Department of Informatics
Technical University of Munich

Sentiment Analysis Using BERT and Multi-Instance Learning

Burak Iryna[1], **Restrepo Pablo**[2]
[1]Department of Mathematics, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany
[2]Department of Informatics, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany
✉ iryna.burak@tum.de, pablo.restrepo@tum.de
31 July 2020

Abstract — We consider the task of sentence-level sentiment analysis for data coming from the organic reviews domain. In this project, we used Multi-Instance Learning Networks combined with various initial sentiment embeddings to explore both monolingual and cross-lingual settings. Additionally, we conducted experiments on the simplified task by dropping one of the classes.

1 Introduction

Today, the long-term goal of developing sustainable food systems is considered a high priority by several intergovernmental organizations [Mie et al., 2017]. This has lead to developments in modern organic food production, largely driven by the ideal of sustainability and environmental concern [Ditlevsen et al., 2019].
In order to create successful strategies to promote production and consumption of organic food, it is necessary to understand the opinions of the general public regarding organic products. For this, it is useful to understand the sentiment of comments about organic food in a sentence level. This is illustrated in Figure 1 showing that even though the overall sentiment of the comment is positive, there are sentences that are clearly negative. This would be ignored if we analyzed only the sentiment of the comment, and not the sentiment of each sentence.
In addition to the above, Web 2.0 has led to the emergence of blogs, forums, and online social networks that enable users to discuss any topic and share their opinions about it [Dang et al., 2020]. For this reason, coarse-grained document-level annotations are relatively easy to obtain. Despite this, the acquisition of sentence- or phrase-level sentiment labels remains a laborious and expensive endeavor even tough they are relevant to various opinion mining applications [Angelidis and Lapata, 2017], including sentiment analysis.
For all these reasons, a model that allows us to understand sentiment in a sentence level for the organic food domain based on comments is important.
In this paper, we will explore the use of Multi-Instance Learning Networks, which only require document-level supervision and learns to introspectively judge the sentiment of constituent segments [Angelidis and Lapata, 2017] for sentence-level sentiment analysis in the context of organic food. For this, we will use use data from Amazon reviews in English and German, together with a dataset that contains annotated sentences from Quora about organic products.
Also, we will explore the effects of using transfer learning together with Multi-Instance Learning Networks and compare the results obtained by using different initial sentence embeddings.
Finally, we will investigate the use of Multi-Instance Learning Networks in cross-lingual tasks for English and German data.

Rating: 4 stars
I like this product, because it is organic and healthy. It is also very tasty. The only thing I don't like is that they make it expensive, only because it is organic!. In any case, it is worth the money. I would buy it again.

- Overall sentiment of the comment: positive
- But there are still some comments with negative sentiment.

Figure (1) Example of a positive comment with negative sentences.

2 Theory

2.1 MilNet architecture

Multi-Instance Learning Networks were introduced in [Angelidis and Lapata, 2017] as a tool for predicting segment sentiment labels when having only document sentiment labels as a ground truth. In our setting, sentences are treated as segments, and comments — as documents.

The process of training a MilNet can be described with the following steps (see Figure 2):

(a) Split each document in the dataset into segments;

(b) Compute segment embeddings as a convolution of corresponding word embeddings;

(c) Compute segment labels using a recurrent neural network with gated recurrent units;

(d) Compute document label using attention mechanism;

(e) Do backpropagation with respect to ground truth of **document** labels only.

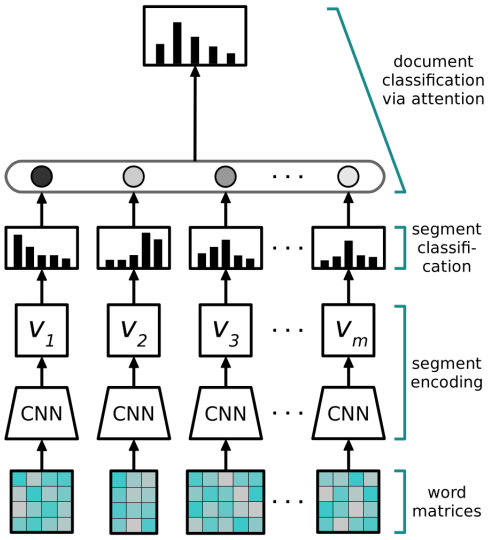


Figure (2) MilNet architecture.

After the training, one can extract segment sentiments by ignoring the last two steps.
For this project, we were interested in exploring the potential of BERT embeddings in such architecture. For this, we skipped the first two steps by using precomputed segment embeddings.

2.2 Embeddings

2.2.1 Vector semantics embeddings

Words that occur in similar contexts tend to have similar meanings. The concept of vector semantics instantiates this linguistic hypothesis by learning representations of the meaning of words, called **embeddings**, directly from their distributions in texts. These representations are used in every natural language processing application that makes use of meaning [Jurafsky and Martin, 2019].

2.2.2 Feature representation for neural networks

We can think of a feed-forward neural network as a function $NN(x)$ that takes as input a d_{in} dimensional vector x and produces a d_{out} dimensional output vector. When dealing with natural language, the input x encodes features such as words, part-of-speech tags or other linguistic information. Perhaps the biggest conceptual jump when moving from sparse-input linear models to neural-network based models is to stop representing each feature as a unique dimension (the so called one-hot representation) and representing them instead as dense vectors called **embeddings**. That is, each core feature is embedded into a d dimensional space, and represented as a vector in that space [Goldberg, 2015].

2.2.3 Contextualized Embeddings

With traditional embeddings such as **GloVe** and **Word2Vec**, each word in a vocabulary gets its own representation, independent of the context on which they are used. Even though this may be useful for many problems, it is logical to think that the semantic information of a word depends on the context in which it is used. For example, consider the word "jaguar", which can have different word senses depending on the context: an animal, a car or a guitar. Even when using the same word sense of a word, there may be semantic differences dependent on the context in which the word is used. Contextual embeddings like **ELMo** and **BERT** address this issue by giving each token of a sentence its own embedding depending on the entire context of the sentence.

2.2.4 Sentence Embeddings

Sentence embeddings are dense vectors that summarize different properties of a sentence (e.g. its meaning), thereby extending the very popular concept of word embeddings. In contrast to task-specific representations, such as the ones trained specifically for tasks like textual entailment or sentiment, such sentence embeddings are trained in a task-agnostic manner on large datasets. As a consequence, they often perform better when little labeled data is available [Rücklé et al., 2018].

2.2.5 Transfer learning and model pretraining

Language model pretraining has been shown to be effective for improving many natural language processing tasks [Devlin et al., 2018]. This model pretraining consists in training a neural network in one task (usually language modeling), and then using the intermediate representation of a deep layer of the network as embeddings for another model. Such approach allows to transfer the syntactic and semantic properties learned by the language modeling model to other models that focus on different tasks.
There are two existing strategies for applying pretrained language representations to downstream tasks: feature-based and fine-tuning. The feature-based approach, such as **ELMo**, uses task-specific architectures that include the pretrained representations as additional features. The fine-tuning approach, such as the Generative pretrained Transformer (**OpenAIGPT**), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning all pretrained parameters.

2.2.6 BERT: Bidirectional Encoder Representations from Transformers

BERT improves the fine-tuning based approaches by using a “masked language model” (MLM) pretraining objective. The masked language model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context [Devlin et al., 2018].

2.2.7 RoBERTa

RoBERTa is an improved recipe for training BERT models, that can match or exceed the performance of all of the post-BERT methods. It includes modifications to BERT such as: (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data [Liu et al., 2019].

2.2.8 XLING

XLING generalizes the concept of average word embeddings to power mean word embeddings. In this method, instead of using the component-wise arithmetic averages of the word embeddings to generate sentence embeddings, the concateniation of the so-called power means is used [Rücklé et al., 2018].

3 Experimental Setup

3.1 Data

For our experiments, we used three datasets. We will be referring to them as: **amazon EN**, **organic**, and **amazon DE**.

3.1.1 Amazon EN

This dataset contains Amazon product reviews and metadata from May 1996 to July 2014. In our case, since we were interested in the organic food domain, we used data from three categories:

- Grocery and Gourmet Food
- Health and Personal Care
- Beauty

These three categories contained the following amount of reviews:

Category	Reviews
Grocery and Gourmet Food	198.502
Health and Personal Care	151.254
Beauty	346.355
Total	696.111

Additionally, we filtered all reviews that contained the word "organic". In the end, we ended up with the following amount of reviews in each category:

Category	Reviews
Grocery and Gourmet Food	9.962
Health and Personal Care	3.272
Beauty	2.227
Total	15.561

The percentage distribution of ratings in the dataset is shown in Figure 3.

We can observe that the dataset is imbalanced. For this reason, we applied a downsampling strategy to balance it for our experiments.
To train the MilNets for our experiments, we were interested in getting the individual sentences of each

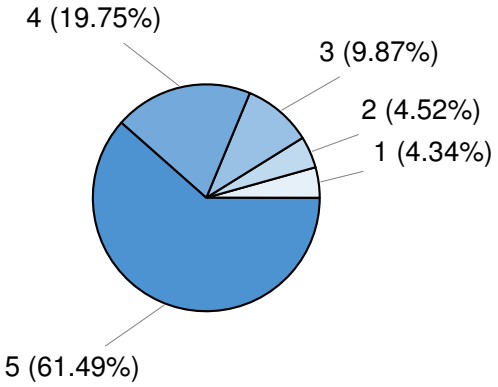


Figure (3) Percentage distribution of ratings in **amazon EN**.

comment. To split the Amazon comments into sentences, we used NLTK, which is a a suite of program modules, covering symbolic and statistical natural language processing [Loper and Bird, 2002].
After splitting our comments into sentences, we ended up with **129.867** sentences. Figure 4 shows the number of sentences per comment in the dataset.

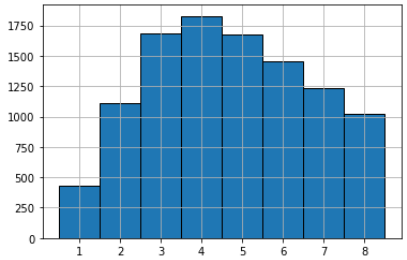


Figure (4) Number of sentences per comment in **amazon EN**.

In this histogram we can see that most of the comments contain between 3 and 5 sentences, which shows that the dataset is appropriate for our MilNet Experiments.
In many of our experiments, we used **BERT** to generate embeddings for our sentences. As **BERT** can only handle sentences with a maximum of 512 tokens including [CLS] and [SEP], we removed the outlier sentences that had a large number of tokens. For **amazon EN**, only 0.2% of the sentences contain more than 100 tokens. For this reason, we removed all sentences that surpassed that limit.
Figure 5 shows the number of tokens per sentence of the dataset after removing the outlier sentences. In this figure we can see that most of the sentences have around 20 tokens.

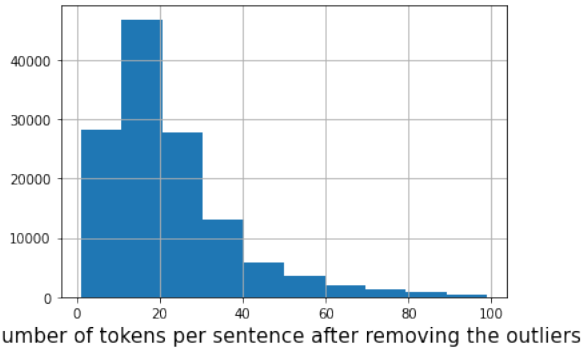


Figure (5) Number of tokens per sentence in **amazon EN**.

3.1.2 Organic

This dataset contains sentences with opinions about organic products from Quora. The sentiment of this sentences has been annotated by domain experts with the categories: positive, negative and neutral. The dataset is divided in train, validation and test data.
After removing all the 'nan' values, we ended up with the following amount of sentences, with the following distributions:

Train dataset

Sentiment	Sentences	%
positive	1500	32%
negative	1375	29.33%
neutral	1812	38.66
Total	4687	

We can observe that the data is only slightly unbalanced. Therefore, no balancing strategy was applied to this dataset for our experiments.
In this dataset, less than 1% of the data has more than 100 tokens. For this reason, we removed those outliers, just as we did with **amazon EN**. Figure 6 shows the number of tokens per sentence of the dataset after removing the outlier sentences. In this figure we can observe that the final distribution is similar to the one for **amazon EN**.

3.1.3 Amazon DE

This dataset contains Amazon reviews in the German language. Since our domain is organic products, we used the data from the two categories:

- Grocery

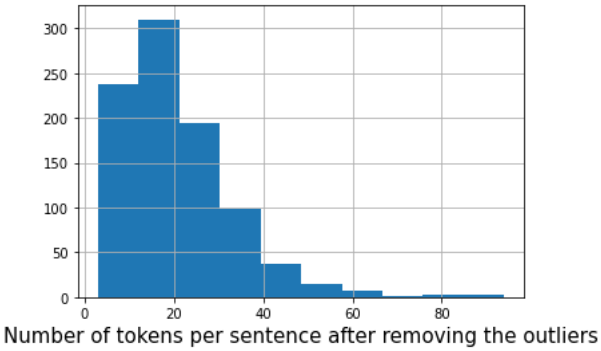


Figure (6) Number of tokens per sentence **Organic**

- Beauty

The dataset contains the following amount of reviews in the selected categories:

Category	Reviews
Grocery	2737
Beauty	7162

The percentage distribution of ratings in the dataset is shown in Figure 7.

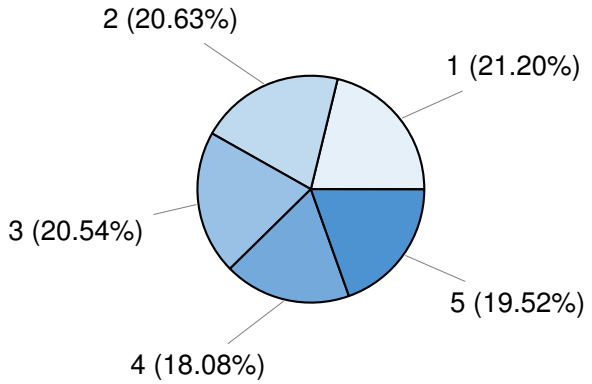


Figure (7) Percentage distribution of ratings in **amazon DE** dataset.

We can observe that the dataset is balanced. For this reason, no balancing strategy was applied to it in our experiments.
Just as we did with **amazon EN**, we used NLTK to split this dataset into sentences. After doing so, we ended up with **28.994** sentences. Figure 8 shows the number of sentences per comment in the dataset.

After filtering the relevant categories, only 3 sentences in the dataset had more than 100 tokens. We removed these outliers for our experiments. Figure 9 shows the number of tokens per sentence after removing the outliers.

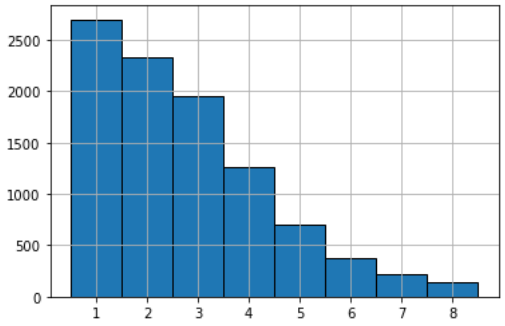


Figure (8) Number of sentences per comment in **amazon DE**.

It is interesting to see how different the distribu-

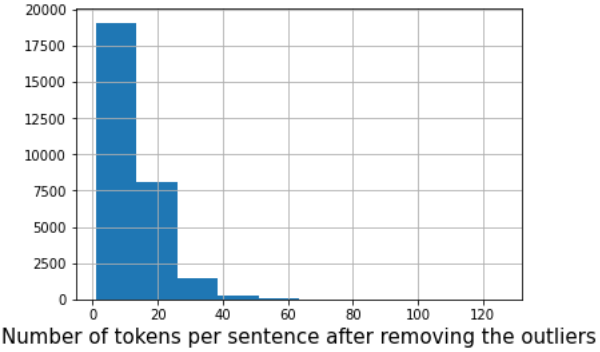


Figure (9) Number of sentences per comment in **amazon DE**.

tions for **amazon DE**shown in figure 8 and figure 9 are from those shown in figure 4 and 5 for **amazon EN**. When comparing these distributions, it can be concluded that German authors tend to write comments with fewer sentences than English ones, but those sentences are longer.

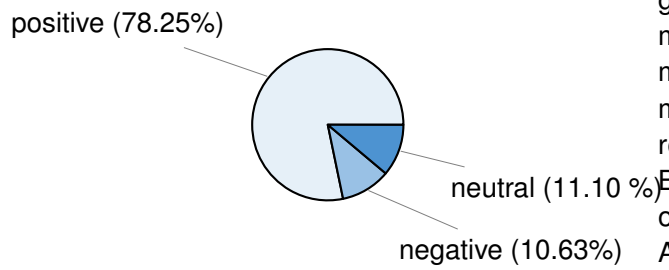
3.2 Sentiment assessment from review ratings

In our case, we are interested in using our data to predict sentiment.
Given that most of our data consists of Amazon reviews with the number of stars given by the customer, we required a strategy to asses the sentiment of a review based on its stars. The strategy that we used for this purpose is described in Algorithm 1.

Figure 10 shows the distribution of sentiment for these datasets after splitting them into sentences and extracting sentiment from them using the described strategy.

In total we had 131.349 sentences in **amazon EN**

Amazon EN:



Amazon DE:

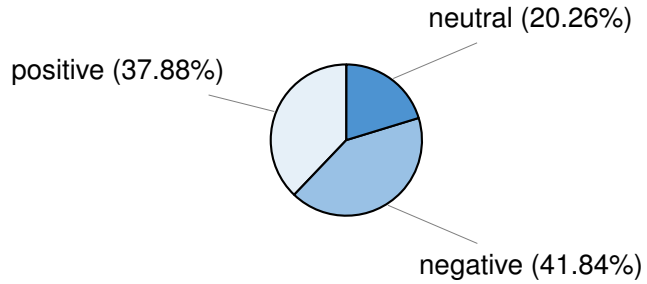


Figure (10) Sentiment distribution for **amazon EN**and **amazon DE**after extracting sentiment from reviews

and 28.994 in **amazon DE**.

3.3 Embeddings

In our experiments, we used different types of embeddings for our data. In this section we will describe how we got embeddings for each experiment using different models and tools.

3.3.1 Embeddings for monolingual experiments

For our monolingual experiments, we used BERT base uncased to generate embeddings for our sentences. For this, we used the tool "Bert as a service".
In order to make use of all the syntactic and semantic properties learned by BERT, it is necessary to use a deep layer of the model for the embeddings generation. Nevertheless, the last layer of the model is too close to the target functions, which means that embeddings extracted from this level may be biased towards those targets. For these reasons, we used the second-to-last layer from the BERT model to generate embeddings for the tokens of our sentences.
As pooling strategy to get sentence embeddings, we used the mean of the tokens of the sentence.

Algorithm 1 Sentiment Assessment from number of stars

```
procedure SentimentAssessment
    if numberOfStars < 3 then
        sentiment ← negative
    else if numberOfStars = 3 then
        sentiment ← neutral
    else
        sentiment ← positive
    return sentiment
```

3.3.2 Embeddings for cross-lingual experiments

For our cross-lingual experiments, we used different models and tools:

Multilingual Bert:
For the cross-lingual BERT experiments, we used the model "Bert Base Multilingual Cased" and the tool "Bert as a service". To generate embeddings for our sentences, we used the -2 layer (second-to-last) and the mean of the tokens as pooling strategy.

RoBERTa:
To generate embeddings with RoBERTa, we used the RoBERTa model provided by the transformers library. Also, we used the library sentence-transformers to generate sentence embeddings. As we did with our other BERT models, we used the -2 layer and the mean of the tokens as pooling strategy.

XLING:
For the generation of XLING embeddings, we used the Tensorflow module: universal-sentence-encoder-xling-many. Also, we used the sentence-piece library as tokenizer for our sentences.

3.3.3 Different context level embeddings

For the experiment described in section 3.5.3, we used BERT base uncased layer -2 and the mean of the tokens to generate sentence embeddings.

3.4 Baselines

In the project, we used different baselines to compare with our results.
In this section we are going to explain the baselines we used and the data used for training them.

3.4.1 Sentiwordnet

SENTIWORDNET is the result of the automatic annotation of all the synsets of WORDNET according to the notions of "positivity", "negativity", and "neutrality". Each synsets is associated to three numerical scores: Pos(s), Neg(s), and Obj(s) which indicate how positive, negative, and "objective" (i.e., neutral) the terms contained in the synset are [Baccianella et al., 2010]. We used this tool to test sentiment in **amazon EN** and **organic**.
Algorithm 2 shows the prodedure we used to extract sentiment from our sentences using sentiwordnet. In our experiments, we used alpha = 0.3
For our experiments with two classes, we used the procedure shown in algorithm 3.

Algorithm 2 Sentiment extraction using Sentiwordnet for three classes

```
procedure GETSENTIMENT (sentence, alpha)
    for each word in sentence do
        word ← lemmatizeSentence(word)
        word ← getMostCommonSynset(word)
        wordSent ← getWordSentiment(word)
        sentenceSent ← sentenceSent + wordSentPositive − wordSentNegative
    if alpha ∗ −1 <= sentenceSent <= alpha then return neutral
    else if sentenceSent >= 0 then return positive
    elsereturn neutral
```

Algorithm 3 Sentiment extraction using Sentiwordnet for three classes

```
procedure GETSENTIMENT (sentence, alpha)
    for each word in sentence do
        word ← lemmatizeSentence(word)
        word ← getMostCommonSynset(word)
        wordSent ← getWordSentiment(word)
        sentenceSent ← sentenceSent + wordSentPositive − wordSentNegative
    if sentenceSent >= 0 then return positive
    elsereturn negative
```

3.4.2 VADER

VADER is a simple rule-based model for general sentiment analysis that performs exceptionally well in the social media domain [Hutto and Gilbert, 2015]. In our case, we used VADER to test sentiment in **amazon EN** and **organic**.

3.4.3 TextBlobDE

TextBlob is a library for processing textual data. It provides a simple API for diving into common natural language processing tasks such as sentiment analysis [tex,]. TextblobDE is an extension of this library for the German Language. We used TextBlobDE to test sentiment on **Amazon DE**.

3.4.4 NLTK sentiment analyzer

For this baseline, we used NLTK Sentiment Analizer to train a Naive Bayes model using our data both for training and testing.

3.4.5 Scikit-learn SVM model

As our last baseline, we trained an SVM model using Scikit-learn. For this model, we used bigrams in order to preserve a little bit more context of the sentence for classification.

3.5 Experiments

For all the experiments, we were using sentence embeddings as input features and comment sentiments as ground truth labels.

3.5.1 Monolingual analysis

We started the project with *monolingual* experiments — for this setting, we worked only with data in English, i.e. **amazon EN** and **organic** datasets.
The motivation for the whole project was the idea that the **amazon EN** dataset contains some useful information that can help with classifying the **organic** dataset as they share the same domain. Thus, the main experiments for monolingual setting were the following:

- `amazon EN-organic` — training on the **amazon EN** dataset and testing on the **organic** dataset.
- `amazon EN-amazon EN` — training and testing on the **amazon EN** dataset.

For both of the experiments, we also had an option of fine-tuning the model on the **organic** dataset.
For comparison purposes, we conducted

`organic-organic` experiment as well. For this, we considered each sentence in the **organic** dataset as a separate comment, and fed such training data into MilNet. This led to the degenerate attention mechanism in the network as the only attention weight was equal to 1.0 for every data point. Using this approach, we turned MulNet into a simple RNN model and obtained a competitive baseline for other experiments on the **organic** dataset.

3.5.2 Cross-lingual analysis

The next stage of the project was introducing new dataset in German — **amazon DE**. We were interested how well the model can transfer semantic features from one language to another. Thus, we held out the whole **amazon DE** dataset as a test set and conducted an experiment `amazon EN-amazon DE`.
In the cross-lingual setting, we expored different initial embeddings, e.g. BERT multilingual, RoBerta and XLING.

3.5.3 Different context level embeddings

Another area of investigation was to see if giving more context to our sentences when generating ist embeddings gave an improvement in the sentiment analysis task. In our previous experiments, we used only the sentence as context for generating embeddings. For this experiment, we generated the embeddings of our sentences using the entire comment as context. The following example illustrates this:

Generation of sentence-level context embeddings:

Tokenized Comment: CLS, 'I', 'really', 'love', 'the', 'product', '.','It', 'is', 'really', 'tas', '##ty', SEP.

- Context for the generating embeddings for sentence 1: 'I', 'really', 'love', 'the', 'product', '.'
- Tokens used for the generating embeddings for sentence 1: 'I', 'really', 'love', 'the', 'product', '.'
- Context for the generating embeddings for sentence 2: 'It', 'is', 'really', 'tas', '##ty'.
- Tokens used for the generating embeddings for sentence 2: 'It', 'is', 'really', 'tas', '##ty'.

True	-	0	+
-	154	67	12
0	70	174	44
+	46	55	141

Figure (11) Confusion matrix for `amazon EN-organic` trained on BERT base embeddings.

Generation of comment-level context embeddings:

Tokenized Comment: CLS,'I', 'really', 'love', 'the', 'product', '.','It', 'is', 'really', 'tas', '##ty',SEP.

- Context for the generating embeddings for sentence 1: 'I', 'really', 'love', 'the', 'product', '.','It', 'is', 'really', 'tas', '##ty'
- Tokens used for the generating embeddings for sentence 1: 'I', 'really', 'love', 'the', 'product', '.'
- Context for the generating embeddings for sentence 2: 'I', 'really', 'love', 'the', 'product', '.','It', 'is', 'really', 'tas', '##ty'.
- Tokens used for the generating embeddings for sentence 2: 'It', 'is', 'really', 'tas', '##ty'.

For getting both types of embeddings, we used BERT base uncased layer -2 and the mean of the tokens as pooling strategy to get sentences embeddings.
As mentioned before, BERT can only handle sentences with a maximum of 512 tokens including [CLS] and [SEP]. In our data, 9,3% of the comments had 510 tokens or more.
After removing these otulier comments, we lost 34% of our total sentences. This explains why the results in this experiment are lower compared to the other experiments we did.

3.5.4 Two-class analysis

The confusion matrix for the conducted experiments gave us a motivation for exploring the reduced problem with only two classes (*negative* and *positive*). As you can see in Figure 11, *neutral* class introduces a lot of confusion for the model. Thus, we dropped the *neutral* class from all the datasets and repeated the experiments on this reduced data.

4 Results

We were using both micro and macro F1 scores as target metrics.

4.1 Monolingual

We trained our network on BERT base embeddings for two experiments: `amazon EN-amazon EN` and `amazon EN-organic`. The main interest was in comparing fine-tuned and not fine-tuned models.

As you can see in Figure 12, training only on the

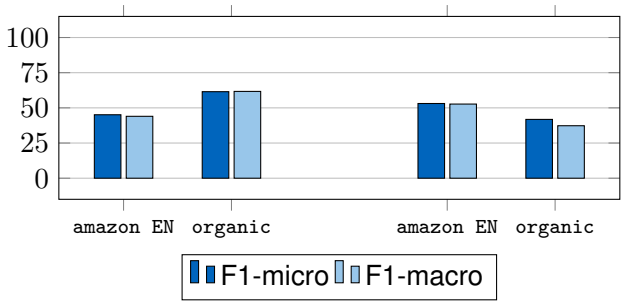


Figure (12) Monolingual `amazon EN-amazon EN` and `amazon EN-organic`: with fine-tuning (left) and without fine-tuning (right).

amazon EN yields poor results for the **organic**, while fine-tuning on the **organic** significantly drops the performance on the **amazon EN**. Such effect may occur because comments in **amazon** EN and **organic** have different structure or they've been annotated in slightly different ways.

Figure 12 shows that we should not fine-tune the model if we want to get the best results for **amazon EN**. Thus, from now on experiment `amazon EN-organic` will always assume that fine-tuning on **organic** was performed; experiment `amazon EN-amazon EN` – that it was not.

In Figure 13 you can see the results for all the monolingual experiments. The plots show that task `organic-organic` is easier than `amazon EN-organic` not only for our model but for the baselines as well. Again, it may be the evidence of some notable distinctions between **amazon EN** and **organic** datasets.

4.2 Cross-lingual

Figure 14 shows the results for our cross-lingual experiments performed on different initial embeddings. We can see that for the first two experiments all the embeddings produced similar results. Interestingly,

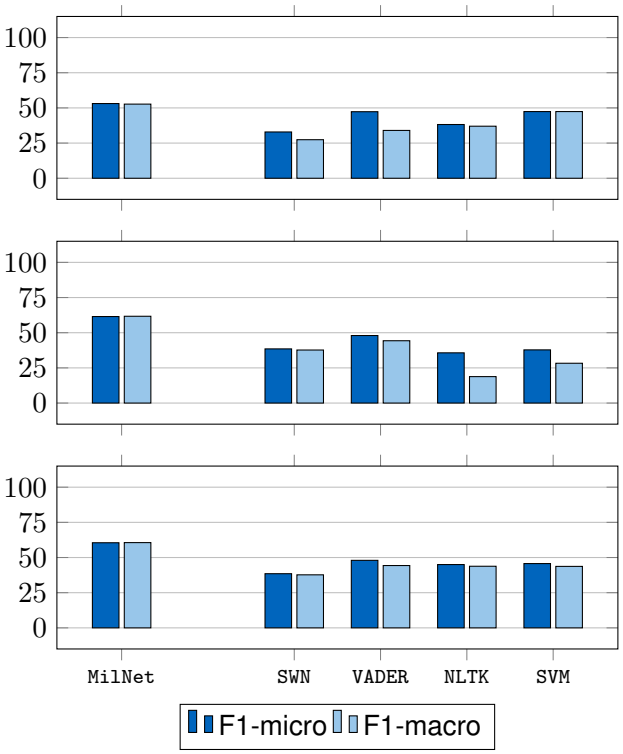


Figure (13) Comparison of MilNet and baselines for the monolingual experiments: `amazon EN-amazon EN` (top), `amazon EN-organic` (middle) and `organic-organic` (bottom)

.

although **amazon DE** was used only as a test set, all the embeddings yielded better F1-micro scores on the `amazon EN-amazon DE` then on the `amazon EN-amazon DE`. Also, Textblob outperformed both NLTK and SVM as well. Possibly, this result could be explained by the quality of **amazon DE** and the features of the German language — for example, one could suggest that Germans express their opinions in a clearer way.

4.3 Different context level embeddings

In our next experiment, we compared two models trained on BERT multilingual embeddings with different levels of context: "comment as a context" and "sentence as a context". As was described in section 3.3.3, for getting valid embeddings we had to discard a large portion of the data. Thus, the results in the Figure 15 for the `amazon EN-amazon EN` are worse than the ones shows in the Figure 14. Note that comment-level context coincides with sentence-level context for comments having only one sentence. For this reason, we didn't run this experiment for the **organic** dataset.

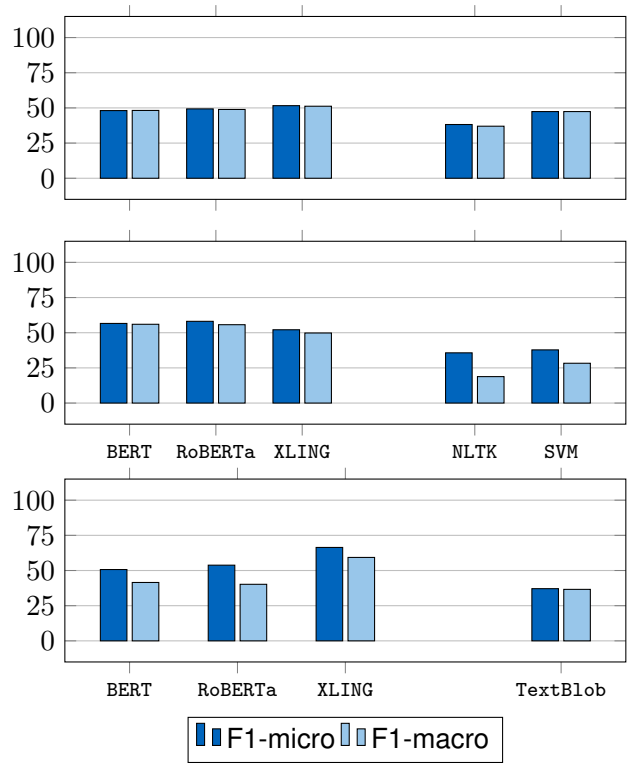


Figure (14) Comparison of different multilingual embeddings and baselines: `amazon EN-amazon EN` (top), `amazon EN-organic` (middle) and `amazon EN-amazon DE` (bottom)

.

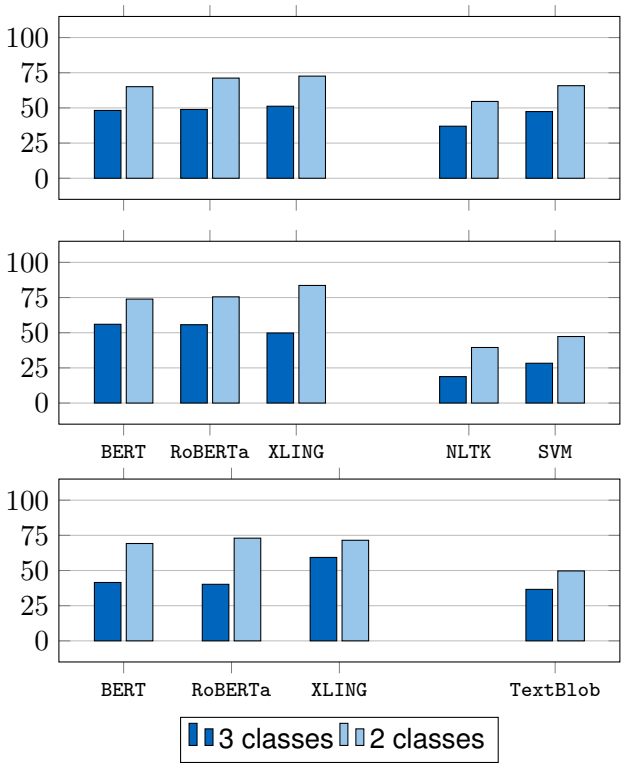


Figure (16) Obtained F1-macro scores for the two-class task: `amazon EN-amazon EN` (top), `amazon EN-organic` (middle) and `amazon EN-amazon DE` (bottom)

.

As you can see in Figure 15, for our task comment-level context didn't cause any significant improvement in the results.

4.4 Two-class

As our last step, we trained a model for predicting one of the two sentiments: *negative* or `positive`. Figure 16 shows that for all the experiments reducing the number of classes triggers major improvements in F1-macro score. Similarly to the Figure 14, for the two-class problem `amazon EN-amazon DE` produces better results than `amazon EN-amazon EN`.

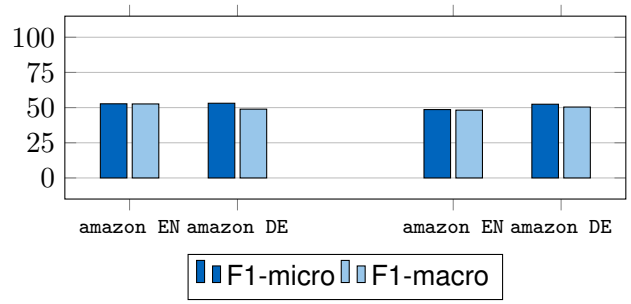


Figure (15) Cross-lingual `amazon EN-amazon EN` and `amazon EN-amazon DE`: comment-level context (left) and sentence-level context (right).

5 Conclusions

- In all of our experiments, MilNet outperformed all the baselines. This shows that it is indeed a valid technique for the sentiment analysis that can lead to good results.
- Even when they come from the same domain and context, **Amazon EN** and **Amazon DE** have notorious differences because of the language. This is reflected not only in the structure of the sentences themselves, but also in the results obtained in our classification experiments. For some experiments, results for the German data are better, possibly, due to the features of the language and the quality of the dataset.
- In our setting, comment-level context for embeddings didn't produce significantly better results than sentence-level context.

- For most of the experiments, different embedding produced similar results. In the overall result, XLING produced the best results for the task. Despite this, this result can not be generalized, because the embedding selection depends heavily on the task and data used.
- Presence of *'neutral'* sentiment makes the task much harder for the models and baselines. This is probably because models choose to predict *'neutral'* when unsure about the decision.
- Originally, the idea of the project was to train a model on **amazon EN** and then fine-tune in on **organic**. This approach gave just a small improvement on the results of pure **organic** training;
- Neither Amazon nor organic data use the whole power of the MilNet architecture. We have only comment-level labels for the Amazon data, hence, we cannot properly test the performance on the task of predicting sentiments for individual sentences. For the organic data, the situation is opposite: we have only sentence-level labels, so we cannot properly train all the levels of the model.

Acknowledgements

We want to thank Gerhard Hagerer, M.Sc., for a great lab course. We learned a lot from it, and we had a lot of fun!

References

[tex,] Simplified text processing.

[Angelidis and Lapata, 2017] Angelidis, S. and Lapata, M. (2017). Multiple instance learning networks for fine-grained sentiment analysis.

[Baccianella et al., 2010] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of LREC*, 10.

[Dang et al., 2020] Dang, N. C., Moreno-García, M. N., and De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

[Ditlevsen et al., 2019] Ditlevsen, K., Sandoe, P., and Lassen, J. (2019). Healthy food is nutritious, but organic food is healthy because it is pure: The negotiation of healthy food choices by danish consumers of organic food. *Food Quality and Preference*, 71:46–53.

[Goldberg, 2015] Goldberg, Y. (2015). A primer on neural network models for natural language processing.

[Hutto and Gilbert, 2015] Hutto, C. and Gilbert, E. (2015). Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*.

[Jurafsky and Martin, 2019] Jurafsky, D. and Martin, J. H. (2019). *An Introduction to Natural Language Processing,Computational Linguistics, and Speech Recognition*. 3 edition.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

[Loper and Bird, 2002] Loper, E. and Bird, S. (2002). Nltk: the natural language toolkit. *CoRR*, cs.CL/0205028.

[Mie et al., 2017] Mie, A., Andersen, H. R., Gunnarsson, S., Kahl, J., Kesse-Guyot, E., Rembiałkowska, E., Quaglio, G., and Grandjean, P. (2017). Human health implications of organic food and organic agriculture: a comprehensive review. *Environmental Health*, 16(1):111.

[Rücklé et al., 2018] Rücklé, A., Eger, S., Peyrard, M., and Gurevych, I. (2018). Concatenated power mean word embeddings as universal cross-lingual sentence representations.