UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL DE CALCULATOARE

PROIECT DE DIPLOMĂ

Analiză de sentimente bazată pe aspecte pentru recenzii online

Alin-Georgian Pisică

Coordonatori științifici:

Conf. Dr. Ing. Elena-Simona Apostol
Ș.L. Dr. Ing. Ciprian-Octavian Truică
As. Drd. Ing. Alexandru Petrescu

BUCUREȘTI
2022

UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

DIPLOMA PROJECT

Aspect Based Sentiment Analysis for Online Reviews

Alin-Georgian Pisică

Thesis advisors:

Conf. Dr. Ing. Elena-Simona Apostol
Ș.L. Dr. Ing. Ciprian-Octavian Truică
As. Drd. Ing. Alexandru Petrescu

BUCHAREST
2022

CONTENTS

SINOPSIS

Analiza de sentimente bazată pe aspecte este o subproblemă a analizei de sentimente, ce implică extragerea polarității sentimentale în raport cu aspecte cheie din text. Această lucrare extinde metodele de analiză de sentimente în vederea extragerii de termeni cheie din recenzii oferite restaurantelor, împreună cu polaritatea corespunzătoare acestora. În implementare am plecat de la două variante preantrenate de transformers (BERT Base și BART BASE), pe care le-am extins prin specializare asupra domeniului și adăugare de noi straturi și rețele în cadrul modelului. Evaluarea este făcută pe seturile de date *"SemEval 2016 - Task 5 Restaurants"* și *"MAMS for Aspect Term Extraction"*. Rezultatele arată o îmbunătățire a scorurilor de acuratețe comparativ cu cercetări anterioare făcute asupra topicului.

Cuvinte cheie: Analiză de Sentimente, Extragere de Aspecte, LSTM, CNN, Analiză de sentimente bazată pe aspecte, BERT, BART, transformer, Notare IOB, Atenție

ABSTRACT

Aspect based sentiment analysis is a sub-task of the sentiment analysis, aiming to identify the polarity towards the main aspect terms from the text. This thesis extends the sentiment analysis methods into the aspect extraction and term analysis on online reviews offered to restaurants. To implement the solution, we started from two pre-trained transformers (BERT Base and BART Base) and extended them through fine-tuning and adding new layers and networks. The evaluation is done on *"SemEval 2016 - Restaurants"* dataset and *"MAMS for Aspect Term Extraction"* dataset. The results show a great improvement in accuracy compared to the previous researches done on the ABSA topic.

Keywords: Sentiment Analysis, Aspect Term Extraction, LSTM, CNN, Aspect Based Sentiment Analysis, BERT, BART Transformer, IOB tagging, Attention

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude for the support received from my advisors, Elena, Ciprian, and Alexandru, for their guidance and the motivation boost that kept me going even when things were not working out. I am grateful for the opportunity that they offered me and for all the things that I've learned in the past year. I am also thankful to all the people I've met during my years as a student. Last, but not least, I would be remiss in not mentioning the support received from my girlfriend, and also my family, without whom I would not be who I am today. To you, all, thank you for having my back and trusting me!

1 INTRODUCTION

Aspect Based Sentiment Analysis comes as an extension of the so-popular Sentiment Analysing problem in the NLP domain by introducing an additional difficulty represented by the aspect identification. The current literature proposes various models, from simple classifiers that solve specific tasks (aspect extraction, keywords extraction or sentiment analysis) to more complex architectures involving transformers, pre-trained models or chained neural networks.

Due to the necessity of correctly labelled corpora which, until a few years ago, was hard to find, the ABSA solutions have shown a slow start in the early development years, with small improvements from a solution to another. However, in the last years, with the apparition of new datasets that encapsulate specific domain data with multiple polarities in each specific sample, ABSA gained a big amount of interest that lead to high accuracy implementations, based on various architectural frameworks and models.

1.1 Motivation

In the last decades, social networks have provided a new and efficient way of communication and sharing opinions about products, actions, places, and general feelings. Aspect Based Sentiment Analysis (ABSA) is an important task in natural language processing domain, solving the opinion observing and polarity extractions by collecting sentiments analyzed related to the specific aspect terms.

Even though the general sentence polarity analysis (sentiment analysis) has been used in a large variety of domains, the ABSA task is not something that is implemented very often due to the huge challenge created by the large variety of possible aspects and categories. ABSA aims to identify the fine-grained polarity towards specific aspects of the text, allowing users to easily visualize and understand complex language formations, giving a granular understanding of the quality and the polarities transmitted.

1.2 Objectives

In the last years, the pre-trained language models have proven to be effective in replacing and improving the feature engineering phase of the development. Transformers like BERT [7] (Bidirectional Encoder Representations from Transformers) and GPT [22] (Generative Pre-trained Transformer) have introduced the possibility of creating state-of-the-art NLP models with as little as a fine-tuning process. However, due to the complexity of the ABSA task, a

simple fine-tune is not enough for obtaining an accurate solution.

To observe the impact of transformers over the complicated looking NLP tasks, we will put under test three different models against two datasets consisting of online reviews with either multiple or simple aspects and polarities. After the fine tuning procedure the models should obtain state of the art results with consistency.

The objective of the thesis is to propose a solution for the aspect based sentiment analysis problem by reducing the difficulty through the two sub-tasks that we will solve: Aspect Term Extraction (ATE), in which we aim to identify the correct aspects of each input and Aspect Term Sentiment Analysis (ATSA), that obtains the polarity, either negative, neutral or positive, for each aspect of the review. We propose three models that aim for state of the art results in both tasks, on two datasets.

1.3 Structure of the thesis

The thesis is structured on 6 main chapters, containing the current state of the art, previous results obtained using basic models and transformers, the data analysis, implementation and analysis of the models and comparisons between each subtask and datasets.

In the State of the Art Chapter, we look at the current literature, researches and implementations, spotlighting the approach, results and the key points that can be utilised in understanding the difficulties that may appear, as well as extracting the useful breakthroughs made.

For the third chapter, Methodology, we present the networks and techniques that we'll be using and go through the architectural structure of the models.

In the Implementation Chapter, we present the application's components and the way they interact.

The exploratory data analysis for the datasets, corpus distribution, hardware used are presented in the first half of the Experimental Results Chapter. In the second half, we present the results obtained on all models and datasets and a comparison with existing models that hold the state of the art status in the ABSA task.

2 STATE OF THE ART

In this chapter, we present and analyze the current State of the Art architecture proposed in aspect based sentiment analysis task. For further explanations, we will split the ABSA problem in 2 sub-tasks and will go through them independently: *Aspect Term Extraction* and *Aspect Term Sentiment Analysis*.

2.1 Sentiment Analysis

Sentiments are views or opinions expressed towards actions or objects. Sentiment Analysis is the task of identifying the polarity of a given text and marking it as negative, positive or neutral. The sentiment detection can be made in various ways, ranging from a rudimentary implementation that identifies bad and good words or expressions, to linguistic models or deep learning techniques like BERT and RoBERTa.

The current proposal for the Sentiment Analysis task consists of a language representation model that claims to achieve state-of-the-art performances on eleven NLP tasks - BERT [7] (described in Subsection 3.1.1).

Previously, Pal et. al made use of LSTMs in *Sentiment Analysis in the light of LSTM Recurrent Neural Networks* [19] for analysing a dataset consisting of IMDB reviews labelled as positive or negative. The methodology presented 3 implementations, a conventional LSTM which peaked at an accuracy of 80.92%, a deep LSTM which reached 81.32% and the final one, based on bidirectional deep LSTMs, which topped the ranking with an accuracy of 83.83%. The complexity increase proved to reduce the validation loss value by 0.3 between the conventional LSTM and the BiLSTM.

Himanshu Batra, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal presented in *BERT-Based Sentiment Analysis: A software Engineering Perspective* [4] an approach for sentiment analysis problem using BERT Transformer. The implementation uses as a base layer the BERT transformer, leveraging the fine-tuning possibility by adding an additional untrained layer on top of the model, aiming to solve task-specific problems. Over multiple datasets, the model is keeping high accuracy, in the range of 93-94%. The proposed solution shows a big improvement in solving the Sentiment Analysis task by just fine-tuning a pre-trained transformer, obtaining in short time (over the span of few epochs), with decent

sized datasets, state of the art results.

2.2 Aspect and Feature Extraction

Features or aspects are entities present in the text on which the focus and sentiments are targeted. The entities can range from products or topics to individuals, places, services or, more general, categories. The main goal of the aspect extraction is to identify the targeted entities or groups of entities and match them together with the classified polarity, in order to identify the positive and negative areas and how the overall feelings of the reviewer can be improved.

Due to a great and fast growth of the internet as how we know it today, opinions started rising on every product available and going through them manually can be a time consuming task. If, until now, simple sentiment analysis has shown to be efficient in filtering good and bad reviews on various applications, introducing the aspect extraction can provide a better analysis in the automatic understanding of the emotions and identify the downsides that are presented in an entity. In this way, the manual reading and analysis can be shifted to dashboards, reports or visualisations of the reviews, offering a faster identification of the key elements present in the feedback received.

"Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction" by Hu Xu, Bing Liu, Lei Shu and Phillip S. You [32] proposed an aspect extraction method based on two embedding layers (one for general-purpose and one domain-specific) combined with 4 convolutional neural networks. The processed text sequence results in two embeddings that are concatenated, passed through the CNN layers and forwarded to a fully-connected layer that outputs the IOB tagging of the sentence. The experiments are executed on SemEval 2014 and SemEval 2016 datasets, resulting in a maximum accuracy of 81.59% on the laptops dataset and 74.37% on the restaurants dataset.

He et. al presented in 2018, in *"An Unsupervised Neural Attention Model for Aspect Extraction"* [9] a neural approach in aspect extraction, focused on discovering coherent aspects. The model exploits the distribution of word co-occurences using word embeddings that encourage and help identify words that are located at a similar distance over multiple contexts. The novel technique makes use of an attention mechanism for down-weighting non-aspect words. Trained and validated on *Cityseach corpus* and *BeerAdvocate*, the model obtained an average precision over the tree aspects presented (food, staff and ambience) of 95.3%, 80.2% and 81.5%.

2.3 Aspect Based Sentiment Analysis

Aspect Based Sentiment Analysis combines the previous mentioned sub-tasks (presented in Section 2.1 and Section 2.2). Based on the results obtained in the previous stages, ABSA aims to identify and pair the aspect extracted with the corresponding sentiment.

Even though, lately, most of the implementations and research done on the topic make use of transformers as state of the art techniques, in 2015, Duyu Tang et. al [28] presented a promising approach of handling the aspect identification and polarity matching based on Long Short Term Memory (LSTM) networks over a benchmark dataset consisting of tweets. The approach is presented in three stages. In the first step, the solution is based on a simple LSTM, merging the word embeddings and the vocabulary into an embedding matrix in order to carve the semantic representation of the sentences. Even though the target words (aspects) are not took into consideration, it still manages to achieve an accuracy of 66.5%. In the second stage, the simple LSTM turns into a Target-Dependent LSTM, considering the aspects and the surrounding context, based on two LSTM that model the previous and following contexts (a LSTM that runs from left to right and another one that runs from right to left). The improvements of contextual analysis raise the accuracy by almost 5 percent, to a value of 70.8%. The last improvement, Target-Connection LSTM, brings into light the connections between the aspects and the contextual related words, and reaching a peak of 71.5% accuracy.

Aspect-Based Sentiment Analysis using BERT [10] solves the ABSA problem by creating 3 models: one for aspect classification, one for polarity classification and one that combines the previous two for obtaining the final result. For the SemEval16 restaurants dataset the model obtains promising results on both the sentence-level datasets and text-level datasets. The aspect classification model uses BERT for sentence pair classification. On the sentence-level analysis over the SemEval16 restaurants reviews, the Aspect Category Classifier obtains an F1 score of 79.90% and 96.30% accuracy, while the text-level analysis is situated at 85% F1 score with 88.70% accuracy. For the sentiment model, on the sentence-level is obtained a F1 score of 87.0% with an accuracy of 87.3%, while on the text-level a F1 score of 86.3% with an accuracy of 87.5%.

Aspect-based Sentiment Analysis using BERT with Disentangled Attention [15] makes use of DeBERTa (Decoding-enhanced BERT with Disentangled Attention), which is an improvement of BERT and RoBERTa transformers by using a disentangled attention mechanism and an enhanced mask decoder. The disentangled attention technique represents each word using two vectors for encoding the content and the position, while the attention weights are computed in accord to these. The second mechanism, the enhanced masked decoder replaces the softmax layer to predict the masked tokens. The research uses the SemEval 2014 datasets, both the restaurants and laptop datasets. For the restaurants dataset, the model performs with a mean F1 score of 81.39% and an average accuracy of 86.11%.

The three researches presented above proved that transformers can bring a great improvement in the contextual understanding of the text and the related aspects, with a major increase in accuracy. The two implementations that make use of BERT were based on simple, fine-tuned, transformers with one additional layer for classifying the result and mainly using datasets that have either a single aspect per sentence or multiple aspects but with the same opinion (the SemEval16 is presented and analysed in Subsection 5.1.1).

3 METHODOLOGY

This chapter aims to create an overall picture of the architecture, the models used and the reasons behind the structure. In Subsection 3.1, we'll walk through the transformer used, the neural networks and algorithms, while in Subsection 3.2, we'll analyse the architecture and why we think they might be a good choice in the implementation of the model.

3.1 Algorithms/Models

3.1.1 BERT (Bidirectional Encoder Representations from Transformers)

BERT (Bidirectional Encoder Representations from Transformers) [7] is based on pre-trained deep bidirectional representations through conditioning on left and right context. Thus, fine-tuning can be made using a single additional output layer resulting in state-of-the-art performances. Since a transformer contains two mechanisms - an encoder that processes the input text and a decoder that produces the prediction, BERT is using only the encoder mechanism for producing the language model. The encoder reads the entire input sentence at once, therefore being considered bidirectional, as opposed to the directional models which read text sequentially (left to right or right to left). The training phase consists of two tasks: masked language model (MLM) and next sentence prediction (NSP), presented in Figure 1.

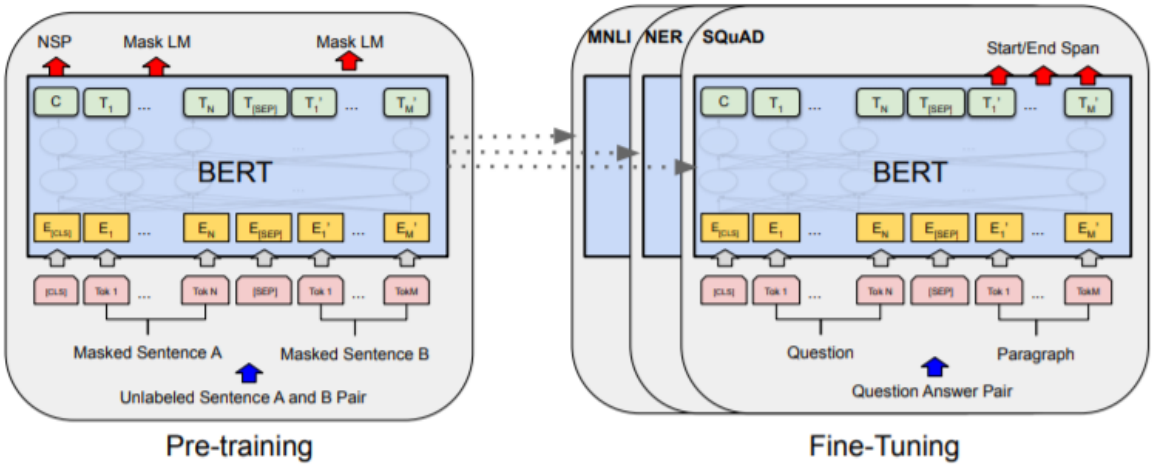

Figure 1: BERT pre-training and fine tuning process [7]

In MLM, before feeding the text, random words are replaced with a [MASK] token. The model then attempts to predict the original input sentence based on the context provided by the non-masked words in the sentence. Usually, the masked token percentage is around 10-20% of the total tokens being masked, most common the value being situated at 15%. The advantage that this method is bringing represents a better understanding of a particular language usage in a specific domain. The weights are optimised through predicting known words to gain the ability to predict unseen words that are contextually related. We will exemplify the MLM procedure using the phrase *"I loved the french fries so much"* with one masked token in Figure 2.

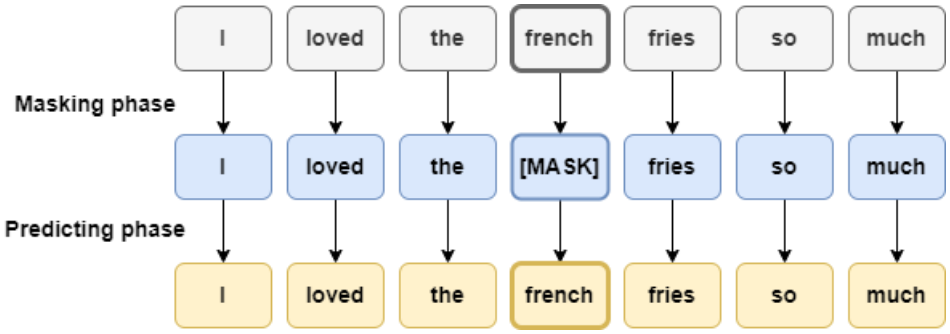

Figure 2: Masked Language Modeling - masking and prediction phases

For the NSP task (exemplified in Table 1), the model is feeded with pairs of sentences and learns to predict the relation between them. For the sentences separation, a [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is appended at the end of every sentence. When training the BERT model both tasks are performed simultaneously, in order to minimize the loss functions of the two techniques. As an example, we will present in Table 1 the correlation between the phrase *"I like the french fries"* and two other sentences: *"They were perfect"* which is in direct relation with the first sentence and *"It was 8 A.M."* which has no connection with the initial phrase.

Table 1: Next Sentence Prediction example

Start token	First sentence	Second sentence	Next Sentence Result
[CLS]	I like the french fries [SEP]	They were perfect [SEP]	Related
[CLS]	I like the french fries [SEP]	It was 8 A.M [SEP]	Not Related

The BERT model comes pre-trained on a huge dataset, with one downside: the training processed is being done on unlabelled text, without a fixed objective or a pre-defined goal of the training phase. The main scope of the training is to gather the contextual representation and connections between words from different domains, categories and parts of speech, in order to create connections between them or between expressions. There are two main pre-trained versions of BERT, BERT Base and BERT Large, that differ in the number of encoder layers used (BERT Base makes use of 12 encoding layers while BERT Large doubles the amount, resulting in 24 different encoding layers). BERT Base comes with 110 million trainable parameters while the double number of stacked encoding layers in BERT Large obtain a number of 340 million trainable data. However, in most of the cases, BERT case proves to be sufficient for obtaining state of the art results with minimal hardware resources.

In order to specialise BERT over a specific dataset or task, there is a need to perform a fine-tuning task, by adding an additional layer on top of the transformer. The data used in the fine-tuning process comes as a labelled text dataset, and with the already training phase done, the particularisation of BERT can be done with a small amount of epochs (the recommended number being between 2 and 4 epochs according to Devlin et al. [7]).

3.1.2 BART (Bidirectional Auto-Regressive Transformers)

BART (Bidirectional Auto-Regressive Transformer) was proposed by Lewis et. al [13] as a generalised version of BERT (summarised in Subsection 3.1.1) due to the existence of the bidirectional encoder. The pre-training phase is similar, in technique, to the pre-training method used on BERT. To pre-train the model, which was proposed as a denoising sequence-to-sequence solution capable to claim the state-of-the-art title, there is a need for generalisation over the MLM and NSP tasks used in BERT. Similar to the MLM used in BERT, randomly positioned tokens are replaced with the *[MASK]* token. Additionally, on top of masking, we approach into discussion the token deletion for randomly placed tokens, permutations and rotations of the document and sentence, and also text infilling (presented in Figure 3).

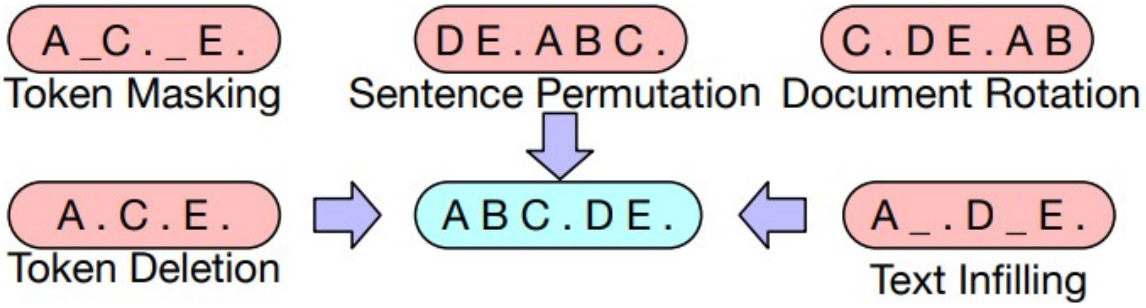

Figure 3: BART text transformations [13]

As stated by Lewis [13], BART outperforms BERT on most common tasks and proves a strong consistency thanks to the text infilling and rotation procedures. Even though it comes in base and large forms, the additional number of trainable parameters that are added compared to BERT (more than 130M parameters for the base version, compared with BERT base, which has 110M trainable parameters) increases drastically the hardware requirements for fine-tuning and working with BART, even in its smallest form.

3.1.3 BiLSTM (Bidirectional Long-Short Term Memory)

In 1997, Sepp Hochreiter and Jürgen Schmidhuber proposed *Long Short-Term Memory* [11], a novel network that differs from the classical feedforward neural networks by having feedback connections. The vanishing gradient problem present in Recurrent Neural Networks is being solved with the apparition of the LSTM networks, these allowing the information to persist.

The LSTM cell (Figure 4) consists of three gates that process the information and control the persistence of the data: Forget Gate (f), Input Gate (I) and Output Gate (O). Based on the three gates, we can split the data processing into three phases.

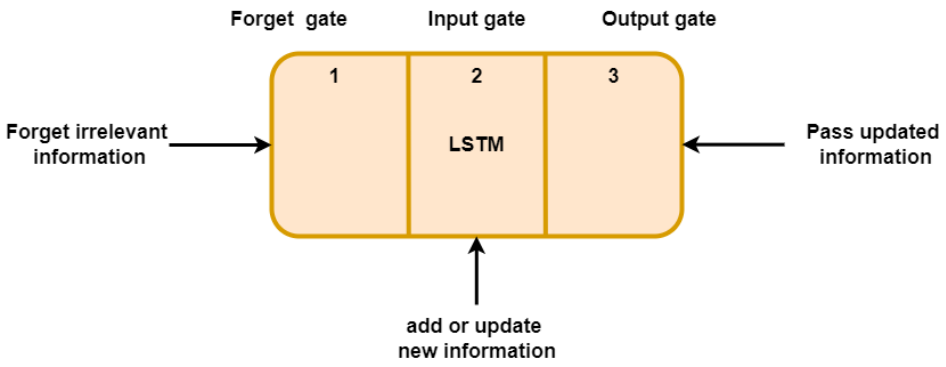

Figure 4: LSTM Cell

The first step consists in deciding whether the information from the previous step should be kept or discarded. The Input Gate, based on contextual dependencies, quantifies the importance of the input and processes the data as a function of the previous hidden state (H) and the input at the current step. As the name suggests, the Output Gate, consisting of the third and last step in the LSTM cell, handles the calculation of the hidden layer step and obtaining the result using an activation function. [25]

With W, U being weight vectors for the previously mentioned gates (forget gate, candidate, input gate, output gate), X being the input vector, H_t and C_t the cell output and cell memory at any given time t, we can reduce the LSTM calculations to the following relations (Equation (1)) [18]:

$$\begin{aligned} f_t &= \sigma(X_t * U_f + H_(t-1) * W_f) \\ \overline{C_t} &= \tanh(X_t * U_c + H_(t-1) * W_c) \\ I_t &= \sigma(X_t * U_i + H_(t-1) * W_i) \\ O_t &= \sigma(X_t * U_o + H_(t-1) * W_o) \\ C_t &= f_t * C_(t-1) + I_t * \overline{C_t} \\ H_t &= O_t * \tanh(C_t) \end{aligned} \tag{1}$$

Bidirectional LSTM (BiLSTM) [8] is a recurrent neural network. Unlike the LSTM, the BiLSTM network contains one additional LSTM layer, which allows the information transport in both ways. The BiLSTM architecture brings a big improvement in the NLP domain, due to the fact that every element contains information from both directions, being able to produce a more meaningful, contextual output. The bidirectional analysis is simplified in Figure 5 from Mohan and Gaitonde [17].

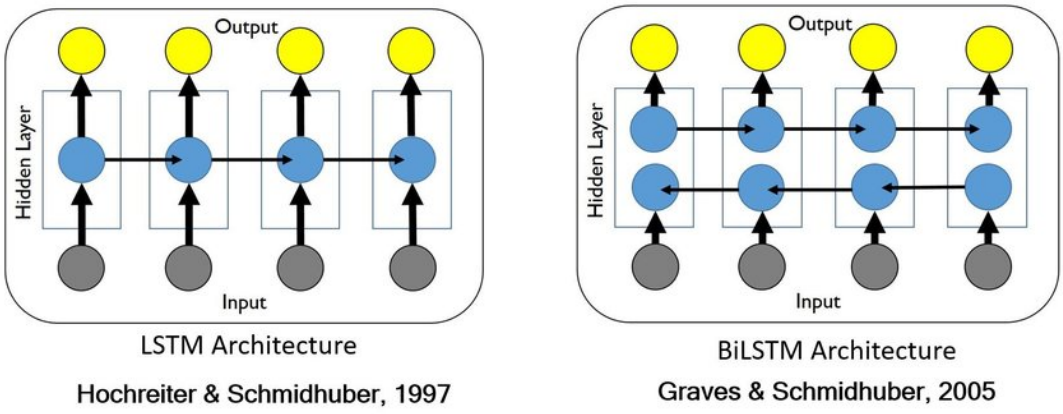

Figure 5: LSTM and BiLSTM Architectures [17]

In 2019, Augustyniak et al. conducted a series of experiments in *Comprehensive Analysis of Aspect Term Extraction Methods using Various Text Embeddings* [2], focusing particularly on LSTM architectures. The experiments were executed over the SemEval datasets proving

that BiLSTM outperforms classical LSTMs in term of scores and results when using the same embeddings.

3.1.4 CNN (Convolutional Neural Network)

Convolutional Neural Networks (CNN) are networks that instead of proceeding with the basic matrix multiplication are using a convolution operation through the help of a kernel / filter. The kernel is an auxiliary matrix that is multiplied over the convolutional layer, resulting in an abstracted, uniformed feature map (also called activation map). The convolution operation is executed as a sliding window dot product, in which the kernel keeps moving over every section of the input data, thus creating the feature map. Due to an additional pooling layer and a reduced size of the kernels, CNNs can have a reduced number of parameters compared to other neural networks (architectural example presented in Figure 6).

The convolution operation can be simplified to a matrix of size NxN, a mxm kernel and weights w to the following equation (Equation (2)), x being the result of the convolution process, $aandb$ the starting index of the matrix and y a function used for adjusting the numerical value output: [29]

$$x_{ij} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab} y_{(i+a)(j+b)}^{l-1} \tag{2}$$

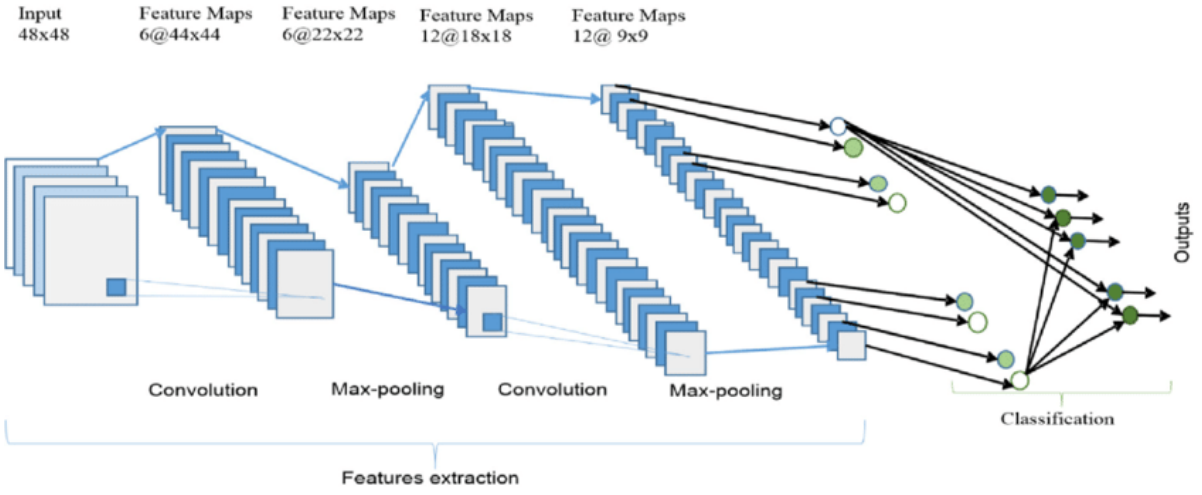

Figure 6: CNN Architecture

The pooling operation can be perceived as a function application over consecutive sub-matrices of the input data, in order to generalise and reduce the parameters used in training. Adjacent input values are combined making the model less sensible to small changes in the dataset.

The most common pooling methods are the max pooling (Figure 7) in which the maximum element is kept and average pooling (Figure 8) in which the pooled value is represented by the average of the total values in the current sliding window.

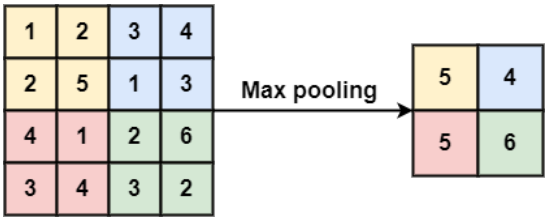

Figure 7: Max Pooling

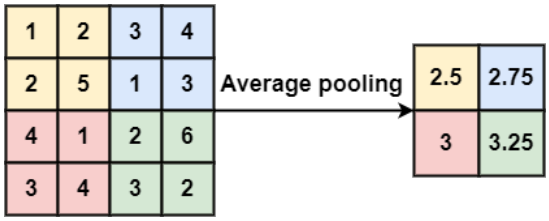

Figure 8: Average Pooling

Even though the CNNs are mostly used in image processing tasks, due to the easy transform over 3 dimensional spaces and the feature maps obtained, the idea behind can be easily implemented in a NLP task. In the case of Aspect Based Sentiment Analysis, where each word is processed through a word embedding layer and resulting in a vectorized form, the CNN can easily be implemented on top of any word embedding, resulting in a high impact of the features extracted over the output.

3.1.5 IOB (Inside-Outside-Beginning) tagging

IOB Tagging is a tagging format for tagging tokens. Each important token receives a corresponding value from I (inside), O (outside) or B (beginning). The B value represents the beginning of the tag, the elements are marked with I if they are inside of an important attribute / entity / aspect term and the O value is assigned to all the outside elements. Cho et al. [6] conducted an experiment with different tagging methods using the datasets CoNLL-2003 Shared Task [23] and BioCreative II [24] designed for Named Entity Recognition, obtaining 86.81% accuracy with the IOB tagging method.

The IOB tagging is the technique we will use for marking the aspect terms of the reviews. The following example (Figure 9) shows the sentence "I liked the pizza and the open kitchen" with the aspect terms identified being "pizza" and "open kitchen". Since "pizza" is a single noun, it will be marked with the B tag, while the "open kitchen", being compound, will be identified using both the B and I tags.

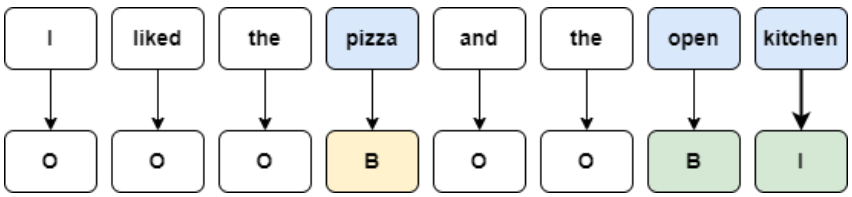

Figure 9: IOB Tagging

3.2 Architecture

From an architectural point of view, we created three different models for the two sub-tasks (Aspect Term Extraction and Aspect Term Sentiment Analysis), having as a starting point the BERT and BART transformers. Each model will be implemented in four variants: with BERT base pre-trained and the fine-tuned variants and using the BART transformer, also with the pre-trained and fine-tuned variants.

In order to convert the text into processable units, we use a tokenizer. Tokenization is a NLP task that handles the text splitting into smaller and structured units called tokens. A token can be represented either by a character, a word or a n-gram (group of n words). In order to obtain the tokens, the text is splitted into segments by delimiters (that can be either white spaces or custom delimiters), followed by a word tokenization phase, in which each token is assigned a unique identifier (Example in figure 10). For splitting the text into segments we used spaCy, a NLP framework that obtains state of the art results on NLP tasks of part of speech identification and sentence splitting over English datasets.

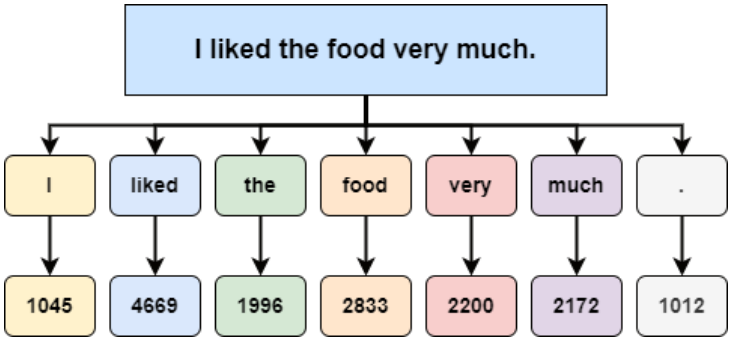

Figure 10: Tokenization process of "*I liked the food very much.*" using the BERT Base Tokenizer

Small and medium sized datasets processed in neural networks can lead to an over fitting behaviour. BERT base already provides 110 million parameters which combined with additional modules can produce a big trainable amount of values. On a relatively small dataset like SemEval, this can emerge into an over fitting situation pretty quickly. In order to avoid this scenario, on top of each BERT/BART instance we will add a dropout layer as a regularisation method to improve the generalization and reduce the overfitting possibility.

The final layer of the model is represented by a linear layer, reducing the final predictions to 3 possible labels for the aspect term extraction procedure (0, 1 and 2 representing the Outside, Beginning and Inside tags from the IOB tagging method) and 4 possible labels for the sentiment analysis task (0 if no sentiment was extracted, 1 if the sentiment is negative, 2 for neutral and 3 for positive).

On top of the trained/fine-tuned models we added a web interface to easily test, analyse

and visualise the behaviour of the models on different inputs, scenarios and domains. The communication between the models and the web application is made through a REST API service that exposes the models to be consumed (more details in Section 4.2 and Section 4.3).

3.2.1 BERT/BART + Dropout + Linear

Figure 11: Model using Dropout and Linear

The simplest architecture (Figure 11), being at the same time represented by the model that comes into use in the fine tuning phase, consists of two additional layers added on top of BERT and BART models: a dropout layer for balancing the input data and classes and reducing the overfitting scenario and a linear layer for producing the final classification of the data. In order to process the data, the transformer has to receive the text input as a tokenized form of the initial sentence, where each word gets assigned a individual token.

Saving the transformer model obtained after the first training phase will obtain the fine-tuned version, which will later be used in all three models. Once fine-tuned, from a general version of the model we will obtain a specialised model over the dataset. The downside of the fine-tuning procedure is represented by the additional time needed to produce the specialised variant and also by the additional memory needed to store the model.

3.2.2 BERT/BART + Dropout + BiLSTM + Linear

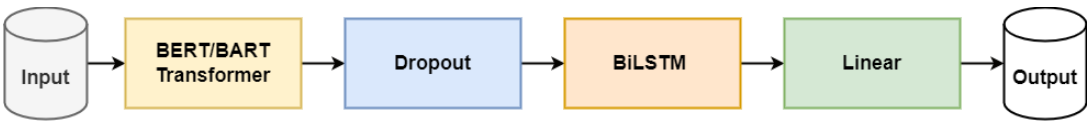

Figure 12: Model using Dropout, BiLSTM and Linear

For the second model (Figure 12), we will add a Bidirectional LSTM between the dropout and linear layers, in order to better understand and analyse the contextual dependencies between the tokens. As proven by Augustyniak et al. [2], a bidirectional LSTM adds an improvement on the traditional LSTM in terms of scoring and efficiency.

The final goal of the experiment being to analyse and understand online reviews from a sentimental point of view, the contextual dependency is one of the main challenges since online reviews have no structured and formalised syntactic construction. In this manner, the objective of the newly added network is to directly create and generalise connections between words and memorise patterns over the multiple reviews applied to the same domain and aspects.

3.2.3 BERT/BART + Dropout + CNN + BiLSTM + Linear

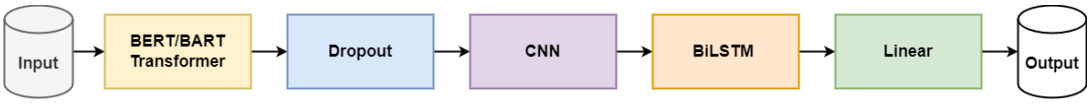

Figure 13: Model using Dropout, CNN, BiLSTM and Linear

The last architecture upgrades (Figure 13) the model previously stated in Subsection 3.2.2 by inserting a Convolutional Neural Network before the BiLSTM, in order to obtain a feature map of the review. Since the number of trainable parameters needed for a convolutional network is of a moderate size, we expect the training time to not be affected in a substantial manner. The one dimensional convolutional network can also improve the grouped textual analysis, by executing kernel convolutions over multiple tokens at a time, also improving the contextual connections between the words. The bidirectional LSTM combined with the convolution operations in the hidden layers of the CNN may show an improvement in the scoring of the experiments.

4 IMPLEMENTATION

The implementation of the models has been done using the PyTorch Framework. PyTorch is a machine learning library, as proposed by Paszke et. al [20], that gives full control to the user over the model designing and, at the same time, offers high efficiency and natural structuring while offering support for CPU and GPU processing.

For interactive running, visual analysis and an easy method of testing, we also developed a web application in React that communicates through a REST API with a Flask server that processes the data received from front-end and forwards it to the model (the connection between the components is presented in Figure 14).

Figure 14: System architecture

4.1 Model

In order to be processed, the data has to go through several phases. In the first step, the whole text gets tokenized, a procedure in which each word is assigned a unique identifier (token). The tokenized sentences are then batched in sequences of multiple inputs together, in order to advance to the training / testing steps. The batches obtained are then gathered together into a dataloader, that handles the input serving towards the model. The data flow through the system using the model containing CNN and BiLSTM network is presented in Figure 15.

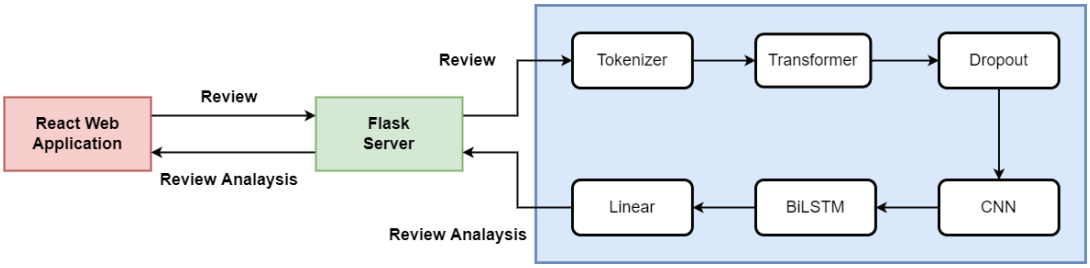

Figure 15: System architecture using Dropout + CNN + BiLSTM + Linear layers

Regardless of the transformer used, or the number of layers and networks added, the models are constructed so that each specific sub-task (ATE - Aspect Term Extraction or ATSA - Aspect Term Sentiment Analysis) has a unique, general available, input structure, in order to allow for easy scaling, upgrades and changes. The models receive data under a JSON format specific to each task. The common properties of the two input structures are represented by the full text review and a list of all tokens in the input data. For the training phase, the Aspect Term Extraction task (Figure 16) needs a third property, consisting of the IOB tagging for the entry, where the O tags are marked with 0, B tags with 1 and I tags with 0. On the other side, the Sentiment Analysis task (Figure 17), takes as a third property, a list of polarities for each token, where 1 represents the negative polarity, 2 neutral, 3 positive while 0 marks the fact that no polarity was identified for the specific token.


```
{
    "text": "Judging from previous posts this used to be a good place, but not any longer.",
    "tokens": ["Judging", "from", "previous", "posts", "this", "used", "to", "be",
               "a", "good", "place", "but", "not", "any", "longer"],
    "iob_aspect_tags": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
}
```


Figure 16: Aspect Term Extraction Input JSON Example


```
{
    "text": "The decor is not special at all but their food and amazing prices make up for it.",
    "tokens": ["The", "decor", "is", "not", "special", "at", "all", "but", "their", "food",
               "and", "amazing", "prices", "make", "up", "for", "it", "."],
    "absa_tags": [0, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0, 0]
}
```


Figure 17: Sentiment Analysis Input JSON Example

Each model is saved, after the training phase, in order to be loaded into the server (Section 4.2). Thanks to PyTorch's interface, the models are exported with all the network's summary and weights in a single file output, ready to be uploaded in any PyTorch-compatible application, allowing for easy interchangeable modules.

4.2 Flask Server

Flask is a web framework structured and introduced as a Python module, used mainly for creating simple and accessible APIs with minimal effort from developers. In order to expose the models towards the Web Application Interface (described in Section 4.3) we used Flask for exposing a public route on which the review's content is received then it starts entering the text processing procedure, in order to feed the newly retrieved data into the model. The content is forwarded to two different models, one for extracting the aspect terms and one for identifying the sentiments expressed, the result obtained being concatenated in one single JSON object that is sent back to the web application.

4.3 Web Application

The web application is written in React, a web framework that structures a single page application (SPA) in modular and reusable components, allowing direct integration and communication between HTML and Javascript.

In order to create the interface pleasable to the user's eye, the styling is done using Tailwind CSS, a utility-first CSS framework that contains atomic classes that can directly be applied, inline, to HTML components. In this way, we ensure an easy scalable and upgradable interface, by keeping a structured method for creating new elements and components.

For uploading reviews to be analysed we can opt for a custom text input (Figure 18) or a text file upload with multiple reviews, containing one review per line (Figure 19). The reviews are then forwarded to the backend, represented by the Flask Server (described in Section 4.2), which returns the processed data under the form of JSON objects. To create an easy to follow processing flow, during the upload of a file containing multiple reviews, the results will appear in the interface as the model finishes processing them.

The identified aspect terms are marked with a bold, underlined font and the sentiments extracted are highlighted using a green background for positive sentiments, red for negative and yellow for neutral polarities.

Figure 18: Web Application: Custom text input

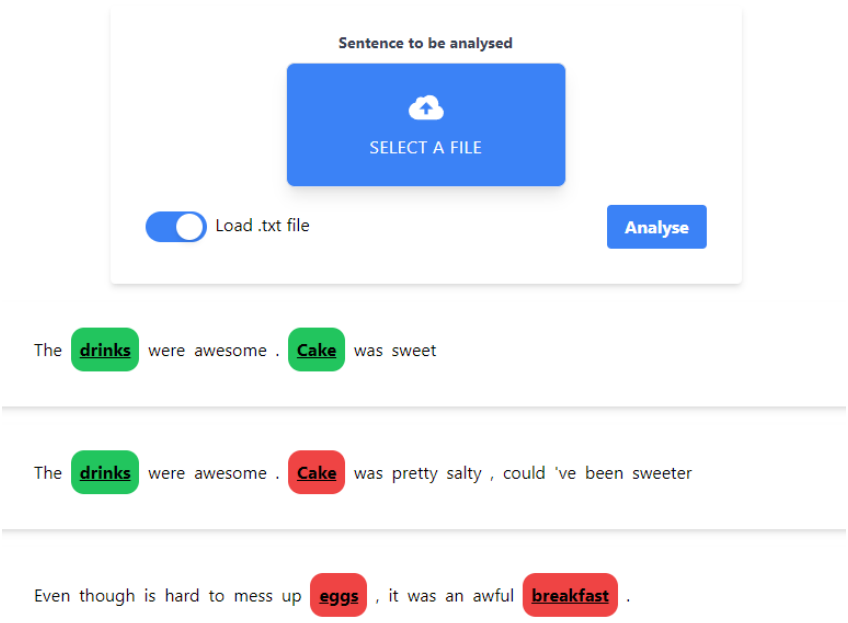

Figure 19: Web Application: File input with multiple reviews

5 EXPERIMENTAL RESULTS

5.1 Datasets details

We evaluate our methods on two datasets: *SemEval-2016 Task 5 - Restaurants* which consists of 350 reviews, summing up to 2000 sentences that have one aspect, multiple aspects or even none and the *MAMS (Multi-Aspect Multi-sentiment) dataset*, each sentence containing at least two aspects with different sentiment polarities.

5.1.1 SemEval 2016 - Task 5 - Restaurants

SemEval 2016 - Task 5 [21] is a shared task proposed in 2016 by Pontiki et. al as a challenge for solving the Aspect Based Sentiment Analysis problem. The resources proposed for the task consist in 19 datasets that span across 8 languages and 7 domains. All the datasets were expanded on top of previous versions of the challenge, and the newly added data was annotated using BRAT [26]. The English variant contains two datasets with data based on restaurants reviews and laptop reviews. For the experimental setup, we opted for the Restaurant dataset from SemEval 2016, due to high quality and relevant annotations, proved already by several researches that developed state of the art techniques.

Each review consists of a list of sentences, each sentence being structured as a list of opinions based on the sentence's text. The opinions refer to the category, aspect terms, polarity and its position indicators. The dataset contains a total of 722 aspect terms identified, the most frequent being *NULL*, meaning that no aspect was found and a general sentence polarity is offered (627 times), *food* (216 times), *place* (124 times), *service* (121 times) and the rest occurring less than 50 times each.

When it comes to reviews length, the majority of the reviews have between 3 and 8 sentences, the rest being situated at less than 10 sentences / review, the distribution being presented in Figure 20.

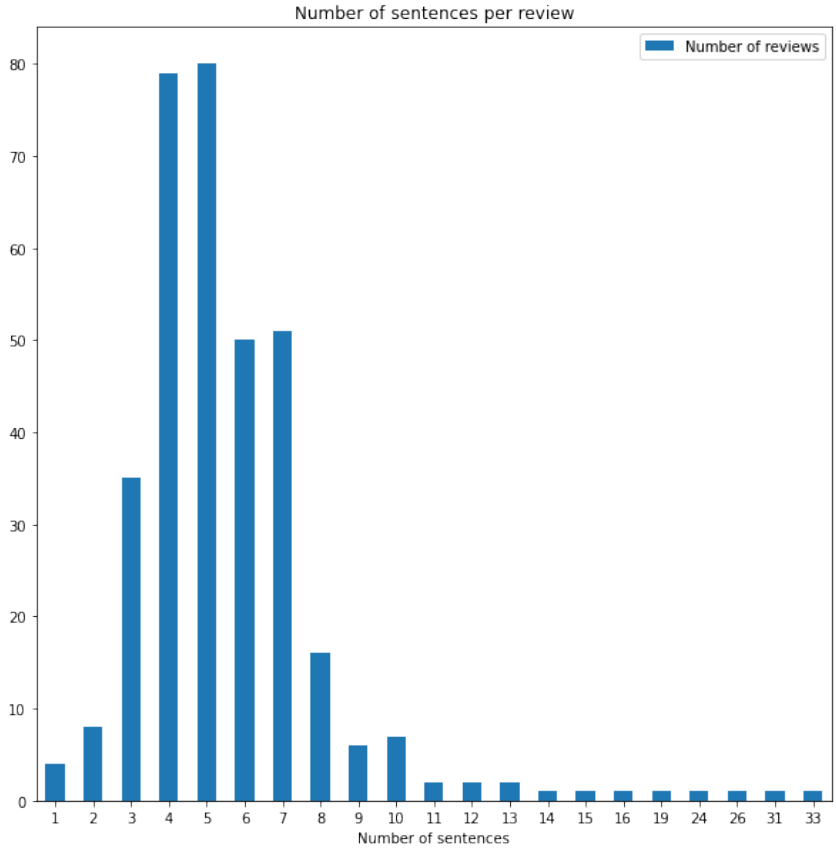

Figure 20: SemEval16 - Number of sentences per review

When analysing the categories, we can clearly see that the most frequent categories (the ones that will give the most accurate analysis in benchmarks) are *FOOD*, followed by *SERVICE* and *RESTAURANT*. As seen in Figure 21, each category has multiple sub-categories that classify the aspect terms in a more relevant manner.

The dataset presents the category in direct connection with the aspect term, overlapping the start and end indices in the review's text. In this way, the category classification can be either executed as a separate extraction from the sentence of the present categories, or a classification task over the extract aspects and labeling them into the corresponding category.

Even though the category labels are not evenly distributed, with a possibility to incline the balance towards a specific subset of aspect terms if analysed under simple and more general circumstances (simple classification), thanks to using contextual processing thorough the entire model, we can provide an in-depth analysis without worrying for the unbalanced spread of categories.

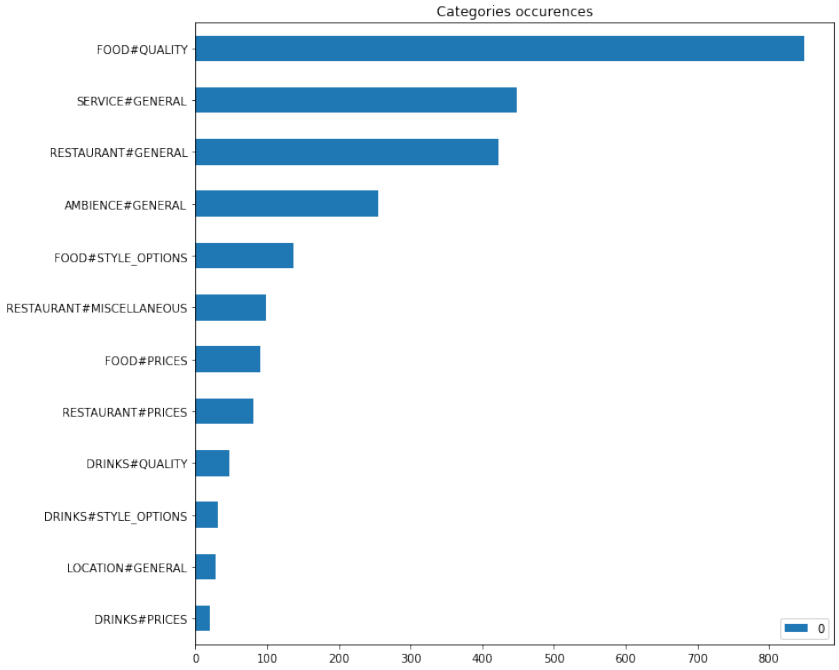

Figure 21: SemEval16 - Categories occurences

Moving to the occurrences of polarities for each category, as confirmed in *"A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis" [12]*, we can observe that the data is not evenly distributed and most of the reviews are inclined into more "sentence-polarity" than "aspect-based polarity". Only the SERVICE#GENERAL, FOOD#PRICES and RESTAURANT#PRICES categories has almost a perfect balance between the number of positive and negative reviews, the others being unbalanced (visible in Figure 22).

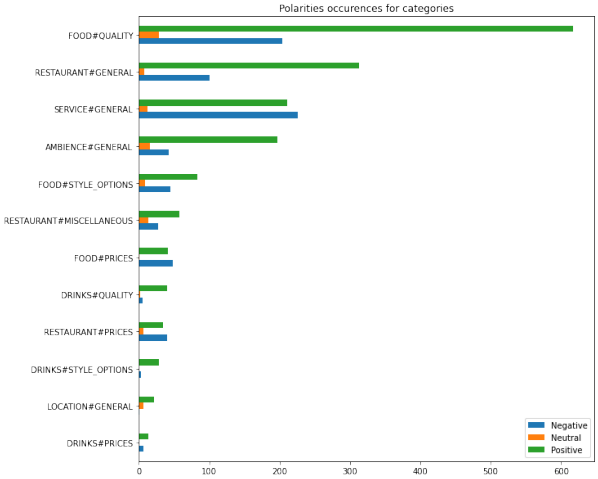

Figure 22: SemEval16 - Polarity occurrences for categories

For the number of polarities per sentence, we can see in Figure 23 that the most frequent sentences are the ones that contain only one polarity, making it hard to go into deep details when it comes to aspect-based analysis. Only less than 200 sentences have 2 different types

of polarities inside them, thus making it more inclined into the sentence analysis area.

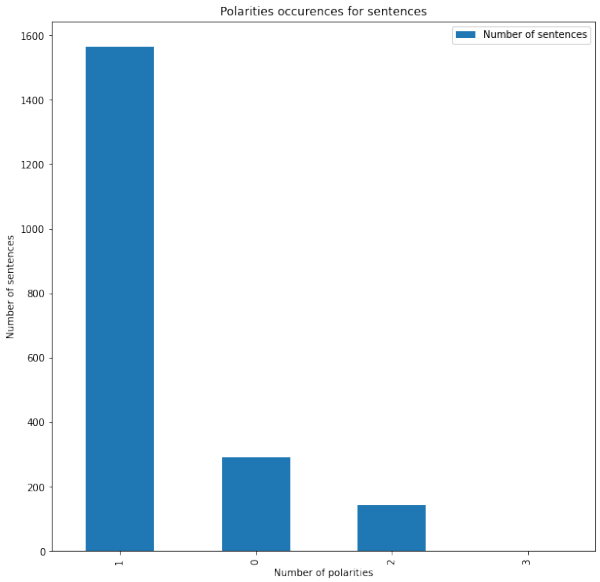

Figure 23: SemEval16 - Polarity occurrences for sentences

5.1.2 Multi-Aspect Multi-Sentiment Aspect Term Extraction (MAMS-ATE)

MAMS is a dataset created specifically for the Aspect Based Sentiment Analysis task. The issue with the existing datasets until MAMS was released was the low presence of sentences and reviews that contain more than one polarity. Every sentence in the dataset contains at least two different aspects with different sentiment polarities. [12]

As opposed to the SemEval, where the dataset contains for each aspect term the category, MAMS dataset is split in two parts: Aspect Term Sentiment Analysis and Aspect Category Sentiment Analysis. This project is using the MAMS Aspect Term Sentiment Analysis dataset, the analysis and detection being done on the aspect terms themselves.

As stated in *"A Challenge Dataset And Effective Models For ABSA" [12]*, MAMS dataset contains sentences with at least two opinions per review, going up to 8 for a small amount of sentences (Figure 24).

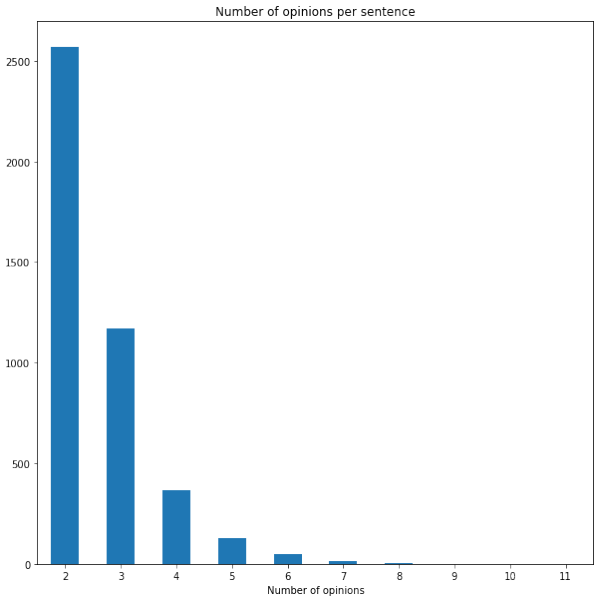

Figure 24: MAMS ATSA - Number of opinions per sentence

The dataset consists of 4297 sentences, which sum up to 11186 opinions. The total number of aspect terms identified is 2586, the five most common being *food*, *menu*, *waiter*, *service*, *dinner*. Compared to the SemEval dataset, the difficulty which, at the same time can be considered an advantage, is the number of aspects present in every review, being at least two. The distribution of the aspects throughout the dataset is shown in Figure 25).

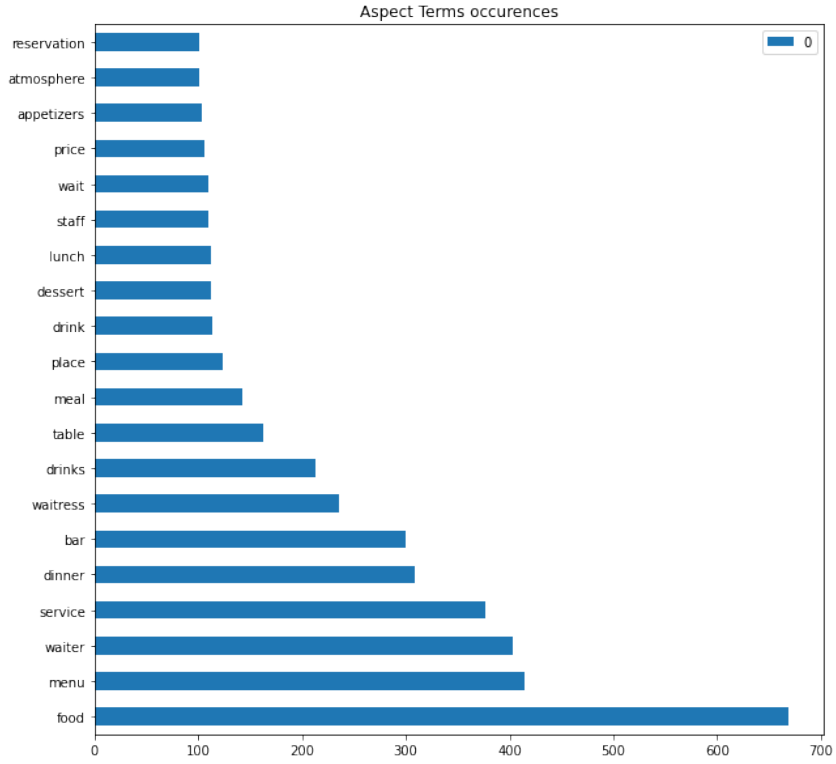

Figure 25: MAMS ATSA - Most common terms

For the polarities count, we can find in the dataset 5042 aspects with a neutral polarity (the most common polarity), 3380 positive and 2764 negative (Figure 26). Even though the dataset is not perfectly balanced, there is still a better ratio between the distribution of polarities.

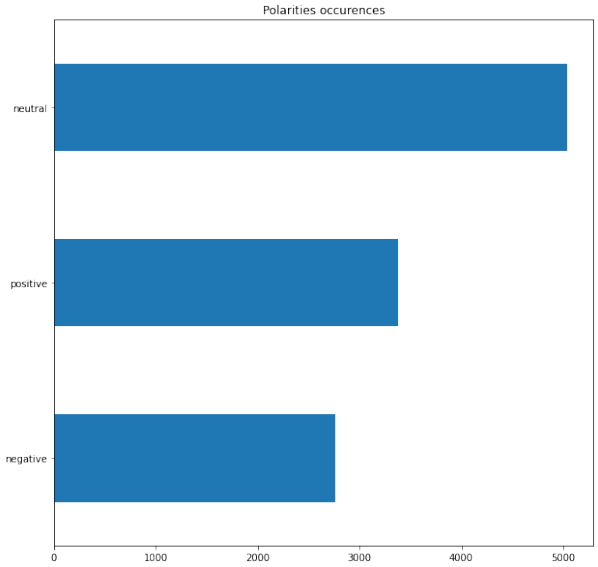

Figure 26: MAMS ATSA - number of polarities occurrences

5.2 Experimental setup

The first step in setting up the experimental environment was structuring the application's modules in easily distinguishable and accessible sections. The entire code used for training and fine-tuning the models, as well as results, plots, exploratory data analysis, web application and Flask server can be found on GitHub. For the fine tuning and training procedures we structured the code in Interactive Python Notebooks, which are computational environments that keep track of code execution, allowing it to be analysed and run in batches, which facilitated the implementation. However, part of the implementation was run on a remote server, which reduced to keeping the fine tuning and training steps for the models that use the BART transformer as simple Python files.

All the training and validation phases were ran on GPUs for a faster execution time. For working with BERT, since it allows to be run on end-user hardware specifications, the training was done using a RTX 2060 Super GPU card, with 8 GB of VRAM, 16 GB RAM and an Intel i5-9400F CPU. On the other side, BART transformer being more greedy when it comes to resources due to the higher number of trainable parameters, the training and validation phases were executed on a DGX Server with Tesla V100-DGXS GPU with 32 GB VRAM, Intel Xeon CPU E6-2698 and 264 GB RAM.

To easily process the data, which comes as XML files (Figure 27), all the reviews are converted into a JSON format(Figure 28) by extracting the review's text, aspect term's positions and

indices, as well as for polarity related to each aspect extracted. The data processed is stored in local files, which are then loaded into memory under a PyTorch Dataloader which handles the batch serving towards the model. Each run was executed using batches of size 4 for the training and validation phases, in order to keep a relevant tracking information about the execution time.

```
<sentence>
    <text>
        The decor is not special at all but their
        food and amazing prices make up for it.
    </text>
    <aspectTerms>
        <aspectTerm from="4" polarity="negative" term="decor" to="9" />
        <aspectTerm from="42" polarity="positive" term="food" to="46" />
        <aspectTerm from="59" polarity="positive" term="prices" to="65" />
    </aspectTerms>
</sentence>
```

Figure 27: Input data as seen in datasets, XML formatted


```
{
    "text": "The decor is not special at all but their food and amazing prices make up for it",
    "tokens": [
        "The", "decor", "is", "not", "special", "at", "all", "but", "their", "food",
        "and", "amazing", "prices", "make", "up", "for", "it", "."
    ],
    "iob_aspect_tags": [0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,0, 1, 0, 0, 0, 0, 0],
    "absa_tags": [0, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0, 0]
}
```


Figure 28: Input data processed as JSON file

In order to correctly analyse a model, we executed 10 runs for each model, for each dataset. A run consisted of creating a dataset split to obtain the training and validation data, with a percentage of 80% of the data towards the training step and 20% of the data being used for validation, followed by the training procedure and the validation step. The splits are randomly generated at each run, so there is relatively consistent distribution of the labels. Each training phase consists of 2 epochs, number which sits at the lower part of the recommended number by Devlin et al. [7] (from 2 to 4 epochs).

For each run we computed the accuracy score, F1 score (micro and macro), precision score (micro and macro), recall score (micro and macro) and tracked the total execution time (the time of the training and validation phases combined). The metrics are measured based on the number of correct predictions compared to the ground truth value of the dataset (prediction types are explained in Table 2).

Accuracy is the most frequent metric in general models comparison because it express the correctness of the model over all the possible classes. The metric is calculated (Equation (3)) as the ratio between the total number of correct predictions to the total number of predictions regardless of correctness.

$$accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}} \tag{3}$$

Table 2: Prediction types based on ground truth values

		Predicted	
		Positive	Negative
Ground truth	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

The precision is formulated as ratio between the number of correctly labelled as positive samples to the total number of samples computed as positive (Equation (4)), measuring the accuracy classification of a given sample as a positive result.

$$precision = \frac{True_{positive}}{True_{positive} + False_{positive}} \tag{4}$$

The recall calculation ignores the negative results, computing the ability of the model to detect positive predictions. The distribution of negative samples becomes irrelevant, making the metric to be formulated as ratio between the total number of correctly predicted positives and the total number of positive samples (Equation (5)).

$$recall = \frac{True_{positive}}{True_{positive} + False_{negative}} \tag{5}$$

Based on the recall and precision results we can compute the F1 score (Equation (6)), which can highlight the balance between the two.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

When we talk about unbalanced datasets, it's a good rule of thumb to take a look at the micro and macro average scores in order to analyse and visualise classes prediction individually. If the classes are imbalanced, the micro-average is preferred, studying the individual classes, while the macro average deals with aggregates, studying the dataset as a whole. An uneven distribution of ratio 3:2:1 in SemEval for sentiment analysis (Described in subsection 5.1.1) should result in a ratio of approximately 2:1 between the micro and macro scores in results of the SA task (Table 5).

5.3 Results

For testing and validating the solution, we extracted the average scores of the 10 runs for each combination of the datasets (SemEval or MAMS) and the sub tasks (Aspect Term Sentiment Analysis and Aspect Term Extraction). We highlighted the best values obtained with variations of the yellow colour (the brighter the yellow, the better the value). We also

extracted the Standard deviation value for all the samples and in order to easily compare the results and the behavioural changes over unbalanced datasets when the training and validation splits separate the data in a very unbalanced way.

The difference is slightly notable between the pre-trained and fine-tuned variants of the models. However, on all runs, the models obtained state of the art performance for both datasets. The difference between the size of the two datasets is clearly notable, execution time being almost 6 times bigger for some models on the MAMS dataset. Despite the difference between the micro and macro averages computed due to the unbalanced number of labels, the model kept high values for all the possible scoring methods.

For the aspect term extraction task, for both the datasets (Table 3 and Table 4), the fine tuned version of BERT with only the additional Dropout and Linear layers top the ranking with an accuracy of 99.86% on the SemEval dataset and 99.93% on the MAMS dataset. Regarding the sentiment analysis task (Table 5 and Table 6), the same model as in the ATE problem manages to get the top results with an accuracy of 99.74% on SemEval dataset and 99.87% on the MAMS dataset.

The BART transformer, even though the number of trainable parameters is clearly bigger than the BERT model, it didn't outperform the smaller rival. The classification task consist in assigning a small number of labels in a relatively medium sized text, so using BART can result in an added overhead in memory and time resources for a not-so-great improvement.

On the time scale (Figure 29 and Figure 30), the models managed to maintain a similar impact over the average execution time, for both task, ATE and SA. The 6 times bigger size of the MAMS dataset compared to SemEval16 resulted in a directly proportional time growth over the same task, with the same model.

The difference in execution time between the models that use BERT and the ones with BART is caused by the difference between the systems on which the models were run, an RTX 2060 Super with 2176 CUDA cores against a Tesla V100 with 5120 CUDA cores.

Table 3: ATE results on SemEval 2016 - Task 5 - Restaurants (Note: the highlighted cells show the best results)

Model	Accuracy	Precision		Recall		F1		Execution time (s)
		Micro	Macro	Micro	Macro	Micro	Macro	
BERT Pre-Trained Dropout Linear	99.73 ± 0.03	99.73 ± 0.03	76.16 ± 2.36	99.73 ± 0.03	69.48 ± 4.28	99.73 ± 0.03	69.71 ± 4.24	332.47 ± 0.5846
BERT Pre-Trained Dropout BiLSTM Linear	99.57 ± 0.05	99.57 ± 0.05	57.67 ± 16.27	99.57 ± 0.05	47.08 ± 8.62	99.57 ± 0.05	47.92 ± 7.88	372.70 ± 5.14
BERT Pre-Trained Dropout CNN BiLSTM Linear	99.50 ± 0.01	99.50 ± 0.01	52.26 ± 8.96	99.50 ± 0.01	35.66 ± 0.78	99.50 ± 0.01	37.21 ± 1.16	371.96 ± 22.81
BERT Fine-Tuned Dropout Linear	99.86 ± 0.03	99.86 ± 0.03	87.32 ± 4.11	99.86 ± 0.03	87.32 ± 2.77	99.86 ± 0.03	87.11 ± 3.03	326.77 ± 1.23
BERT Fine-Tuned Dropout BiLSTM Linear	99.78 ± 0.03	99.78 ± 0.03	82.24 ± 1.83	99.78 ± 0.03	72.81 ± 4.70	99.78 ± 0.03	72.04 ± 5.80	345.86 ± 16.66
BERT Fine-Tuned Dropout CNN BiLSTM Linear	99.51 ± 0.03	99.51 ± 0.03	52.30 ± 10.06	99.51 ± 0.03	36.61 ± 3.13	99.51 ± 0.03	38.23 ± 3.71	369.86 ± 4.07
BART Pre-Trained Dropout Linear	99.70 ± 0.03	99.70 ± 0.03	80.64 ± 2.68	99.70 ± 0.03	83.51 ± 2.39	99.70 ± 0.03	81.80 ± 1.78	211.28 ± 0.50
BART Pre-Trained Dropout BiLSTM Linear	99.57 ± 0.03	99.57 ± 0.03	75.50 ± 2.69	99.57 ± 0.03	67.89 ± 2.42	99.57 ± 0.03	68.10 ± 4.46	243.29 ± 0.93
BART Pre-Trained Dropout CNN BiLSTM Linear	99.33 ± 0.04	99.33 ± 0.04	49.85 ± 10.50	99.33 ±	35.68 ± 3.08	99.33 ± 0.04	36.61 ± 3.89	243.40 ± 0.86
BART Fine-Tuned Dropout Linear	99.79 ± 0.02	99.79 ± 0.02	86.16 ± 1.71	99.79 ± 0.02	92.24 ± 1.10	99.79 ± 0.02	88.94 ± 0.83	213.07 ± 13.81
BART Fine-Tuned Dropout BiLSTM Linear	99.77 ± 0.03	99.77 ± 0.03	84.64 ± 2.48	99.77 ± 0.03	89.38 ± 0.92	99.77 ± 0.03	86.75 ± 1.50	246.07 ± 20.36
BART Fine-Tuned Dropout CNN BiLSTM Linear	99.34 ± 0.03	99.34 ± 0.03	47.65 ± 12.05	99.34 ± 0.03	34.78 ± 1.61	99.34 ± 0.03	35.77 ± 2.77	245.83 ± 12.57

Table 4: ATE results on MAMS ATE (Note: the highlighted cells show the best results)

Model	Accuracy	Precision		Recall		F1		Execution time (s)
		Micro	Macro	Micro	Macro	Micro	Macro	
BERT Pre-Trained Dropout Linear	99.78 ± 0.02	99.78 ± 0.02	89.17 ± 1.42	99.78 ± 0.02	94.13 ± 2.08	99.78 ± 0.02	91.47 ± 0.97	1432.14 ± 7.92
BERT Pre-Trained Dropout BiLSTM Linear	99.74 ± 0.03	99.74 ± 0.03	85.44 ± 1.78	99.74 ± 0.03	94.90 ± 1.13	99.74 ± 0.03	89.72 ± 0.92	1594.72 ± 2.75
BERT Pre-Trained Dropout CNN BiLSTM Linear	99.20 ± 0.06	99.20 ± 0.06	70.70 ± 1.55	99.20 ± 0.06	51.87 ± 7.30	99.20 ± 0.06	56.76 ± 8.28	1616.63 ± 1.99
BERT Fine-Tuned Dropout Linear	99.93 ± 0.01	99.93 ± 0.01	96.07 ± 0.62	99.93 ± 0.01	98.74 ± 0.37	99.93 ± 0.01	97.37 ± 0.20	1439.25 ± 22.51
BERT Fine-Tuned Dropout BiLSTM Linear	99.90 ± 0.01	99.90 ± 0.01	94.63 ± 0.81	99.90 ± 0.01	98.58 ± 0.53	99.90 ± 0.01	96.54 ± 0.38	1583.13 ± 48.36
BERT Fine-Tuned Dropout CNN BiLSTM Linear	99.40 ± 0.02	99.40 ± 0.02	75.72 ± 1.57	99.40 ± 0.02	71.82 ± 2.77	99.40 ± 0.02	73.16 ± 1.65	1632.40 ± 9.71
BART Pre-Trained Dropout Linear	99.50 ± 0.04	99.50 ± 0.04	80.94 ± 2.06	99.50 ± 0.04	91.05 ± 1.19	99.50 ± 0.04	85.30 ± 1.15	959.79 ± 2.34
BART Pre-Trained Dropout BiLSTM Linear	99.48 ± 0.03	99.48 ± 0.03	79.48 ± 1.64	99.48 ± 0.03	92.40 ± 0.90	99.48 ± 0.03	85.01 ± 0.86	1089.97 ± 6.72
BART Pre-Trained Dropout CNN BiLSTM Linear	98.87 ± 0.06	98.87 ± 0.06	65.12 ± 5.74	98.87 ± 0.06	57.16 ± 6.51	98.87 ± 0.06	60.00 ± 5.90	1104.66 ± 21.54
BART Fine-Tuned Dropout Linear	99.68 ± 0.02	99.68 ± 0.02	86.95 ± 1.16	99.68 ± 0.02	96.20 ± 0.41	99.68 ± 0.02	91.12 ± 0.83	947.81 ± 3.50
BART Fine-Tuned Dropout BiLSTM Linear	99.68 ± 0.02	99.68 ± 0.02	87.14 ± 0.88	99.68 ± 0.02	96.14 ± 0.37	99.68 ± 0.02	91.27 ± 0.51	1090.74 ± 2.53
BART Fine-Tuned Dropout CNN BiLSTM Linear	99.06 ± 0.07	99.06 ± 0.07	70.82 ± 2.69	99.06 ± 0.07	66.28 ± 4.40	99.06 ± 0.07	67.96 ± 3.08	1088.94 ± 2.01

Table 5: ATSA results on SemEval 2016 - Task 5 - Restaurants (Note: the highlighted cells show the best results)

Model	Accuracy	Precision		Recall		F1		Execution time (s)
		Micro	Macro	Micro	Macro	Micro	Macro	
BERT Pre-Trained Dropout Linear	99.73 ± 0.02	99.73 ± 0.02	42.79 ± 6.40	99.73 ± 0.02	46.31 ± 1.18	99.73 ± 0.02	43.29 ± 0.96	339.95 ± 11.62
BERT Pre-Trained Dropout BiLSTM Linear	99.58 ± 0.04	99.58 ± 0.04	42.67 ± 2.39	99.58 ± 0.04	34.03 ± 5.19	99.58 ± 0.04	35.77 ± 4.31	367.49 ± 8.08
BERT Pre-Trained Dropout CNN BiLSTM Linear	99.49 ± 0.01	99.49 ± 0.01	35.95 ± 10.56	99.49 ± 0.01	25.13 ± 0.26	99.49 ± 0.01	25.19 ± 0.50	362.88 ± 19.41
BERT Fine-Tuned Dropout Linear	99.79 ± 0.01	99.79 ± 0.01	56.93 ± 10.99	99.79 ± 0.01	49.23 ± 2.46	99.79 ± 0.01	46.82 ± 3.75	327.17 ± 5.87
BERT Fine-Tuned Dropout BiLSTM Linear	99.74 ± 0.02	99.74 ± 0.02	41.09 ± 0.83	99.74 ± 0.02	46.94 ± 0.33	99.74 ± 0.02	43.55 ± 0.53	357.82 ± 18.75
BERT Fine-Tuned Dropout CNN BiLSTM Linear	99.51 ± 0.02	99.51 ± 0.02	36.31 ± 12.46	99.51 ± 0.02	25.44 ± 0.81	99.51 ± 0.02	25.74 ± 1.44	381.66 ± 9.65
BART Pre-Trained Dropout Linear	99.62 ± 0.02	99.62 ± 0.02	49.95 ± 4.28	99.62 ± 0.02	46.92 ± 0.80	99.62 ± 0.02	44.44 ± 1.43	213.05 ± 1.42
BART Pre-Trained Dropout BiLSTM Linear	99.53 ± 0.03	99.53 ± 0.03	39.54 ± 0.68	99.53 ± 0.03	41.23 ± 2.79	99.53 ± 0.03	40.20 ± 1.04	243.48 ± 0.81
BART Pre-Trained Dropout CNN BiLSTM Linear	99.34 ± 0.02	99.34 ± 0.02	30.06 ± 7.98	99.34 ± 0.02	25.11 ± 0.18	99.34 ± 0.02	25.13 ± 0.35	242.74 ± 0.81
BART Fine-Tuned Dropout Linear	99.73 ± 0.03	99.73 ± 0.03	58.69 ± 2.68	99.73 ± 0.03	55.47 ± 4.88	99.73 ± 0.03	54.50 ± 5.02	208.67 ± 1.38
BART Fine-Tuned Dropout BiLSTM Linear	99.68 ± 0.02	99.68 ± 0.02	41.00 ± 0.75	99.68 ± 0.02	47.60 ± 0.75	99.68 ± 0.02	43.72 ± 0.42	239.87 ± 0.44
BART Fine-Tuned Dropout CNN BiLSTM Linear	99.36 ± 0.03	99.36 ± 0.03	33.15 ± 7.39	99.36 ± 0.03	26.82 ± 3.20	99.36 ± 0.03	27.31 ± 3.82	241.19 ± 0.90

Table 6: ATSA results on MAMS ATE (Note: the highlighted cells show the best results)

Model	Accuracy	Precision		Recall		F1		Execution time (s)
		Micro	Macro	Micro	Macro	Micro	Macro	
BERT Pre-Trained Dropout Linear	99.69 ± 0.02	99.69 ± 0.02	82.68 ± 1.55	99.69 ± 0.02	88.20 ± 2.17	99.69 ± 0.02	85.20 ± 0.73	1445.99 ± 12.80
BERT Pre-Trained Dropout BiLSTM Linear	99.62 ± 0.03	99.62 ± 0.03	78.08 ± 1.78	99.62 ± 0.03	85.84 ± 1.39	99.62 ± 0.03	81.58 ± 1.21	1594.97 ± 1.49
BERT Pre-Trained Dropout CNN BiLSTM Linear	99.12 ± 0.03	99.12 ± 0.03	49.65 ± 3.42	99.12 ± 0.03	31.52 ± 3.39	99.12 ± 0.03	33.56 ± 3.98	1612.20 ± 8.09
BERT Fine-Tuned Dropout Linear	99.87 ± 0.01	99.87 ± 0.01	91.44 ± 1.10	99.87 ± 0.01	96.04 ± 0.48	99.87 ± 0.01	93.64 ± 0.46	1437.03 ± 1.19
BERT Fine-Tuned Dropout BiLSTM Linear	99.84 ± 0.01	99.84 ± 0.01	89.42 ± 1.36	99.84 ± 0.01	95.29 ± 0.73	99.84 ± 0.01	92.18 ± 0.63	1584.32 ± 1.01
BERT Fine-Tuned Dropout CNN BiLSTM Linear	99.19 ± 0.02	99.19 ± 0.02	49.09 ± 8.84	99.19 ± 0.02	40.42 ± 1.79	99.19 ±0.02	40.75 ± 2.01	1612.22 ± 1.85
BART Pre-Trained Dropout Linear	99.41 ± 0.04	99.41 ± 0.04	76.74 ± 2.12	99.41 ± 0.04	84.11 ± 1.34	99.41 ± 0.04	80.04 ± 1.03	948.75 ± 1.42
BART Pre-Trained Dropout BiLSTM Linear	99.35 ± 0.02	99.35 ± 0.02	73.30 ± 0.90	99.35 ± 0.02	82.69 ± 1.54	99.35 ± 0.02	77.36 ± 0.81	1083.69 ± 0.95
BART Pre-Trained Dropout CNN BiLSTM Linear	98.75 ± 0.04	98.75 ± 0.04	49.41 ± 6.37	98.75 ± 0.04	33.97 ± 4.65	98.75 ± 0.04	36.44 ± 5.42	1088.83 ± 5.80
BART Fine-Tuned Dropout Linear	99.61 ± 0.03	99.61 ± 0.03	82.87 ± 1.57	99.61 ± 0.03	91.91 ± 0.43	99.61 ± 0.03	87.00 ± 0.86	944.19 ± 2.13
BART Fine-Tuned Dropout BiLSTM Linear	99.58 ± 0.05	99.58 ± 0.05	81.65 ± 2.25	99.58 ± 0.05	92.16 ± 0.56	99.58 ± 0.05	86.39 ± 1.37	1093.80 ± 3.60
BART Fine-Tuned Dropout CNN BiLSTM Linear	98.86 ± 0.04	98.86 ± 0.04	55.91 ± 3.71	98.86 ± 0.04	42.25 ± 3.69	98.86 ± 0.04	45.26 ± 4.57	1086.15 ± 2.69

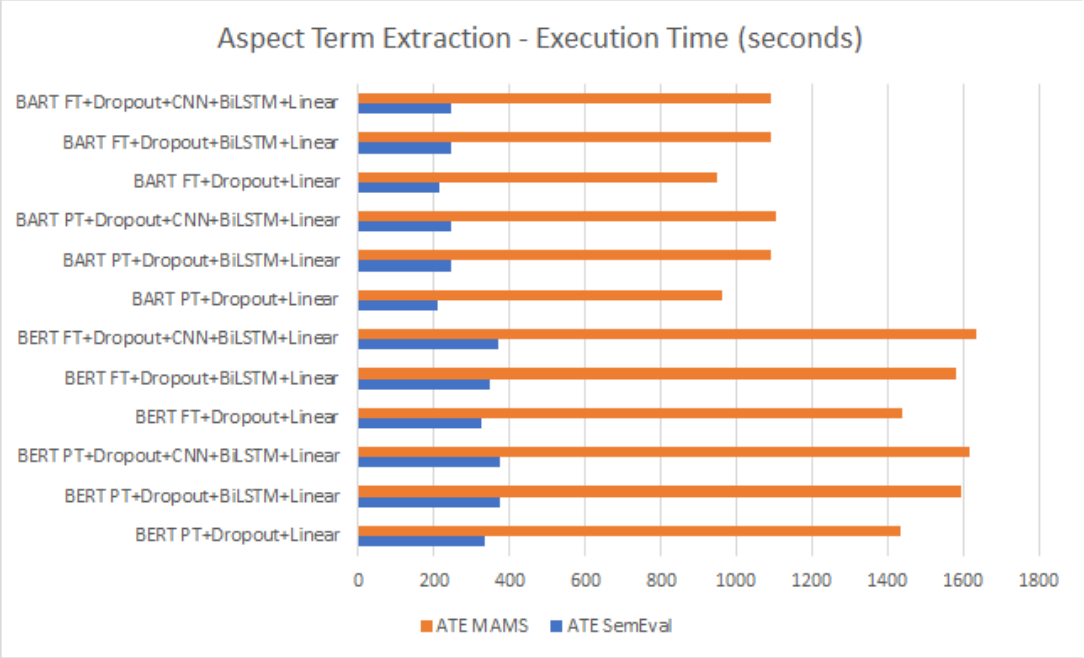

Figure 29: Execution time on Aspect Term Extraction task

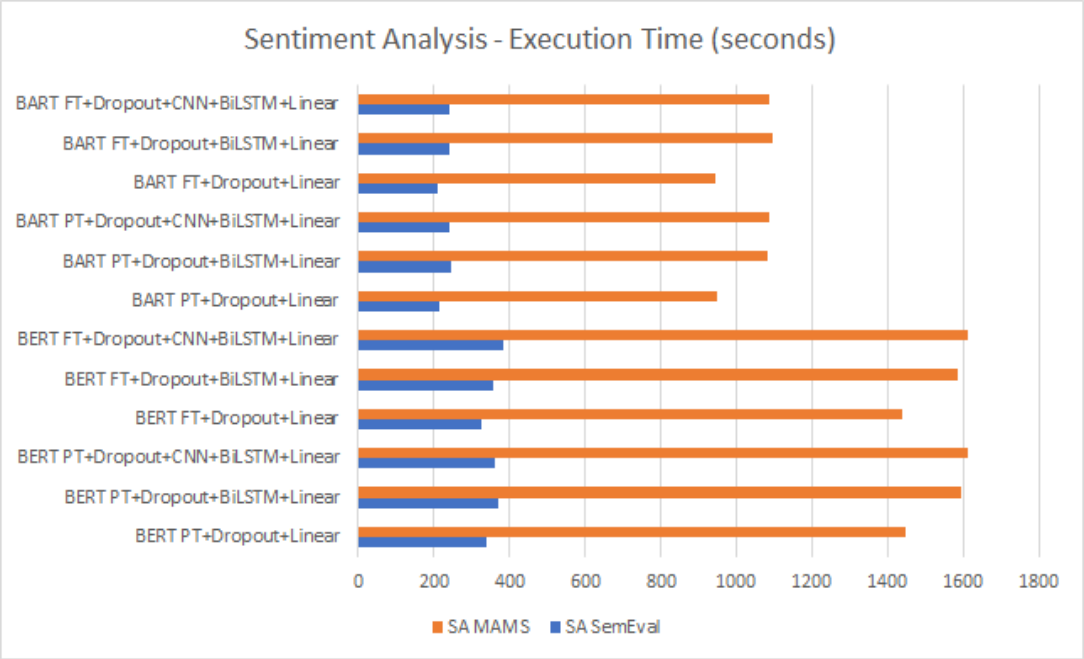

Figure 30: Execution time on Sentiment Analysis task

For a better understanding of the results, compared to previous research, we will go through a comparison for both tasks, on similar versions of the datasets.

For the Aspect Term Extraction problem, we picked as reference the SemEval datasets from 2014 (due to more frequent occurrences in the already existent experiments) and from 2016 (which is an extension of the 2014 dataset). We chose for the aspect term extraction task as baseline models the following models (presented in Table 7):

- **CRF** [5], which uses Conditional Random Fields as state of the art method for sequence labelling. The model is based on the IOB tagging, with an additional POS analysis and probability calculation for multiple variations of the possible labelings for the same input sequence.
- **W+L+D** and **W+L+D+B** [33], based on three, respective four main concepts: **W**ord embeddings, **L**inear context embedding features, **D**ependency context embedding features and **B**aseline feature templates. The models are combining the embeddings with the contextual dependency connections, in order to capture more than the surface information of the input data.
- **CMLA** [30], Coupled Multi-Layer Attentions, that is based on constructing different auxiliary pairs of attentions for Aspect Term Extraction and Opinion Term Extraction that aim to assign an attention score for each token.
- **DE-CNN-CRF** and **DE-CNN** [32], models based on double embeddings: a general purpose embedding and a domain specific embedding, processing the dataset through IOB tagging. The two embeddings are concatenated and redirected into CNN layers.

Table 7: Models comparison for Aspect Term Extraction using the SemEval dataset (Note: **bold** blue text marks our models, while highlighted cells show the best results)

Model	Dataset	F1 score
CRF	SemEval14 Restaurants	79.62
W+L+D	SemEval14 Restaurants	84.31
W+L+D+B	SemEval14 Restaurants	84.97
CMLA	SemEval14 Restaurants	77.80
DE-CNN-CRF	SemEval16 Restaurants	74.10
DE-CNN	SemEval16 Restaurants	74.37
BERT Fine-Tuned Dropout Linear	SemEval16 Restaurants	87.11
BERT Fine-Tuned Dropout BiLSTM Linear	SemEval16 Restaurants	72.04
BART Fine-Tuned Dropout Linear	SemEval16 Restaurants	88.94

Regarding the aspect term sentiment analysis task, we mantained as reference the same SemEval Restaurants datasets, but we switched the comparison onto the accuracy score (presented in Table 8), in order to maintain the same scoring method over all the following models:

- **BERT-single**, **BERT-pair-QA-M**, **BERT-pair-NLI-M**, **BERT-PAIR-QA-B**, **BERT-PAIR-NLI-B** [27] are solving the problem by converting the ABSA problem

to a sentence-pair classification task through question answering and natural language inference methods

- **BERT for ABSA** [10] uses a simple fine-tuned version of BERT for solving out-of-domain aspect classification.
- **DeBERTa** [15] includes a disentangled attention mechanism for simplifying and extracting the semantic and syntactic features from BERT, using a fine-tuned variant of the transformer.

Table 8: Models comparison for Aspect Term Sentiment Analysis using the SemEval dataset (Note: **bold** blue text marks our models, while highlighted cells show the best results)

Model	Dataset	Accuracy
BERT-single	SemEval14 Restaurants	93.3
BERT-pair-QA-M	SemEval14 Restaurants	95.4
BERT-pair-NLI-M	SemEval14 Restaurants	94.4
BERT-pair-QA-B	SemEval14 Restaurants	95.6
BERT-pair-NLI-B	SemEval14 Restaurants	95.1
BERT for ABSA	SemEval16 Restaurants	89.8
DeBERTa	SemEval14 Restaurants	89.46
BERT Pre-Trained Dropout Linear	SemEval16 Restaurants	99.73
BERT Fine-Tuned Dropout Linear	SemEval16 Restaurants	99.79
BERT Fine-Tuned Dropout BiLSTM Linear	SemEval16 Restaurants	99.74

For the MAMS dataset, since most of the current researches are focused on the final result of the ABSA task, it being either Aspect Term Sentiment Analysis or Aspect Category Sentiment Analysis, we picked the most relevant existent models until now and we will compare them with our models on the ATSA task for MAMS Aspect Term Category dataset (Table 9). The models that we used as baseline are the following:

- **HAGNN-BERT** and **HAGNN-GloVe** [1], a heterogeneous graph neural network (a composed network of multiple types of objects, structured as a graph through nodes and edges) based on three different nodes, related to words, aspect and sentences. The mentioned nodes can interchange information in order to update the embeddings based on the corpora used. HAGNN-BERT makes use of the BERT transformer in order to extract the representations of the input sentences, while HAGNN-GloVe is based on LSTMs and CNNs for the same specific task.
- **CapsNet-BERT** and **CapsNet-BERT-DR** [12] computes the input representations using BERT then feeds the output into the capsule networks, in order to model and

interpret the contextual connections for the aspects. The DR extension of the model is introducing a dynamic routing mechanism [16] for a better inference of the context.

- **RoBERTa-TMM** [31], which came as a reformulation of the ABSA task, having in mind the transformers in order to capturate the direct dependencies between words and sentences towards detecting at the same time the sentiments for each aspect present in the sentence.
- **TransEncAsp+SCAPT** and **BERTAsp+SCAPT** [14], the first one using a randomly initialized transformer while the second makes use of a BERT transformer, both being trained through a context-aware fine tuning procedure
- **RGAT-BERT** [3], a relational graph attention network with syntactic dependency information, incorporating label features into the attention mechanism.

Table 9: Models comparison for Aspect Term Sentiment Analysis using the MAMS dataset (Note: **bold** blue text marks our models, while highlighted cells show the best results)

Model	Dataset	Accuracy
HAGNN-BERT	MAMS ATE	66.92
HAGNN-GloVe	MAMS ATE	72.58
CapsNet-BERT	MAMS ATE	83.39
CapsNet-BERT-DR	MAMS ATE	82.97
RoBERTa-TMM	MAMS ATE	85.64
TransEncAsp+SCAPT	MAMS ATE	80.54
BERTAsp+SCAPT	MAMS ATE	85.63
RGAT-BERT	MAMS ATE	84.52
BERT Pre-Trained Dropout Linear	MAMS ATE	99.69
BERT Fine-Tuned Dropout Linear	MAMS ATE	99.87
BERT Fine-Tuned Dropout BiLSTM Linear	MAMS ATE	99.84

5.4 Discussions

The aspect based sentiment analysis was, until the last years, a difficult problem that could give a hard time even to the most performant models. We observed that in the previous research, the most common datasets used are represented by the SemEval problems proposed yearly that, unfortunately, present a big discrepancy in the distribution of the positive, negative and neutral labels. In order to eliminate the doubt of the training phase and the scoring, we opted for running the models on the SemEval dataset for a general comparation with other

papers and on the MAMS dataset in order to observe the behaviour on a larger dataset that contains only multi aspect and multi sentiment labels, thus bringing in the center of the attention an additional challenge.

Contextual analysis and extraction represents a big section of the whole problem. Since the BERT and BART embeddings are contextual, we already had the insurance that the model is going to identify with high accuracy the dependencies between the parts of speech and sentences. But, still, in order to make sure of the evaluation and upgrades that transformers are bringing, we created new models by adding new networks to better evidentiate the contextual dependencies.

The fine-tuning procedures of BERT and BART models specialised the models in the restaurant domain, through iterating and analysing both corpuses, SemEval and MAMS. In order to switch the target of the models to a different domain, the fine-tuning procedure has to be re-done on new domain targeted datasets. Specialising a model on a specific dataset increases the scoring over the learned aspects and sentence structures, with a higher miss detection rate on new unlearned input.

Observing the results presented in Section 5.3, we can confirm that BERT and BART rise up to the state-of-the-art status that they gained over time. We managed to obtain high accuracy with a small number of training epochs on small and medium sized datasets.

In the end, we can not confirm that the ABSA problem has been fully solved on general inputs, due to a lack of public datasets that contain multi domain reviews, but we can say that even with a small sized dataset containing reviews from a specific domain, we can obtain high accuracy results.

6 CONCLUSIONS

Sentiment Analysis was always considered one of the trending topic of the NLP and will remain like this for a long time. Extracting the aspects from online reviews with corresponding polarities can result in a high efficiency in understanding customer problems for different domains, ranging from simple shops and restaurants to online retailers and business to business scenarios.

We managed to implement 12 different models, starting from pre-trained transformers BERT-base and BART-base, in order to solve the Aspect Based Sentiment Analysis problem. We splitted the problem in two sub-problems, for a better understanding and to better understand and observe the behaviour of the models. For each model we executed 10 runs on two datasets, *SemEval16 Task 5 Restaurants* [21] and *Multi Aspect Multi Sentiment* [12]. Each model was trained using only two epochs, with a split of 80% of the dataset for training and 20% for validation for each run.

The best models were represented by fine tuned versions of BERT with additional layers and networks, obtaining a maximum accuracy for the aspect term extraction task of 99.93% on MAMS and 99.86% on SemEval. For the aspect term sentiment analysis task, the max was reached at an accuracy of 99.79% for the SemEval dataset and 99.87% for MAMS. The BART models, with an additional resource requirement status and over 30 million additional trainable parameters compared with BERT, did not manage to overcame his smaller competitor.

In conclusion, we managed to obtain high scoring results for the ABSA task over the restaurant domain with 12 different models over two different datasets, for both sub-problems of aspect term extraction and sentiment analysis. As a future direction, the current solution can be analysed on a new dataset that combines multiple corpora from different domain, in order to understand if the ABSA problem can be generalised and solved through a single, cross-domain, implementation.

BIBLIOGRAPHY

[1] Wenbin An, Feng Tian, Ping Chen, and Qinghua Zheng. Aspect-Based Sentiment Analysis With Heterogeneous Graph Neural Network. *IEEE Transactions on Computational Social Systems*, pages 1–10, 2022.

[2] Lukasz Augustyniak, Tomasz Kajdanowicz, and Przemyslaw Kazienko. Comprehensive analysis of aspect term extraction methods using various text embeddings. 09 2019.

[3] Xuefeng Bai, Pengbo Liu, and Yue Zhang. Investigating Typed Syntactic Dependencies for Targeted Sentiment Classification Using Graph Attention Neural Network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:503–514, 2021.

[4] Himanshu Batra, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Bert-based sentiment analysis: A software engineering perspective. *CoRR*, abs/2106.02581, 2021.

[5] Maryna Chernyshevich. Ihs r&d belarus: Cross-domain extraction of product features using crf. 2014.

[6] Han-Cheol Cho, Naoaki Okazaki, Makoto Miwa, and Jun'Ichi Tsujii. Named entity recognition with multiple segment representations. *Inf. Process. Manage.*, 49(4):954–965, jul 2013.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT 2019*, 01(17):4171–4186, 2019.

[8] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks : the official journal of the International Neural Network Society*, 18:602–10, 2005.

[9] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, July 2017.

[10] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. Aspect-based sentiment analysis using BERT. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196, 09 2019.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[12] Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis. *Neural networks : the official journal of the International Neural Network Society*, pages 6281–6286, 01 2019.

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

[14] Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. Learning Implicit Sentiment in Aspect-based Sentiment Analysis with Supervised Contrastive Pre-Training. pages 246–256, November 2021.

[15] Ricardo Marcondes Marcacini and Emanuel Silva. Aspect-based Sentiment Analysis using BERT with Disentangled Attention. *LatinX in AI at International Conference on Machine Learning 2021*, 2021.

[16] Mason McGill and Pietro Perona. Deciding How to Decide: Dynamic Routing in Artificial Neural Networks. 2017.

[17] Arvind Mohan and Datta Gaitonde. A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks. 04 2018.

[18] Christopher Olah. Understanding LSTM Networks. 2015.

[19] Subarno Pal, Soumadip Ghosh, and Amitava Nag. Sentiment analysis in the light of lstm recurrent neural networks. *International Journal of Synthetic Emotions*, 9:33–39, 01 2018.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *CoRR*, abs/1912.01703, 2019.

[21] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee de clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Zafra, and Gülşen Eryiğit. Semeval-2016 task 5: Aspect based sentiment analysis. pages 19–30, 01 2016.

[22] Alec Radford and Karthik Narasimhan. Improving Language Understanding by Generative Pre-Training. 2018.

[23] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050, 2003.

[24] L. Smith, L. Tanabe, R. Ando, C. Kuo, I-Fang Chung, C. Hsu, Y. Lin, R. Klinger, Christoph Friedrich, K. Ganchev, M. Torii, Hongfang Liu, Barry Haddow, Craig Struble, Richard Povinelli, Andreas Vlachos, William Baumgartner Jr, Lawrence Hunter, B. Carpenter, and W. Wilbur. Overview of biocreative ii gene mention recognition. *Genome Biology*, 9, 09 2008.

[25] Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. 2019.

[26] P Stenetorp, Sampo Pyysalo, Goran Topic, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for nlp-assisted text annotation. *The 3th Conference of the European Chapter of the Association for Computational Linguistics; Avignon, France*, pages 102–107, 04 2012.

[27] Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *CoRR*, abs/1903.09588, 2019.

[28] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective LSTMs for Target-Dependent Sentiment Classification. 2015.

[29] Hilya Tsaniya, Revlita Rosadi, and A Abdullah. Sentiment analysis towards jokowi's government using twitter data with convolutional neural network method. *Journal of Physics: Conference Series*, 1722:012017, 01 2021.

[30] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. page 3316–3322, 2017.

[31] Zhen Wu, Chengcan Ying, Xinyu Dai, Shujian Huang, and Jiajun Chen. Transformer-based Multi-Aspect Modeling for Multi-Aspect Multi-Sentiment Analysis. 2020.

[32] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. Double embeddings and cnn-based sequence labeling for aspect extraction. *CoRR*, abs/1805.04601, 2018.

[33] Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. Unsupervised word and dependency path embeddings for aspect term extraction. page 2979–2985, 2016.