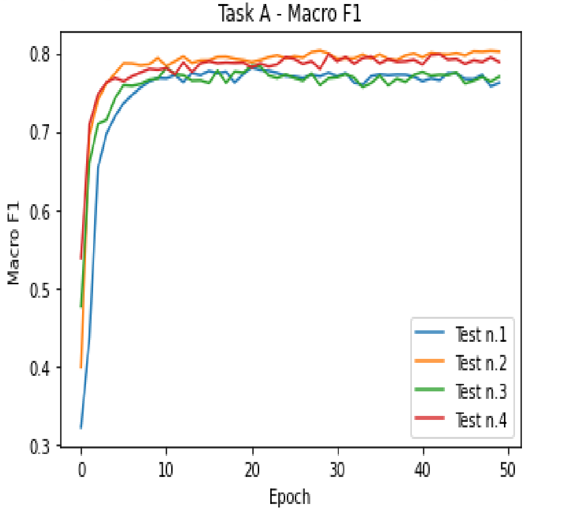


Figure 4: The trend of the macro $F1$-score on the development set (both restaurants and laptops) of the models A over the epochs.

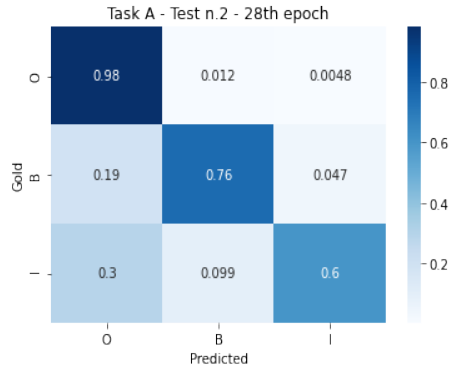
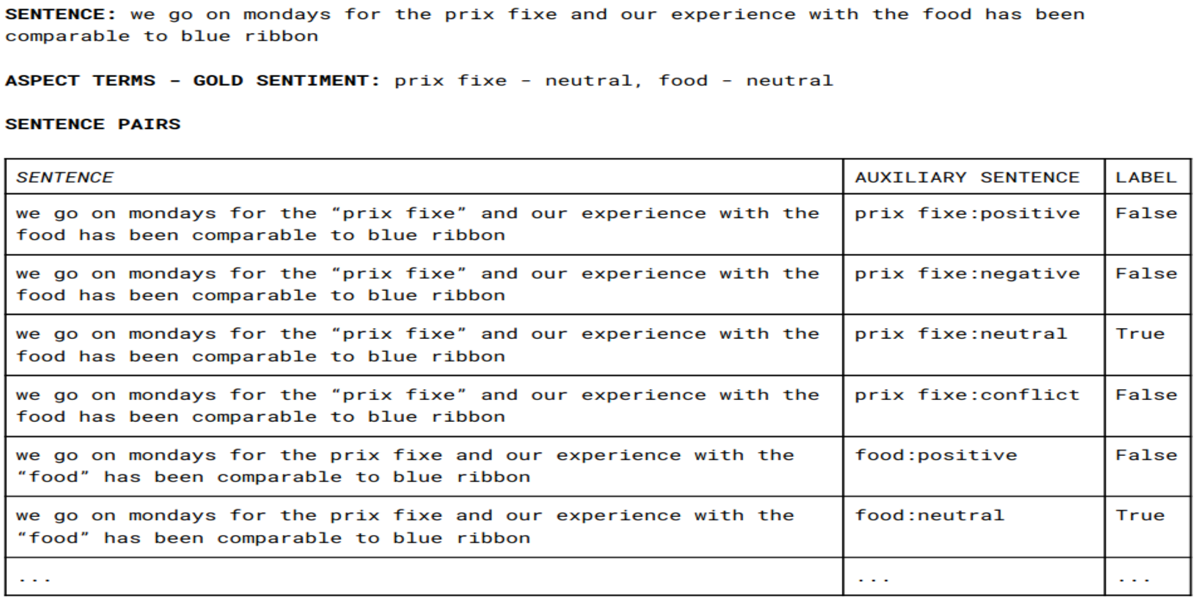


Figure 5: The normalized confusion matrix for the Task A of the best model.

8.2 Task B



SENTENCE: we go on mondays for the prix fixe and our experience with the food has been comparable to blue ribbon

ASPECT TERMS - GOLD SENTIMENT: prix fixe - neutral, food - neutral

SENTENCE PAIRS

SENTENCE	AUXILIARY SENTENCE	LABEL
we go on mondays for the “prix fixe” and our experience with the food has been comparable to blue ribbon	prix fixe:positive	False
we go on mondays for the “prix fixe” and our experience with the food has been comparable to blue ribbon	prix fixe:negative	False
we go on mondays for the “prix fixe” and our experience with the food has been comparable to blue ribbon	prix fixe:neutral	True
we go on mondays for the “prix fixe” and our experience with the food has been comparable to blue ribbon	prix fixe:conflict	False
we go on mondays for the prix fixe and our experience with the “food” has been comparable to blue ribbon	food:positive	False
we go on mondays for the prix fixe and our experience with the “food” has been comparable to blue ribbon	food:neutral	True
...	...	...

Figure 6: Example of generating text-aspect_sentiment pairs starting from a text, the target aspects and the respectively sentiments.

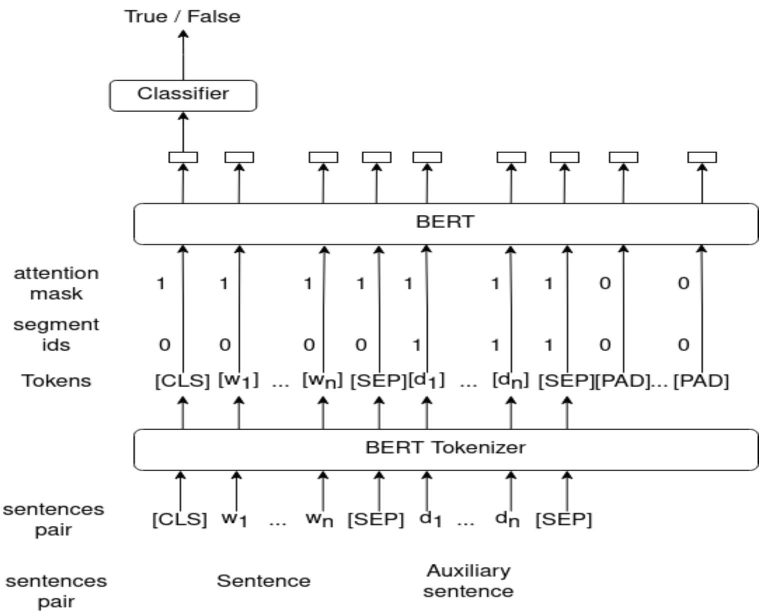


Figure 7: Model B overall architecture.

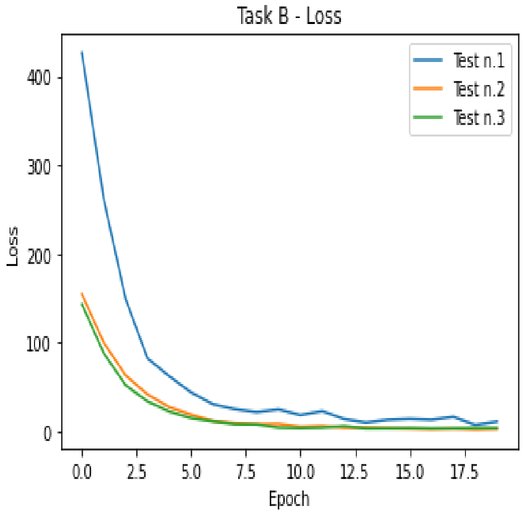


Figure 8: The trend of the loss on the training set (both restaurants and laptops) of the models B over the epochs.

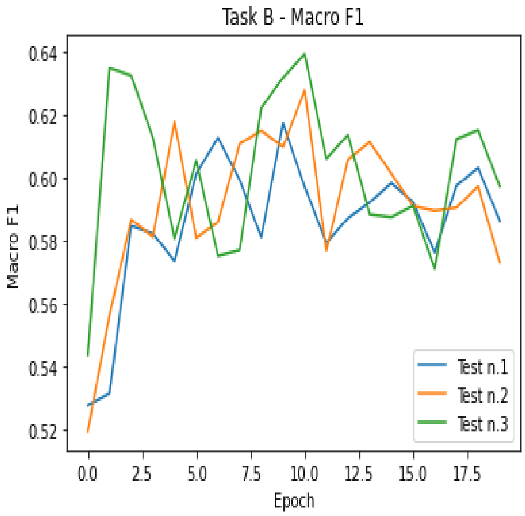


Figure 9: The trend of the macro $F1$-score on the development set (both restaurants and laptops) of the models B over the epochs.

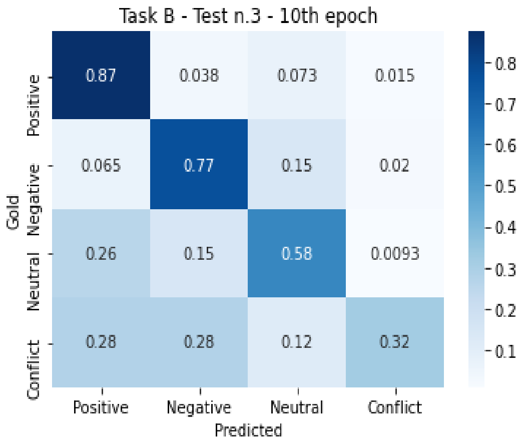
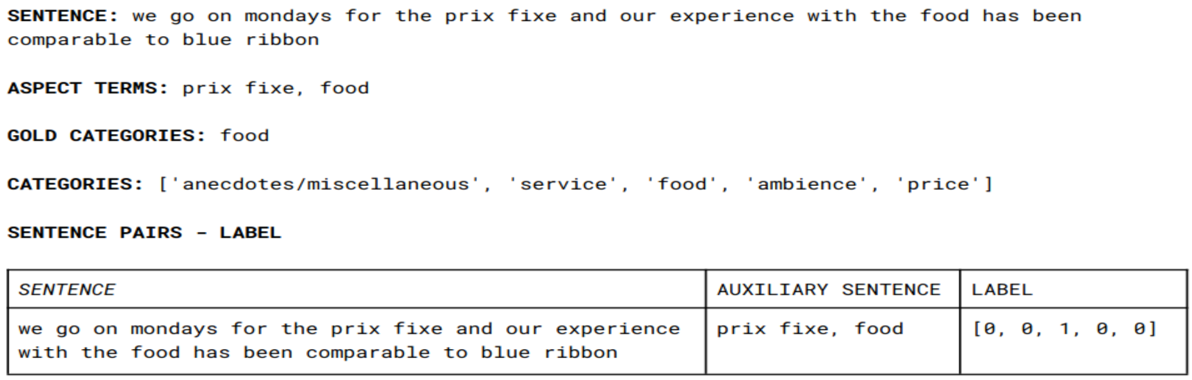


Figure 10: The normalized confusion matrix for the Task B of the best model.

8.3 Task C



SENTENCE: we go on mondays for the prix fixe and our experience with the food has been comparable to blue ribbon

ASPECT TERMS: prix fixe, food

GOLD CATEGORIES: food

CATEGORIES: ['anecdotes/miscellaneous', 'service', 'food', 'ambience', 'price']

SENTENCE PAIRS - LABEL

SENTENCE	AUXILIARY SENTENCE	LABEL
we go on mondays for the prix fixe and our experience with the food has been comparable to blue ribbon	prix fixe, food	[0, 0, 1, 0, 0]

Figure 11: Example of generating text-aspects terms pairs starting from a text and the target aspects.

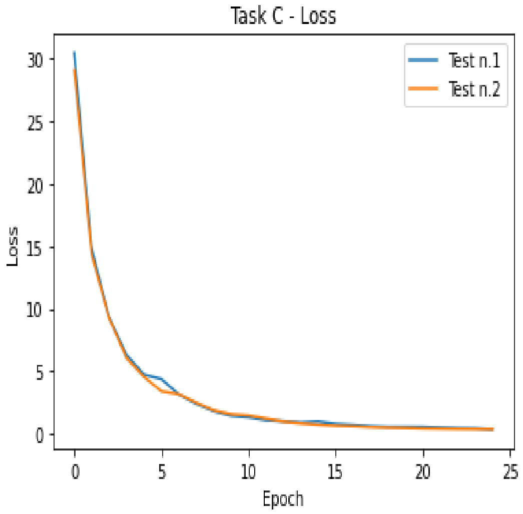


Figure 12: The trend of the loss on the training set (only restaurants) of the models C over the epochs.

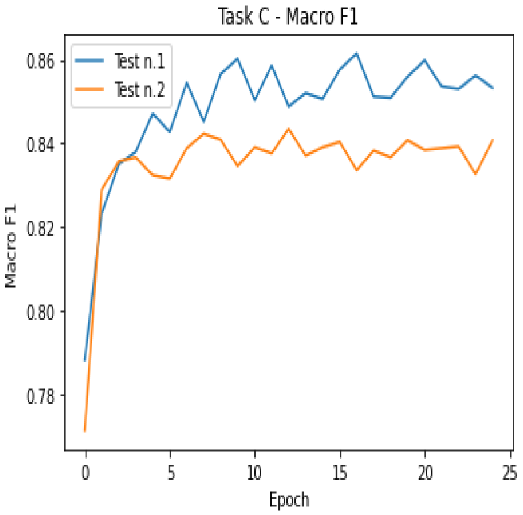


Figure 13: The trend of the macro $F1$-score on the development set (only restaurants) of the models C over the epochs.

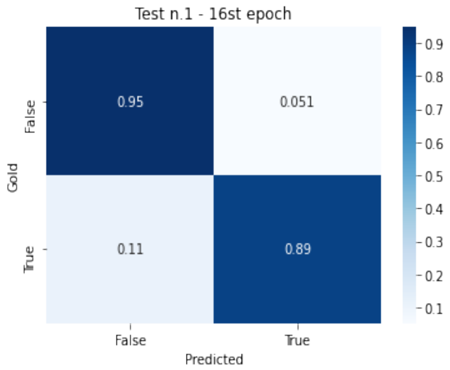
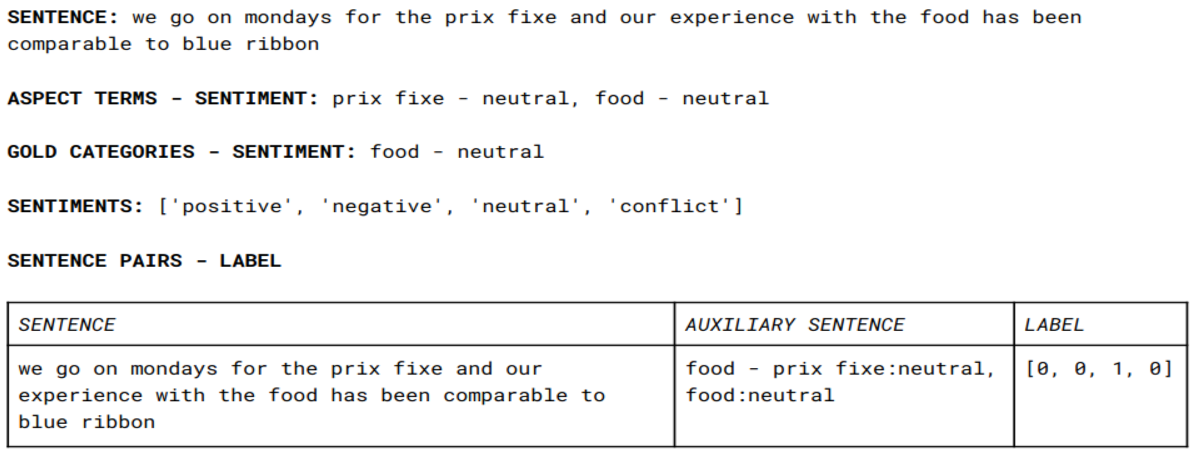


Figure 14: The normalized confusion matrix for the Task C of the best model.

8.4 Task D



SENTENCE: we go on mondays for the prix fixe and our experience with the food has been comparable to blue ribbon

ASPECT TERMS - SENTIMENT: prix fixe - neutral, food - neutral

GOLD CATEGORIES - SENTIMENT: food - neutral

SENTIMENTS: ['positive', 'negative', 'neutral', 'conflict']

SENTENCE PAIRS - LABEL

SENTENCE	*AUXILIARY SENTENCE*	*LABEL*
we go on mondays for the prix fixe and our experience with the food has been comparable to blue ribbon	food - prix fixe:neutral, food:neutral	[0, 0, 1, 0]

Figure 15: Example of generating the pairs for the model D starting from a text, the aspects with relative sentiments and the category.

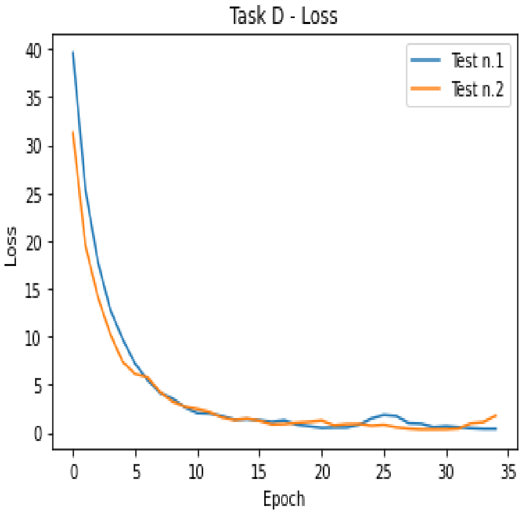


Figure 16: The trend of the loss on the training set (only restaurants) of the models D over the epochs.

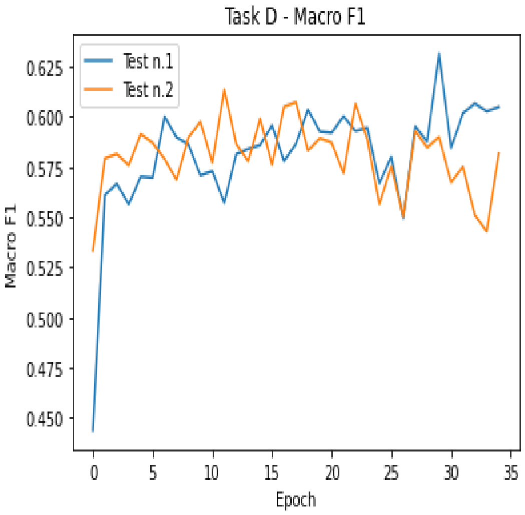


Figure 17: The trend of the macro $F1$-score on the development set (only restaurants) of the models D over the epochs.

9 Tables

9.1 Task A

	Batch	**FastText**	**Emb.**	**H. dim**	**Bidir**	**N. l.**	**Clas.**	**Dropout**	**Optim.**	**lr**
Test n.1	32	$False$	100	128	$False$	1	128	0.2	$Adam$	$1e-3$
Test n.2	32	$False$	100	128	$True$	2	256	0.2	$Adam$	$1e-3$
Test n.3	32	$True$	300	128	$True$	2	256	0.2	$Adam$	$1e-3$
Test n.4	32	$False$	100	192	$True$	2	384	0.2	$Adam$	$1e-3$

Table 1: Tests for Task A. "Emb." stands for embedding dimension. "H. dim" stands for the hidden dimension of the LSTM layer. "Bidir" indicates whether the LSTM is bidirectional or not. "N. l." indicates the number of hidden layers of the LSTM. "Clas." indicates the number of nodes of the linear classifier layer. "Dropout" indicates the dropout probability. "Optim." stands for optimizer. "lr" is the learning rate.

Model	**Macro F1**	**Epoch**
Test n.1	78.1%	20^{th}
Test n.2	**80.3%**	28^{th}
Test n.3	78.4%	20^{th}
Test n.4	79.9%	29^{th}

Table 2: Best performances of the models on the validation set (both restaurants and laptops) for task A.

Macro F1-score		80.3%
Macro precision		82.8%
Macro recall		77.9%
Per class precision	B	78.8%
Per class precision	I	72.3%
Per class precision	O	97.4%

Table 3: Metrics of the best model on the validation set (both restaurants and laptops) for task A.

9.2 Task B

	Batch	BERT Dropout	N. layer c.	N. nodes c.	Dropout	Optim.	lr
Test n.1	16	0.1	2	$(768, 384)$	0.2	$AdamW$	$2e-5$
Test n.2	48	0.1	2	$(768, 384)$	0.2	$AdamW$	$2e-5$
Test n.3	48	0.1	1	768	0.2	$AdamW$	$2e-5$

Table 4: Tests for Task B. ”BERT Dropout” indicates the dropout probability of the BERT layer (left as default). ”N. layer c.” indicates the number of linear layer of the classifier on top of the model. ”N. nodes c.” indicates the number of nodes of each linear layer of the classifier. ”Dropout” indicates the dropout probability. ”Optim.” stands for optimizer. ”lr” is the learning rate.

Model	Macro F1	Epoch
Test n.1	61.7%	9^{th}
Test n.2	62.8%	10^{th}
Test n.3	**64.0%**	10^{th}

Table 5: Best performances of the models on the validation set (both restaurants and laptops) for task B.

Macro F1-score		64.0%
Macro precision		64.3%
Macro recall		63.6%
Per class precision	Positive	85.2%
Per class precision	Negative	79.7%
Per class precision	Neutral	58.9%
Per class precision	Conflict	33.3%

Table 6: Metrics of the best model on the validation set (both restaurants and laptops) for task B.

9.3 Task A+B

Macro F1-score		46.0%
Macro precision		44.5%
Macro recall		47.8%
Micro F1-score		55.6%
Micro precision		54.7%
Micro recall		56.5%
Per-class F1	**positive**	63.7%
Per-class F1	**negative**	56.0%
Per-class F1	**neutral**	37.0%
Per-class F1	**conflict**	27.3%

Table 7: Metrics of the best model on the validation set (both restaurants and laptops) for task A+B.

9.4 Task C

	AS	Batch	BERT Dropout	N. layer c.	N. nodes c.	Dropout	Optim.	lr
Test n.1	$False$	32	0.1	1	768	0.2	$AdamW$	$2e-5$
Test n.2	$True$	32	0.1	2	768	0.2	$AdamW$	$2e-5$

Table 8: Tests for Task C. ”AS” indicates whether the auxiliary sentences with the aspect terms are used or not. ”BERT Dropout” indicates the dropout probability of the BERT layer (left as default). ”N. layer c.” indicates the number of linear layer of the classifier on top of the model. ”N. nodes c.” indicates the number of nodes of each linear layer of the classifier. ”Dropout” indicates the dropout probability. ”Optim.” stands for optimizer. ”lr” is the learning rate.

Model	Macro F1	Epoch
Test n.1	**86.2%**	16^{th}
Test n.2	84.4%	12^{th}

Table 9: Best performances of the models on the validation set (only restaurants) for task C.

Macro F1-score		86.2%
Macro precision		84.3%
Macro recall		88.1%
Micro F1-score		86.8%
Micro precision		84.9%
Micro recall		88.8%
Per-class F1	**ambience**	83.5%
Per-class F1	**anecdotes/miscellaneous**	82.6%
Per-class F1	**food**	91.6%
Per-class F1	**price**	86.2%
Per-class F1	**service**	86.8%

Table 10: Metrics of the best model on the validation set (only restaurants) for task C.

9.5 Task D

	AS	**Batch**	**BERT Dropout**	**N. layer c.**	**N. nodes c.**	**Dropout**	**Optim.**	**lr**
Test n.1	$False$	32	0.1	1	768	0.2	$AdamW$	$2e-5$
Test n.2	$True$	32	0.1	2	768	0.2	$AdamW$	$2e-5$

Table 11: Tests for Task D. ”AS” indicates whether the auxiliary sentences with the aspect terms are used or not. ”BERT Dropout” indicates the dropout probability of the BERT layer (left as default). ”N. layer c.” indicates the number of linear layer of the classifier on top of the model. ”N. nodes c.” indicates the number of nodes of each linear layer of the classifier. ”Dropout” indicates the dropout probability. ”Optim.” stands for optimizer. ”lr” is the learning rate.

Model	**Macro F1**	**Epoch**
Test n.1	**63.1%**	29^{th}
Test n.2	61.2%	11^{th}

Table 12: Best performances of the models on the validation set (only restaurants) for task D.

Macro F1-score		63.1%
Macro precision		61.0%
Macro recall		65.5%
Micro F1-score		72.2%
Micro precision		70.6%
Micro recall		73.9%
Per-class F1	**positive**	79.8%
Per-class F1	**negative**	66.7%
Per-class F1	**neutral**	61.3%
Per-class F1	**conflict**	44.8%

Table 13: Metrics of the best model on the validation set (only restaurants) for task D.

References

[1] Piotr Bojanowski et al. “Enriching Word Vectors with Subword Information”. In: *arXiv preprint arXiv:1607.04606* (2016).

[2] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.