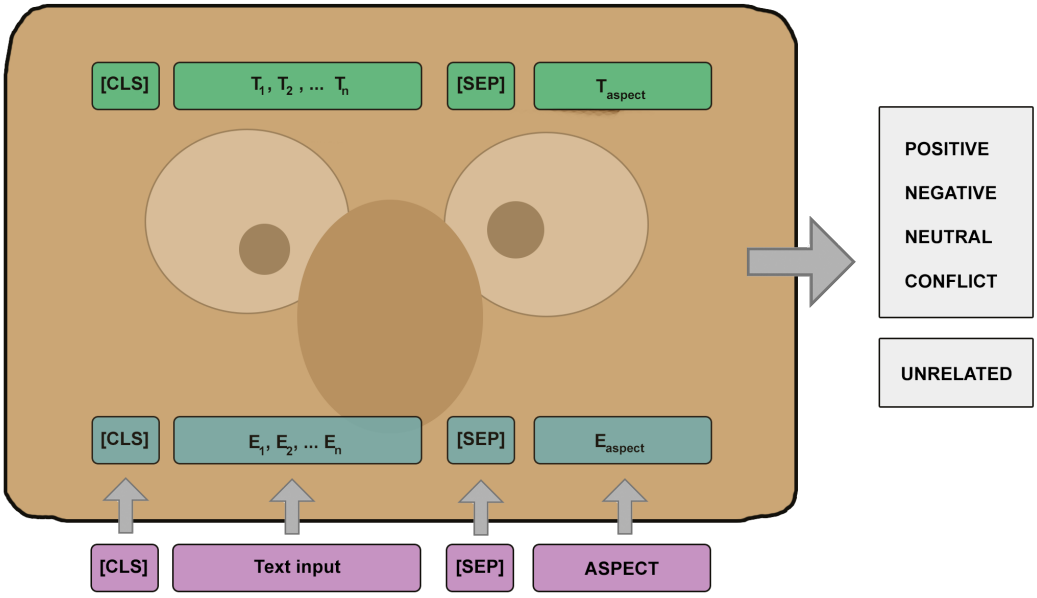
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG



Aspect-Based Sentiment Analysis Using The Pre-trained Language Model BERT

Master's thesis in Computer Science - Algorithms, Languages and Logic

Mickel Hoang
Alija Bihorac

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Aspect-Based Sentiment Analysis Using The Pre-trained Language Model BERT

MICKEL HOANG
OSKAR ALIJA BIHORAC

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2018

Aspect-Based Sentiment Analysis Using The Pre-trained Language Model BERT
MICKEL HOANG
OSKAR ALIJA BIHORAC

Supervisor: Jacobo Rouces Gonzalez, Department of Gothenburg University
Advisor: Sukesh Tedla, ALTEN SWEDEN
Examiner: Richard Johansson, Department of Computer Science and Engineering

Master's Thesis 2019
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: NLP model BERT with ABSA

Typeset in LaTeX
Gothenburg, Sweden 2018

Aspect-Based Sentiment Analysis Using The Pre-trained Language Model BERT
MICKEL HOANG
OSKAR ALIJA BIHORAC
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Sentiment analysis has become popular in both research and business due to the increasing amount of opinionated text generated by Internet users. Sentiment analysis focuses on classifying the overall sentiment of a text, which may not include important information such as different sentiment associated to specific aspects in the text. The more complex task of identifying the sentiment of certain aspects in a text is known as Aspect-Based sentiment analysis (ABSA). This paper show the potential of using the contextual word representations from pre-training language models to solve out-of-domain ABSA by constructing a generic ABSA model using BERT, together with the method of fine-tuning the model to make it learn when aspects are related or unrelated to a text. To our knowledge, no other existing work has been done on out-of-domain ABSA for aspect classification.

Keywords: Pre-training, science, computer science, engineering, thesis, natural language processing, Deep learning, contextual word representation, Aspect-based sentiment analysis, BERT.

Acknowledgements

First and foremost, we would like to thank our academic supervisor Jacobo Rouces Gonzalez for providing us with endless guidance and support throughout the thesis project. He has encouraged our experimentation and ideas but has also given advice and shared interesting thoughts for inspiration. Second, we want to express our gratitude to Sukesh Kumar Tedla, from Alten in the Unbiased team, for giving us the freedom to tackle the thesis problem and he has also given advice and shared interesting thoughts. We would also like to thank Richard Rydell from Alten for hosting this thesis. Finally, we wish to thank our examiner Richard Johansson for reviewing the thesis, giving us feedback and pushing us to do our best.

Mickel Hoang & Oskar Alija Bihorac, Gothenburg, June 2019

Contents

List of Figures

List of Tables

1

Introduction

Sentiment analysis, also known as opinion mining, is a field within Natural Language Processing (NLP) that consists in automatically identifying the sentiment of a text, often in categories like negative, neutral and positive. It has become recently popular in both research and business due to the large and increasing amount of opinionated text from Internet users, such as social media platforms and reviews [1]. Knowing how users feel or think about a certain brand, product, idea or topic is a valuable source of information for companies, organizations and researchers, but it can be a challenging task. Natural language often contains ambiguity and figurative expressions that make the automated extraction of information in general very challenging.

Traditional sentiment analysis focuses on classifying the overall sentiment of a text without specifying what the sentiment is *about*. This may not be enough if the text is simultaneously referring to different topics or entities (also known as *aspects*), possibly expressing different sentiments towards different aspects. Identifying sentiments associated to specific aspects in a text is a more complex task known as aspect-based sentiment analysis (ABSA). ABSA has been a research topic that gained traction during SemEval-2014 Workshop [2], where it was first introduced as Task 4. The task was, given a text about restaurants or laptops, to identify the aspects of the given topic and predict the sentiment polarity for each aspect that was identified.

Similar tasks reappeared in the SemEval-2015 [3] and SemEval-2016 [4] workshops, some of them more generalized. This thesis aims for the Unconstrained evaluation, which are using additional resources such as external training data due to the pre-training of the base model. Furthermore, the thesis evaluates mostly for SemEval-2016, because its ABSA subtasks are evaluating in both Sentence Level reviews and Text Level reviews, as in SemEval-2015, but also provides new test datasets. In addition, submissions from SemEval-2016 use newer techniques and are better than submissions from previous ABSA tasks.

There have been many recent developments in the field of pre-trained NLP models, for example ELMo [5], Universal Language Model Fine-tuning (ULMfit) [6] and BERT [7]. These NLP models are pre-trained on large amounts of unannotated text. Their use has shown to allow better performance with a reduced requirement for labeled data and also much faster training. At SemEval-2016, there were no submissions that used such pre-trained NLP model as a base for the ABSA tasks.

1.1 System overview

The system in this thesis consist of three models. One of the models classifies what the text is about, namely aspects, the other classifies the sentiment for a given text and aspect and the last one classifies both of sentiment and aspect simultaneously. The output of the entire ABSA system is the aspects and the sentiment polarity associated with each of them in the input text.

Aspect classifier. The aspect classifier in this thesis aims to extract the aspects from the given text. Two different machine learning models are implemented and evaluated in this thesis, which is multiple single Sentence classifiers assembled to one model and a Sentence Pair classifier model.
Sentiment classifier. The point of the sentiment classifier is to output the sentiment polarity when given an aspect and text as input. The sentiment classifier in this thesis is constructed as a Sentence Pair classification model.
Combined model. This is a single Sentence Pair Classification model that does the work of both the aspect classifier and the sentiment classifier.

1.2 Aim

This thesis aims to use a pre-trained NLP model to improve performance for aspect-based sentiment analysis, by using the contextual word representation learned from the pre-training of the NLP model. Furthermore, the goal is to make the Aspect Classifier more generic by training the model to also work for out-of-domain aspects.

1.3 Delimitation

This thesis will not consider the Opinion Target Expression (OTE) ABSA subtask of SemEval-2016. Another delimitation of our implementation is the pre-training NLP model, where we choose to use the pre-trained weights from the team behind BERT, and not pre-train a language model ourselves.

1.4 Thesis outline

The remainder of this paper is organized as follows. Chapter 2 will go through the state of the art for the aspect-based sentiment analysis task. It will further explain the two main techniques used in this thesis project, namely the pre-trained NLP model BERT and also Aspect-Based sentiment analysis. This will provide the reader with the necessary background of the techniques and give him the understanding to follow the rest of the thesis. Furthermore, it will also explain how previous work has solved the same task with and without a pre-trained NLP model. Chapter 3 will provide a detailed description of our models, how we use the pre-trained NLP model to downstream and fine-tune it to be able to solve ABSA tasks and how we generate additional data for the generic models. The evaluations of our models and

the metrics we use will be presented in Chapter 4, together with the previous work that scored best followed by a discussion in Chapter 5. Finally, a conclusion is given in Chapter 6.

2
State of the art

In this chapter, we explain the theory and implementation of the techniques and pre-trained NLP models used throughout the thesis. We will further describe the key concepts behind the State of the art results together with the previous works.

Section 2.2 will cover the pre-trained model used in the thesis project, which has achieved State of the art in several NLP-task, together with the model architecture and the key features of the model. Thereafter, Section 2.3 will explain the task ABSA from SemEval. Previous work that reached State of the art with and without a pre-trained model will be briefly described in Section 2.4 and Section 2.5.

2.1 Pre-training Tasks

The classic approach for solving machine learning task is to train a model from scratch with the training data for the specific task. NLP is a diversified research field that contains many distinct tasks which have small sets of human-labeled training data. There has been proven that a large amount of training data has shown to increase the performance of deep learning models, which can be seen in the computer vision field with ImageNet [8]. The same concept can be applied to deep learning NLP models. The development of a general-purpose language model uses a large amount of annotated text, which is called pre-training, and the general purpose for the language model is to learn the contextual representation of words.

2.1.1 Language Model

Programming languages or formal languages can be specified with rules on how to properly use the language. Natural languages differ because they can have ambiguous structures and also possess terms that can be used in ways that make them ambiguous but can still be understood by humans. With words that change their usage, specifying and structuring a language with grammar can be a problem.

The language models are key components when solving NLP problems. They contribute with a form of language understanding, which makes it able to predict the next or missing word in a text input by using the previous context. To be able to uses context to predict words, a form of learning which involves understanding the word occurrence and also the word prediction, are used when training on text data. The language model learns the context by using techniques such as word embedding

which use vectors to represent the words in a vector space. With a large amount of training data, the language model learns more representations of words, depending on the context, and allows similar words to have a similar representation [9].

2.2 BERT

There have been several pre-training model architectures that have performed well in NLP-tasks such as BERT [7], ULMfit [6] and ELMo [5]. This thesis will exclusively make use of the most recent pre-trained model BERT and briefly describe the other pre-training models in the Section 2.2.1.

Pre-trained models aim to learn a generalized language model by extracting features which can be used for other NLP task without the need to retrain the whole model. This form of training is named transfer learning [10] and usually improves the performance of other tasks that is related to the pre-training task.

Bidirectional Encoder Representations from Transformers (BERT), is a pre-trained NLP model that is designed to consider the context of a word from both left and right sides [7]. While the concept is simple, it reaches a new State of the art results for several NLP task such as sentiment analysis and question answering. BERT excels at tasks associated with general language understanding because of the ability to fine-tune the BERT model to a certain task and also because BERT can represent a word context from both left-to-right and right-to-left simultaneously, which means it can extract more context features of a sequence compared to training left and right separately as ELMo [5].

The pre-training of the language model BERT, which trains on both left and right context, is a modified language model using masks called Masked Language Model (MLM) as described in Section 2.2.5.1. The purpose of MLM is to mask a random word in a sentence with a probability of 15%, when the model mask a word it replaces the word with a token [MASK]. The model later tries to predict the masked word by using the context from both left and right of the masked word with the help of transformers. In addition to left and right context extraction using MLM, BERT has an additional key objective which differs from previous works, namely next-sentence prediction, as explained in Section 2.2.5.2. This makes BERT improve on the semantic understanding, which fine-tuned tasks such as question answering and natural language inference use.

BERT is made as a general model that easily can be fine-tuned by adding an additional output layer on top of the Transformer, which is an attention based [11] seq2seq model [12]. The added layer does not need to be trained on a lot of parameters, due to the pre-training and thus only needs minimal training data for downstream tasks. The downstream tasks can either be sequence-level or token-level and are visualized in Figure 2.1.

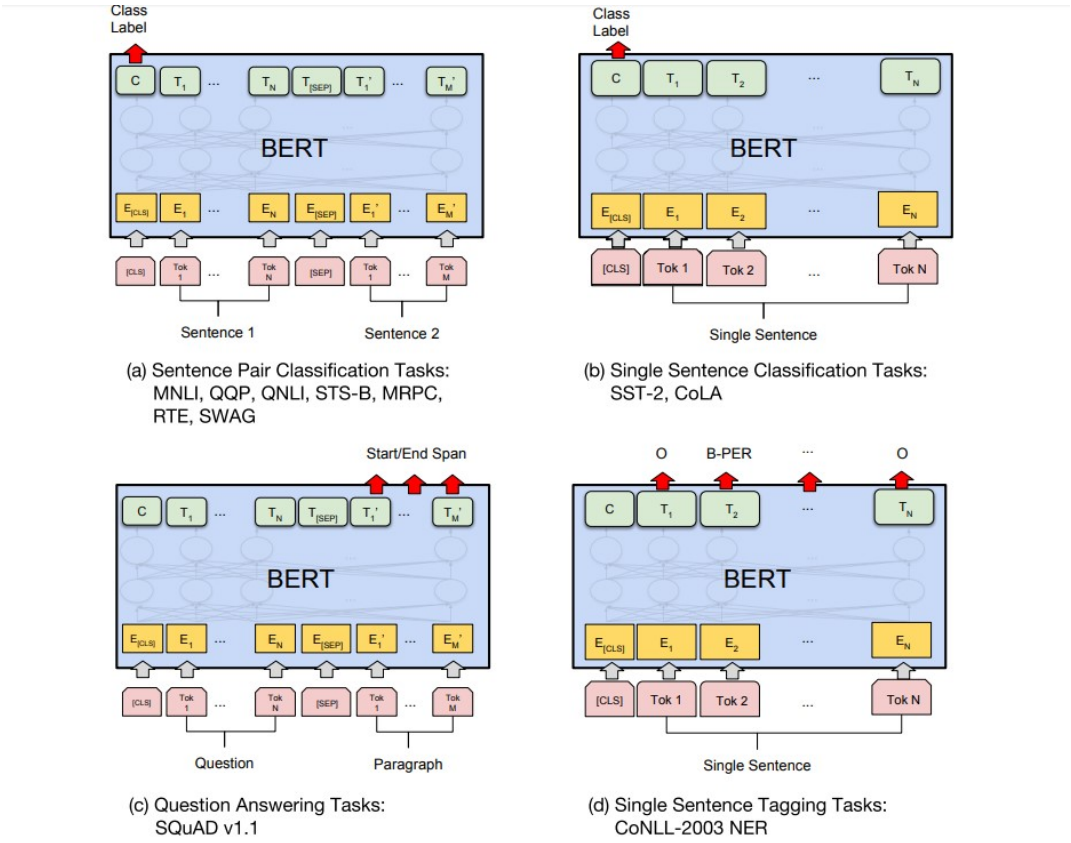


Figure 3: Our task specific models are formed by incorporating BERT with one additional output layer, so a minimal number of parameters need to be learned from scratch. Among the tasks, (a) and (b) are sequence-level tasks while (c) and (d) are token-level tasks. In the figure, E represents the input embedding, T_i represents the contextual representation of token i, [CLS] is the special symbol for classification output, and [SEP] is the special symbol to separate non-consecutive token sequences.

Figure 2.1: Task specific models from original paper [7].

2.2.1 Previous work

Transfer learning has made a big impact in the machine learning field computer vision. Machine learning models that apply the task of computer vision, such as object detection, classification and segmentation, rarely train from scratch. Instead, they utilize transfer learning from pre-trained models that have trained on big datasets such as ImageNet [8]. In the NLP field, transfer learning is important because of the vast amount of training data needed for language understanding. The base models that has trained on large annotated corpus have more word representation and contextual features which can be transferred to other models for task that are related to the tasks used when training on the large corpus, given that the objectives used for the pre-training on the large corpus is extracting general features which can be adapted to other domains than just the base model [6].

In the NLP field, the deep learning models require large datasets because the models are trained from scratch and are task-specific. One of the first breakthroughs of a general language model to address an effective transfer learning method that can be applied to many NLP-task is the Universal Language Model ULMFiT [6]. The concept was to first train the language model on the general-domain corpus to extract general features of the language in different layers. For the fine-tuning of the full language model for a specific task, the original paper proposed the use of discriminative fine-tuning and slanted triangular learning rates to get task-specific features.

BERT is the first deeply bidirectional and unsupervised language representation model, which can be seen in Figure 2.2 that compares the deep architecture of BERT to other architectures such as ELMo. There have been several other pre-trained NLP models before BERT that also uses bidirectional unsupervised learning, one of them is ELMo [5], which also focuses on contextualized word representations. The word embeddings ELMo generates are produced by using a Recurrent Neural Network (RNN) named Long Short-Term Memory (LSTM) [13] to train left-to-right and right-to-left independently and later concatenates the word representation, which Figure 2.2 describes. BERT does not utilize LSTM to get the word context features, but instead uses Transformers [11], which is attention based mechanisms that are not based on recurrence and will be further described in Section 2.2.3.

RNN models [14] generate sequences of hidden states as a function of the previous state combined with the input for the current input position. Essentially, for every recursion, the model has to update the weight for every layer in the architecture to minimize the error, which the network does by propagating back through the layers. When solving for a task, it is sometimes enough to just look at the recent information gathered, but other times the network needs more information further back in the input sequence, which RNNs are not able to provide with, due to the information have been overridden. The RNN has a problem with Long-Term Dependencies, and is thereof not able to process very large input and store all information about the past inputs [15] which is a problem known as vanishing gradients [16].

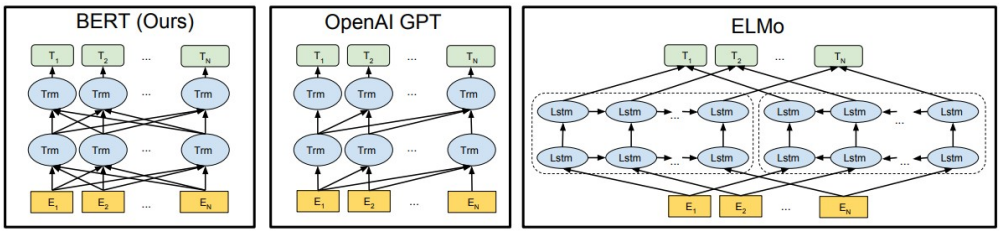


Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

Figure 2.2: Deeply bidirectional model architecture compared to ELMo bidirectional model architecture from the original paper [7].

2.2.2 Input Representaion

The text input for the BERT model is first processed through a method called word-piece tokenization [17], which is a technique to represent words as tokens instead of strings. With this, we get a set of tokens, where each represents a word. There are also two specialized tokens that get added to the set of tokens, and these are classifier token [CLS], which is added to the beginning of the set, and separation token [SEP] which signifies the end of a sentence. If BERT is used to compare two sets of sentences, these sentences will be separated with a [SEP] token, which can be seen in Figure 2.3. This set of tokens is later processed through three different embedding layers with the same dimensions that later get summed together, and then passed to the encoder layer:

- Token Embedding Layer:

  In this embedding, each token in the input will be mapped to a high dimensional vector representation of the given token.

- Segment Embedding Layer:

  As one of the uses in BERT is to be able to find relations between pairs of sentences, this layer is used to separate these sentences. This layer has only two representations: 0 for tokens that belong to the first sentence, and 1 for the tokens that belong to the second sentence.

- Position Embedding Layer:

  As BERT uses Transformers, a position embedding is needed to capture the sequential context of the tokens. BERT has learned the position embedding layer during the pre-training. .

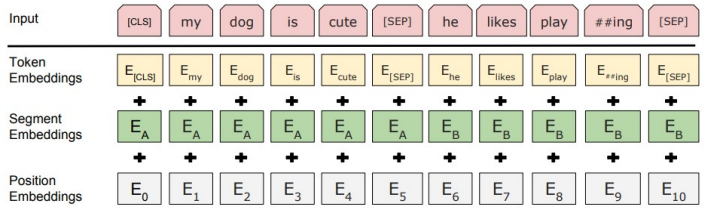


Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Figure 2.3: Input with tokenization and embeddings from original paper [7].

2.2.3 Transformers

Previous works, which established State of the art in sequence modeling, used the common framework sequence-to-sequence (seq2seq) [12], with techniques such as Recurrent neural networks (RNN) [14] and long short-term memory (LSTM) [18]. Seq2seq models consist of Encoder-Decoder based architecture, where the encoders map the input sequence into a high dimensional vector that is then used as an input vector to the decoders, which turns the high dimensional vector to an output sequence.

The architecture of Transformers is not based on RNN but on attention-mechanics [11], which decide what parts are important in each computational step. The encoder does not only map the input to a high dimensional space vector, but also uses the important keywords as an additional input to the decoder. This, in turn, improves the decoder because it has additional information such as important sequences and which keywords that gives context to the sentence.

The transformers use self-attention layers instead of recurrent layers because self-attention have shown to compute better in parallelization but mostly due to the ability to connect all layer positions with a constant number of computational operation, whereas recurrent layers require a constant number of operations, which makes it faster than RNNs. The image in Figure 2.4 describes the architecture of a Transformer, while the Transformer used in BERT only consists of the input embeddings. The left part is the Encoder and the right part is the Decoder. Both Encoder and Decoder are implemented with components that are stackable, described as N_x.

Attention encoders are composed of two modules, namely a multi-head attention layer and a feed forward neural network, while the decoders are composed of a masked multi-head attention layer, multi-head attention layer and a feed forward layer. The multi-mead attention layer, described in Figure 2.5 from the original paper, are based on the attention function:

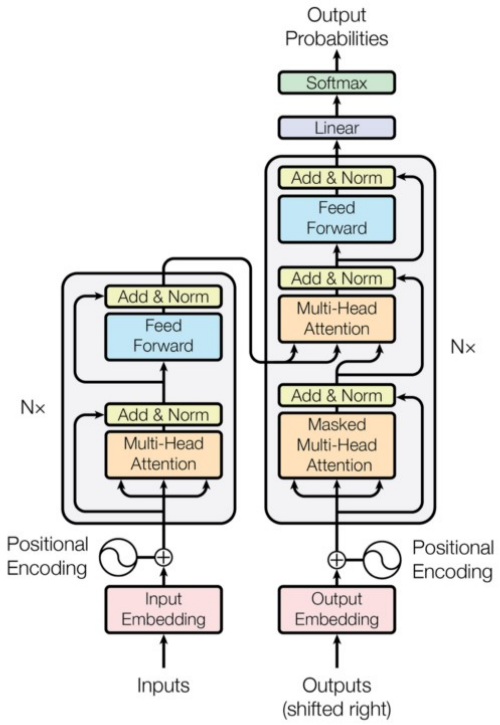


Figure 1: The Transformer - model architecture.

Figure 2.4: Transformers architecture from original paper [11]. The Transformers in BERT uses only the encoder, which is the input embedding (left) part.

$$\text{Attention(Q,K,V)} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

- Q are vector representation of the (query) one word in the sequence
- K are vector representation of all (key) words of (key,value) in the sequence
- V are vector representation of all (values) words of (key,value) in the sequence
- d_k are dimensions of the keys and query

The attention weights are based on how each word in the sequence (Q) is influenced by all other words in the sequence (K). The weights then apply the SoftMax function and lastly apply to all the words in the sequence (V). The system learns different representations of Q, K and V by applying this attention-mechanism repeatedly. The Transformers do not know how the sequences are fed into the model, like the RNN, and instead uses the Encoder input-sequence where the position is taken into account from the Multi-Head Attention module. Each position of the attention modules has different matrices for Q, K and V, which variate depending on the usage of the whole encoder input sequence or parts of the decoder input sequence.

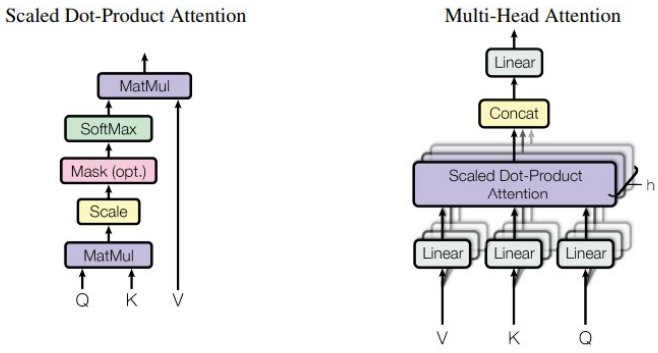


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Figure 2.5: Multi-Head Attention architecture from original paper [11].

2.2.4 Task Specific Models

The original BERT paper pre-trained the model to get word embeddings to make it easier to fine-tune the model for a specific task without having to make a major change in the model architecture and parameters. Usually, only one additional output layer on top of the model was required to make the model more task specific.

2.2.4.1 Single Sentence Classifier

To capture the context of longer text sequences, single word semantic vectors are not enough, instead compositional semantic vector spaces are used instead [19]. Benchmark tasks for single sentence classifier, which is used in BERT, are The Stanford Sentiment Treebank (SST-2) [20] and the Corpus of Linguistic Acceptability (CoLA) [21]. SST-2 is a single sentence classification task for sentiment analysis while the task of CoLA is to determine whether a sentence, in English, is linguistically acceptable or not.

2.2.4.2 Sentence Pair Classifier

The Sentence Pair classification deals with tasks such as the determination if two sentences are equivalent in terms of semantic meaning, the model has to take two text input as described in Section 2.2.2. This kind of task evaluates how good a model is on a comprehensive understanding of natural languages and the ability to do further inference on full sentences [22]. There exist a benchmark that evaluates the natural language understanding on models named General Language Understanding Evaluation (GLUE) [23], which consist of several tasks such as Multi-Genre Natural Language Inference (MNLI) [24], The Semantic Textual Similarity Benchmark (STS-B) [25] and Microsoft Research Paraphrase Corpus (MRPC) [26].

2.2.5 BERT pre-training methods

The methods used in the pre-training of BERT aims to generate generalized features for bidirectional language representation. Furthermore, BERT shows that this approach reduces the need for feature-engineered task-specific architectures by transfer the learned word representation and fine-tune it to other tasks. Generalized methods are crucial to achieve a model that can be utilized in transfer learning and BERT contributes with two pre-training tasks named Masked Language Model, which is a more complex version of the normal language model described in Section 2.1.1, and Next Sentence Prediction. These two tasks will be further explained in Section 2.2.5.1 and Section 2.2.5.2.

2.2.5.1 Masked Language Model

BERT uses a mask token [MASK] to pre-train deep bidirectional representation for the language model. But the difference from the normal conditional language model that trains left-to-right or right-to-left prediction of words, where the predicted word has positioned the end or start of the text sequence, BERT masks a random word in the sequence. The other reason for using a mask token to pre-train is because the standard conditional language model is only able to explicitly train left-to-right or right-to-left due to the words can have the masked word, from left-to-right, unmasked in the right-to-left, in a multilayered context and thus know what the masked word should be.

The original BERT paper masked a word with a probability of 15%, which was distributed as:

- 10% were replaced with a random token
- 10% were left intact
- 80% were replaced with the [MASK] token

The reason for this is because of the conflict that otherwise would arise if the pre-training only made the model predict the mask tokens while the fine-tuning task would not contain any mask tokens. The model would then try to find mask token to predict but not find any in the fine-tuned task, which would result in bad performance. The pre-training would only make the model learn to extract the features from the mask token, which would not be much due to the masking only have a probability of 15% and thus make it converge slower. However, the mask token probability, used in the paper, showed that the language model learned to extract contextual word features instead.

2.2.5.2 Next Sentence Prediction

To understand the relationship between two text sentences, BERT has been pre-trained to predict whether or not there exists a relation between two sentences. Each of these sentences, Sentence A and Sentence B, have their own embeddings, in which we call embedding A and embedding B. An example given from the BERT

paper was the following:

sentence A : [CLS] the man went to the store . [SEP]
sentence B : he bought a gallon of milk . [SEP]
Label : IsNextSentence

sentence A : [CLS] the man went to the store . [SEP]
sentence B : penguins are flightless . [SEP]
Label : IsNotNextSentence

During training, sentence B is the follow up of sentence A in half of the time to be used to predict IsNextSentence label. On the other half of the time, a random sentence is chosen for sentence B to predict IsNotNextSentence label.

2.3 Aspect-Based Sentiment Analysis

Sentiment analysis is a field in NLP and is a task to predict the sentiment in a text. A typical sentiment analysis usually focuses on predicting the overall negative or positive polarity of a text sentence, which generally works for real-world applications if the text sentence only contains one topic or aspect and one sentiment.

Aspect-Based Sentiment Analysis (ABSA) is a more complicated task that focuses on identifying the attributes or aspects of an entity. For example, laptops can have aspects like the battery, screen or touch-pad, etc. The ABSA task would then be able to determine the sentiment for each aspect mentioned in the text. This makes ABSA tasks work on both Sentence- and Text-Level because it can identify when there are several opinions made in the same sentence. An example could be "The actor is so good, but this movie just horrible", which ABSA would output as a positive sentiment about actor but a negative about movie while sentiment analysis would classify as negative and neglect the actor as positive because movie has a more negative sentiment and thus gives the text an overall negative sentiment.

2.3.1 SemEval ABSA task

Aspect-Based Sentiment Analysis was first introduced in SemEval-2014 [2] and provided datasets with annotated reviews about restaurants and laptops. The ABSA task in SemEval-2014 did not contain full reviews until SemEval-2015 [3] and the dataset for SemEval-2016 [4] did not change from 2015 except additional test data.

The goal of the SemEval ABSA task is to identify opinions expressed towards specific aspects of a topic within customer reviews. Especially, given a text review about a certain topic, from the dataset (e.g. laptop, restaurant), the objectives for SemEval-2016, are to address the following:

Aspect Category Classification aims to identify the topic and aspect pair, about an opinion, is expressed in the text. The topic and aspect should be chosen from an already defined set of topic types (e.g. LAPTOP, RESTAURANT, FOOD) and aspects (e.g. PRICE, QUALITY) per domain. The topic can be assigned one or more aspects, depending on the context of the text, in which the topic appears.

Opinion Target Expression (OTE) is the task to extract the linguistic expression, used in the text input that refers to the reviewed entity, for each entity-aspect pair. The OTE is defined with one starting and ending offsets in the sequence. If no entity is explicitly mentioned, the value returned is "NULL".

Sentiment Polarity Classification has the objective to predict a sentiment polarity for each identified topic and aspect pair with the labels positive, negative, neutral, (conflict). The neutral label indicates that a topic and aspect pair are moderately positive or negative and conflict indicates that the sentiment for the aspect have multiple sentiments for example "The spaghetti was good but the scallops were overly salted", which the spaghetti was positive but the scallops, which is also in the spaghetti, is negative.

Subtask 1: Sentence Level is a text-input which only consist of one single sentence. Usually, the sentence level is divided from the text level data samples.

Subtask 2: Text Level are full reviews or text-input. Where several aspects can be mentioned and also several opinions on the aspect can be given in the reviews.

2.4 ABSA without BERT

This section will briefly explain the key concepts for the submissions which ranked highest on the benchmark for SemEval-2016. Furthermore, this thesis will use these submissions as the baseline for the evaluations.

The submissions which took first spot on the SemEval ABSA challenges, for the first submissions of 2016, was using the machine learning algorithm support-vector-machine (SVM) [27][28] or conditional random field classifiers [29], even though deep learning models have shown to perform well in sentiment analysis [30], the deep learning submission for those years ended in a very bad spot. The bad performance on ABSA task does not indicate that deep learning is a bad choice for this kind of task, but rather the choice of model and training tasks was lacking.

2.4.1 NLANGP

The NLANGP [31] placed first on SemEval-2016 subtask 1 and 2, which was the subtask of aspect classification and opinion target expression, used a deep learning approach to extract additional features for their solution. Their aspect-classifier model was implemented by training multiple binary classification models for each

aspect. For the target-extraction-classifier model, they used sequential labeling classifiers, which they trained using conditional random fields.

The aspect-classifier included an additional category as "NIL" that was used if a text did not include any of the aspects that it was trained with. The binary classification model had a softmax as output which was modified with a threshold to solve the problem where a text might contain multiple aspects.

2.4.2 UWB

The UWB model [32] leaned heavily towards using contextualized word representation features for their solution by using techniques such as GloVe [33] and different variations of Bag-of-Words [34].

Both their aspect classifier and sentiment classifier model used a maximum entropy classifier [35] to decide between the categories with a certain threshold to be able to solve for multiple aspects in one text-input. Their solution for text level sentiment classification was to use an algorithm to decide the sentiment by:

1. last seen sentiment polarity for a category: If the polarity frequency for the category is the same as the last polarity seen for the category.
2. polarity label with the highest frequency for the entity: If the polarity frequency for the category is the same as for the entity.
3. Conflicted: If none of the states above is true, which they justified by explaining that the last polarity tends to reflect the final sentiment for a certain aspect category.

2.4.3 GTI

The GTI [36] got the highest score for text level aspect classifier on the restaurant dataset. Their solution was using multiple binary classifiers, one trained for one aspect. Instead of using deep learning techniques, a machine learning linear SVM [28] was combined with word lists. The word lists were generated by extracting all nouns and adjectives from the sentences for each category aspect predicted from the linear SVM. There existed a word list for each aspect in the category (e.g. restaurant had ambience, price, quality, etc). For the text level aspect category classifier, their model was taking the sum of all outputs from sentence level aspect category classifier.

2.4.4 XRCE

The XRCE model [37] scored highest on the sentiment polarity benchmark for the restaurant dataset. Their model is using a syntactic parser as the base to extract information from text (e.g. tokenization and extraction of dependency relations such as subject, object and modifiers). The model further implemented a semantic extraction component on top of the syntactic parser to get semantic information about aspect target and their polarities. The additional information gathered from

the component where the context and polarity scope of the words associating with the lexical semantic features for the different aspect categories (e.g. ambience, price, quality, etc).

2.4.5 IIT-T

The best score on the sentence level sentiment classifier for the laptop dataset was achieved by IIT-TUDA, which used SVM for both the aspect classifier and sentiment classifier [38]. For the aspect classifier, the IIT-T generated an aspect list for each domain to be able to combine all the domain-specific content words. For the sentiment classifier, they used lexical acquisition [39][40] together with a polarity lexicon to make the model learn sentimental word features.

2.4.6 ECNU

ENCU took the first spot on the text level sentiment classifier for the laptop dataset [41]. They used features from linguistics, sentiment lexicon, topic model and word2vec to predict the sentiment of an aspect. Their concept was to use opinion target expression to acquire potential words related to the given aspect as a pending word, which could be words in the same fragment as the aspect. ENCU used several external sentiment lexicons to train their sentiment lexicon features and their word2vec. A logistic regression classifier, together with the generated features, was used to detect the sentiment polarity in a given aspect for text input.

2.4.7 Lsislif

For the out-of-domain ABSA in sentiment classification task on SemEval-2015 Task 12 subtask 2 with the hotel dataset, Lsislif [42] took the first spot and is the baseline for our thesis, due to ABSA in SemEval did not have this kind of task. This model uses a logistic regression model with a weighting schema consisting of positive and negative labels, which they used to extract the different features: lexical, syntactic, semantic and lexicon.

2.5 ABSA with BERT

Pre-trained models, which have been trained on large amounts of data, have over the years been shown to yield very good results on NLP tasks [23]. The use of contextual word representation results in much better performance comparing to non-contextual word representations methods, such as word2vec[9] and bag-of-words [34].

2.5.1 (BERT-PT) BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis

BERT has been pre-trained on a large corpus such as Wikipedia articles, which makes the NLP model lack in the domain of opinion text. The paper of BERT-PT [43] proposes a post-training named review reading comprehension for the BERT

model before the fine-tuning to ABSA tasks. The post-training, which is inspired by the machine reading comprehensions, works as follows:

1. getting a question from a review about a product
2. finding the sequence of tokens (words) that answers the question correctly

3
Models

This chapter presents the different models implemented in this thesis, how the inputs are processed and the generation of additional data for the fine-tuning, to make the models more generic by being able to predict out-of-domain aspects. Some adjustments and experiments had to be performed for the fine-tuning training, which differs from previous work and will be explained further in Section 3.3 and Section 3.2.

The models implemented in this thesis are two aspect classification models, one sentiment polarity classification model and one combined model that consist of both aspect and sentiment classification. Section 3.5 explains one of the two aspect classification models that is trained to determine which aspects are contained in a text input. This model has a fixed amount of aspects from the training dataset, and for each single sentence classifier it decides whether the text is related to a specific aspect or not and thus predicts which aspects are related to the text. The other aspect classification model, explained in Section 3.6, replaces the need of multiple single-sentence classifiers and instead is trained as a more generic model, using sentence pair classification, to find semantic relation between aspect and text input. The model predicts whether there is a relationship between the text and each aspect, which are given as input and thus can predict relation of a text with aspects that are out-of-scope of what the model was trained on. The sentiment polarity classifier, described in Section 3.4, is a classification model that is trained to determine the sentiment labels [positive, negative, neutral, conflict] for a given aspect and text input. Finally, Section 3.7 explains the last model, which is a combination of both the sentiment and aspect classification models, which gives a sentiment if the aspect is related, otherwise it returns the unrelated label.

3.1 Implementation details

We implemented our model with the usage of BERT in pytorch. We used the pre-trained uncased base BERT model, which consists of 12 layers of transformer blocks, 768 hidden layers, 12 self-attention heads and a total of 110M parameters. The implementation, training and testing were performed in Google Colab, with the graphics card they offered, which is a Tesla K80. The training time for each model varied. The training time on each model took roughly 1 second per 40 samples, as such the training time varied from 5 minutes for sentiment classification in the text level restaurant dataset, and 3 hours on the combined model with both datasets in

sentence level. The hyperparameters used across all models are 5 epochs, with the learning rate $2e-5$, batch size of 32, and dropout of 0.1.

3.2 Pre-processing entity and aspect pairs for BERT

The format of the pairs in the SemEval-2016 dataset is originally structured in the form of "ENTITY#ASPECT". To better fit the BERT model when training and to be able to have the pre-trained data in BERT to be useful for these pairs, we formatted it to have a sentence-like structure, such as the pair "FOOD#STYLE_OPTIONS" gets parsed into only "food, style options". This parsed text is what we use as an aspect in the models for this thesis and in the flowcharts within this chapter.

3.3 Data Generation

Pre-trained NLP model requires less labeled training data to teach a model to solve for a specific task such as ABSA. BERT is constructed as a general language model, which made it easier to use the pre-trained word embeddings to fine-tune to a specific task and is one of the main reasons why this work uses BERT. The data provided by SemEval, on ABSA tasks, was not enough to teach the models to learn generic Aspect classification. Additional generated data was required for the model to be able to discern between aspects with a precision that performed better than the baselines.

The datasets used in this thesis is taken from SemEval-2016 - Task 5 [4], where each sample in the dataset contains text which has been annotated by a list of aspects and each of these pairs has also been annotated sentiment polarity, consisting of 'positive', 'neutral', 'negative' or 'conflict'. The annotations to be generated are those which has an aspect that are not related to the subject, for example, the text "The food tasted great!" and the aspect 'restuarant, ambience' does not have any relations. As the datasets have a fixed amount of aspects (e.g. the restuarant dataset has 12 different types of aspects), we can assume that each aspect that has not been annotated for a specific text is unrelated to said text. The aspects which are not related to the text will be added to the list of aspects for the text with an 'unrelated' label instead of a sentiment label. Table 3.1 and Table 4.1 shows the distribution of the original data and our generated data in the training- and test datasets respectively.

	Sentence Level			Text Level		
	Restaurant	Laptop	Both	Restaurant	Laptop	Both
Texts	2000	2500	4500	334	395	729
Unique Aspects	12	81	93	12	81	93
Aspect with Sentiment	2507	2908	5415	1435	2082	3517
Aspect without Sentiment	21493	199592	413085	2573	29913	64280
Total Aspects	24000	202500	418500	4008	31995	67797

Table 3.1: Describes the distribution of data on each training dataset.

3.3.1 Unbalanced Data

The data generated was far more than the original data from SemEval and it caused the model to be biased. This resulted in the models learning more about the features from the generated data and thus performed worse than using only the original data. The solution was to use weighted cost-function that was based on how much bigger the generated data was, compared to the original data, to balance the learning of features from both the original data and the generated data. The datasets from SemEval-2016 are also very unbalanced, and it becomes even more so when the unrelated data is generated, as seen in Aspects without Sentiment compared to Aspects with Sentiment in Table 3.1.

To compensate for the unbalanced data, three different methods were applied:

- Oversampling the polarity labels. Increases training time but too much will likely cause overfitting.
- Undersampling the unrelated data. Lowers training time and stops overfitting of the unrelated label. However, it might reduce performance.
- Weighting each label depending on how frequent the label shows up in the training set, the higher the frequency of a label, the lower the weight of the given sample($\frac{1}{label\_count}$). The weights are then used in the cross-entropy loss function.

3.4 Sentiment Classifier

This is a model for predicting the sentiment of a text, given a specific aspect. It is implemented using the architecture of a sentence pair classification model, where the first input is the text to be evaluated, and the second input is the aspect that the text will be evaluated on. The output of this model will be one of the labels Positive, Negative, Neutral and Conflict, where Conflict means that there are parts of the text where the aspect is viewed as positive and other parts where the aspect is viewed as negative. With this structure, a model is trained for each of the 6 datasets in Table 3.1.

3.5 Multiple Single Sentence Aspect Category Classifier

The system has to classify aspects in the text input, a normal softmax only gives one output, if not considering the use a certain threshold, while a model with multiple single sentence classifier can give multiple outputs.

This model consists of unique submodels for each pre-defined aspect to be used. These submodels are very similar to single sentence classification tasks in the BERT paper, such as the SST-2 Sentiment analysis. It inputs the text and uses the 'related' and 'unrelated' label to predict whether or not the aspect, in which the submodel is trained on, is related to the text or not. This model is trained solely on the text level

restaurant, as it grows linearly in computing time and disk space by every unique aspect in the dataset.

3.5.1 Implementation

In a full model, with the single sentence classification models, every pre-defined aspect that is related to a text will be prepared to send its corresponding aspect to the sentiment model which performs best within the domain the single sentence classification model is trained on. The entire structure is described in Section 3.4, while the others are ignored. It then outputs the sentiment paired with the aspect. The entire structure is visualized in the flowchart on Figure 3.1.

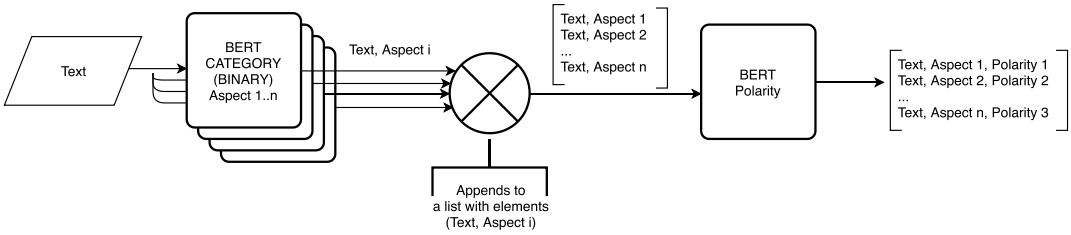


Figure 3.1: Describes entire structure with the single sentence classification model. The text gets passed into a fixed amount of single sentence classification models, which classifies the text as related/unrelated for each model. The related ones gets passed into the sentiment model, with its given aspect.

3.6 Sentence Pair Aspect Category Classifier

This is a model for aspect classification, with the structure of a sentence pair classification task described in Section 2.2.4.2, with the text and aspect as input. This model is used to predict whether or not the aspect input is related to the text or not, with the labels 'related' and 'unrelated'. With the aspect as input, it is possible to handle new aspects out-of-domain of what the model was trained on, as well that it takes less training time, power and space compared to using multiple single sentence classifiers described in Section 3.5. With this structure, a model is trained for each of the 6 datasets in Table 3.1.

3.6.1 Implementation of full model

For implementing the full model, the best performing sentiment classifier and the best performing sentence pair aspect classification classifier is chosen seperately. The sentence pair aspect classification model for all aspects is used to see which aspect the text belongs to. With the use of the labels 'related' and 'unrelated', every aspect that are unrelated to the text will be ignored, while the aspects that is related to a text will be prepared to be sent as an input to the sentiment model, structured as described in Section 3.4. It then predicts the sentiment paired with the aspect. A visual representation of this implementation can be viewed in Figure 3.2

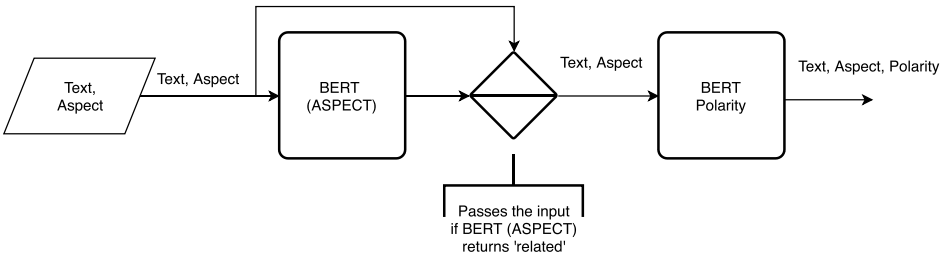


Figure 3.2: Describes the two models in the system. The BERT model to left identifies if the aspect given is related or not. If it is related, the input gets passed to the second model. The second model is the sentiment analysis model for the aspects.

3.7 Combined model

This model is structured as a Multi-class Classification model for predicting both the aspect category and the sentiment using the Sentence Pair classification task described in Section 2.2.4.2. The model also needs both the aspect and the text as input and will return a sentiment label if the aspect is related to the text. Otherwise, it returns the unrelated label. The flowchart of the model can be seen on Figure 3.3. Compared to other models, the entire structure depends on a single BERT model, making it much more lightweight. For each of the 6 datasets in Table 3.1, a combined model is trained.
This model also has the possibility to behave as either an aspect category model by mapping the polarity labels to a 'related' label, or it can behave like a sentiment model by ignoring the value of the 'unrelated' label, it can also act like both aspect and sentiment model simultaneously.

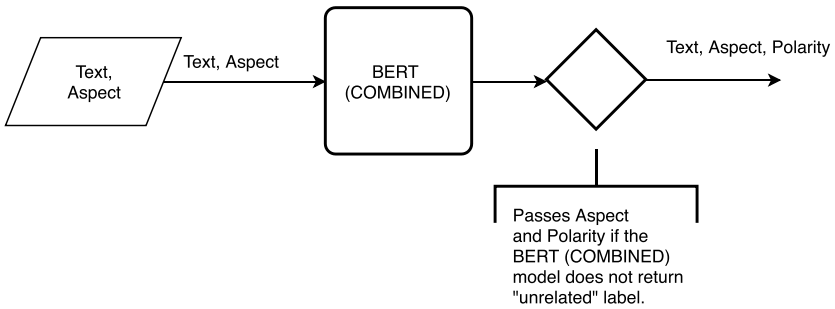


Figure 3.3: Describes a single model. As the combined model directly returns the sentiment labels if it is related, and the 'unrelated' label when it's not, it works as both a categorization and sentiment model.

4
Evaluation

This chapter presents the evaluation of the experiments from the models explained in Chapter 3. Section 4.2 shows the evaluation for aspect category classifiers while Section 4.3 shows the evaluation for sentiment polarity classifiers. For both of these Sections, the evaluation of the Combined model will also be shown. The evaluation and results for each model implemented are presented in Table 4.2a to Table 4.5c, with the previous State of the art models as the baseline. The different models were trained for each dataset (restaurant, laptop and both laptop and restaurant combined) as shown in Table 3.1 and evaluated for both in and out-of-domain.

The evaluation on the SemEval-2016 Task 5, is for the following subtasks:

- aspect categorization (subtask 1 & 2, Slot 1)
- aspect sentiment polarity (subtask 1 & 2, Slot 3)

We evaluate the models on six different datasets, presented in Table 4.1. The restaurant and laptop datasets are from from SemEval-2016 [4], and the hotel dataset is from SemEval-2015 [3]:

- Sentence level & Text level restaurant datasets
- Sentence level & Text level laptop datasets
- Sentence level & Text level hotel datasets

Where sentence level corresponds to subtask 1 and text level corresponds to subtask 2 in SemEval-2016. This makes it possible to examine how well the models compare and perform with different amounts of aspects, out-of-domain and also how well it performs on text with different lengths. The text level hotel dataset had to be generated because the hotel dataset consisted of only a sentence level dataset. This was done by concatenating all the sentences to a full text and label the text with all the aspects from the sentence level inputs.

	Sentence Level			Text Level		
	Restaurant	Laptop	Hotel	Restaurant	Laptop	Hotel
Texts	676	782	226	90	80	30
Unique Aspects	12	81	28	12	81	28
Aspect with Sentiment	859	777	339	404	545	215
Aspect without Sentiment	7253	62565	5989	676	5935	625
Total Aspects	8112	63342	6328	1080	6480	840

Table 4.1: shows the different test datasets and their distribution of data used for evaluations.

The sentence pair aspect category classifier, the sentiment classifier, and the combined classifier, each described in Section 3.6, Section 3.4 and Section 3.7 respectively, have all each trained a model for each dataset described in Table 3.1. This results in 18 models, where each of these models has been tested on every dataset described in Table 4.1. In addition to these models, there is also the model consisting of multiple single sentence aspect category classifiers. This model can only be tested on the dataset, which has been trained on text level Restaurant, due to poor scalability.

4.1 Evaluation Metrics

Accuracy is the simplest metric we use to evaluate our models, it displays the ratio between correct predictions and all predictions.

$$Accuracy = \frac{True\,Positive + True\,Negative}{Total\,Predictions}$$

Precision evaluation metric is used to measure the predicted positive output, the ratio between the correct positive predictions and the total positive predictions is given.

$$Precision = \frac{True\,Positive}{True\,Positive + False\,Positive}$$

Recall is the third metric, it is used to measure how the real positive values are predicted. Out of all the real positive values, a ratio between the predicted positive values and all predictions is given.

$$Recall = \frac{True\,Positive}{True\,Positive + False\,Negative}$$

F1 Score is a harmonic mean of Precision and Recall. It is a useful alternative measure to Accuracy when there's a disproportionate amount of negative labels in the data.

$$F1Score = \frac{2 * Precision * Recall}{Recall + Precision}$$

4.2 Aspect Category Models

In this section, we evaluate how well the aspect classification works with our multiple single sentence aspect classification model, sentence pair aspect classification model and combined model, which are described in Section 3.5, Section 3.6 and Section 3.7 respectively. Each trained in all the different domains and levels described in Table 3.1. All the result tables in this section are ordered by F1 score in descending order.

4.2.1 Sentence Level Datasets

In Table 4.2, the evaluations of the classifiers for aspects in sentence level are shown. The 'Model' column represents which model type the classifiers are. The combined model is defined as 'Combined', and 'Aspect' is a sentence pair aspect category classifier. The other two columns, Domain and Level, is which domain and text type it was trained on, 'Both' indicates that it has been trained on both laptop and restaurant.

- In Table 4.2a, where the classifiers are evaluated on the sentence level restaurant dataset, the aspect classifier which have been trained with the sentence level restaurant dataset shows the best performance, the next best is the combined classifier which also have been trained on the sentence level restaurant dataset. The best performing model has 2 points higher F1 score than the baseline BERT-PT.

- In Table 4.2b, where the classifiers are evaluated on the sentence level laptop dataset, the aspect classifier which have been trained on both datasets in sentence level has the best performance in all categories other than recall, the combined model trained on both datasets shows that it has the best results on recall. The best performing model did not manage to perform better than the baseline NLANGP, it performs slightly worse.

- In Table 4.2c, where the aspect classifiers are evalueted on the hotels dataset in sentence level, there's different classifiers that perfoms best in all the different evaluation metrics used. For the F1 score, the aspect classifier trained on both domains in sentence level performs best, the classifiers with the best recall is the combined model trained on both datasets in sentence level and the aspect classifier trained on text level restaurant dataset. The aspect classifier trained on laptop dataset in text level had the best accuracy and precision.

In out of scope, the restaurant dataset tends to be overly optimistic compared to the other models, as seen on the low precision and accuracy and the high recall. Within all evaluations in Table 4.2, the classifiers trained with the sentence level datasets performs better than the the classifiers trained with the text level datasets. The combined classifier performs worse than the aspect classifier.

Model	Domain	Level	F1	Precision	Recall	Accuracy
Aspect	Restaurant	Sentence	**79.864**	**80.189**	79.542	**96.326**
Combined	Restaurant	Sentence	77.440	75.937	79.004	95.784
Aspect	Both	Sentence	74.423	65.055	86.944	94.526
Combined	Both	Sentence	61.510	46.601	**90.444**	89.632
Aspect	Restaurant	Text	55.478	40.976	85.868	87.376
Combined	Both	Text	53.599	41.964	74.158	88.239
Aspect	Both	Text	52.324	39.558	77.254	87.105
Combined	Restaurant	Text	50.252	36.558	80.349	85.428
Combined	Laptop	Sentence	35.749	30.036	44.145	85.465
Aspect	Laptop	Sentence	34.064	26.385	48.048	82.963
Combined	Laptop	Text	29.178	26.0	33.243	85.219
Aspect	Laptop	Text	26.939	24.347	30.148	85.022
Baseline	Aspect	(BERT-PT)	77.97	-	-	-
Baseline	Aspect	(NLANGP)	73.031	-	-	-

(a) Performance of classifiers in aspect category task with dataset: Restaurant, sentence level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Aspect	Both	Sentence	**51.651**	**40.719**	70.610	**98.412**
Aspect	Laptop	Sentence	49.566	37.582	72.773	98.221
Aspect	Both	Text	38.959	27.521	66.666	97.491
Combined	Both	Sentence	38.729	25.482	**80.661**	96.934
Combined	Laptop	Sentence	38.484	25.890	74.936	97.122
Aspect	Laptop	Text	38.437	27.990	61.323	97.640
Combined	Both	Text	36.378	26.496	58.015	97.562
Combined	Laptop	Text	27.860	18.294	58.396	96.368
Aspect	Restaurant	Sentence	5.7403	2.9980	67.302	73.455
Combined	Restaurant	Sentence	5.2764	2.7471	66.539	71.308
Combined	Restaurant	Text	4.6303	2.3960	68.575	66.075
Aspect	Restaurant	Text	3.8446	1.9711	77.608	53.379
Baseline	Aspect	(NLANGP)	51.937	-	-	-

(b) Performance of classifiers in aspect category task with dataset: Laptop, sentence level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Aspect	Both	Sentence	**34.438**	23.304	65.944	89.111
Combined	Restaurant	Sentence	34.142	22.851	67.492	88.708
Aspect	Laptop	Text	33.788	**28.215**	42.105	**92.843**
Aspect	Restaurant	Sentence	32.301	21.357	66.253	87.956
Aspect	Both	Text	32.049	22.344	56.656	89.581
Combined	Both	Sentence	29.880	18.496	**77.708**	84.183
Combined	Both	Text	29.488	20.481	52.631	89.084
Combined	Laptop	Text	29.176	23.529	38.390	91.917
Combined	Laptop	Sentence	29.087	19.798	54.798	88.412
Aspect	Laptop	Sentence	24.137	15.205	58.513	84.049
Aspect	Restaurant	Text	17.896	10.112	**77.708**	69.078
Combined	Restaurant	Text	17.344	9.9115	69.349	71.334

(c) Performance of classifiers in aspect category task with dataset: Hotel, sentence level

Table 4.2: Evaluations of all aspect classification models on sentence level datasets

4.2.2 Text Level Datasets

In Table 4.3, the evaluations of the classifiers for aspects in text level are shown. The 'Model' column represents which model type the classifiers are. The combined model is defined as 'Combined', 'Aspect' is the sentence pair aspect category classifier, and 'Single' is the Multiple Single Sentence Aspect Classifier. The other two columns, Domain and Level, is which domain and text type it was trained on, 'Both' indicates that it has been trained on both laptop and restaurant.

- In Table 4.3a, where the classifiers are evaluated on the text level restaurant dataset, the aspect classifier trained on the text level restaurant dataset outperforms the baseline model GTI with 1 point. The classifier that achieved the highest recall is the aspect classifier trained on both restaurant and laptop in text level. The highest precision was achieved by the aspect classifier trained with the sentence level restaurant dataset. The accuracy was highest on the multiple single sentence classifiers.

- In Table 4.3b, where the classifiers are evaluated on the text level laptop dataset, the aspect classifier trained with both restaurant and laptop datasets in text level achieved the highest F1 score, beating the baseline model UWB with 4 points. The classifier with the best precision and acuraccy was achieved by the aspect classifier trained on the text level laptop dataset. The model with best recall was achieved by the aspect classifier traiend on the text level restaurant dataset.

- The model that achieved the best in the text level hotels dataset in Table 4.3c was the aspect classifier trained on both laptop and restaurant. The classifier with the best precision and accuracy is the combined classifier trained on text level laptop dataset. The classifier with the best recall is the aspect classifier trained on the text level restaurant dataset.

Within all evaluations in Table 4.2, the classifiers trained with the text level datasets performs better than the classifiers trained with the sentence level datasets. The combined classifier performs worse than the aspect classifier.

Model	Domain	Level	F1	Precision	Recall	Accuracy
Aspect	Restaurant	Text	**85.049**	84.223	85.891	88.703
Aspect	Both	Text	82.948	74.653	**93.316**	85.648
Combined	Both	Text	82.435	78.222	87.128	86.111
Combined	Restaurant	Text	81.562	80.481	82.673	86.018
Aspect	Both	Sentence	78.818	81.866	75.990	84.722
Aspect	Restaurant	Sentence	76.595	**89.700**	66.831	84.722
Combined	Both	Sentence	76.013	69.787	83.460	80.636
Combined	Restaurant	Sentence	70.783	86.715	59.796	81.852
Single	Restaurant	Sentence	70.041	78.484	65.545	**92.043**
Combined	Laptop	Text	67.956	66.430	69.554	75.462
Combined	Laptop	Sentence	65.728	66.066	65.394	74.929
Aspect	Laptop	Sentence	64.194	56.111	75.0	68.703
Aspect	Laptop	Text	62.176	65.217	59.405	72.962
Baseline	Aspect	(GTI)	83.995	-	-	-

(a) Performance of classifiers in aspect category task with dataset: Restaurant, text level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Aspect	Both	Text	**64.298**	60.919	68.073	92.314
Combined	Both	Text	63.902	57.372	72.110	91.717
Aspect	Laptop	Text	63.477	**63.653**	63.302	**92.594**
Aspect	Laptop	Sentence	60.952	57.704	64.587	91.587
Combined	Laptop	Text	58.601	51.236	68.440	90.169
Aspect	Both	Sentence	57.418	60.446	54.678	91.755
Combined	Both	Sentence	56.032	49.147	65.160	89.844
Combined	Laptop	Sentence	55.121	54.511	55.743	90.985
Combined	Restaurant	Sentence	21.621	12.337	87.382	37.086
Aspect	Restaurant	Sentence	21.513	12.256	87.889	34.806
Combined	Restaurant	Text	21.229	12.022	90.642	31.617
Aspect	Restaurant	Text	20.123	11.208	**98.348**	20.630
Baseline	Aspect	(UWB)	60.45	-	-	-

(b) Performance of classifiers in aspect category task with dataset: Laptop, text level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Aspect	Both	Text	**60.765**	48.263	82.005	62.759
Combined	Laptop	Text	59.366	**53.699**	66.371	**68.049**
Combined	Both	Text	58.886	46.218	81.120	60.165
Combined	Both	Sentence	58.762	45.166	84.070	58.506
Aspect	Both	Sentence	57.558	46.389	75.811	60.684
Aspect	Restaurant	Sentence	56.706	44.982	76.696	58.817
Aspect	Restaurant	Text	56.171	41.884	**85.250**	53.215
Combined	Restaurant	Sentence	55.006	46.530	67.256	61.307
Combined	Restaurant	Text	54.726	41.291	81.120	52.800
Aspect	Laptop	Text	53.989	49.152	59.882	64.107
Aspect	Laptop	Sentence	53.265	40.717	76.991	52.489
Combined	Laptop	Sentence	52.319	46.453	59.882	61.618

(c) Performance of classifiers in aspect category task with dataset: Hotel, text level.

Table 4.3: Evaluations of all aspect classification models on text level datasets

4.3 Sentiment Models

In this section, we evaluate how well the sentiment classifications performs with our sentiment model and combined model, which are described in Section 3.4 and Section 3.7 respectively. Each model trained on all the different domains and levels described in Table 3.1. The F1 score measured on the tables in this section is a weighted average of the F1 on each label. All the tables in this section is ordered by Accuracy in descending order.

4.3.1 Sentence Level Datasets

In Table 4.4, the evaluations of the classifiers for sentiment in sentence level are shown. In the tables, the 'Model' column represents which model type it is, 'Combined' is the combined model, 'Sentiment' is the sentiment classifier. The other two columns, Domain and Level, is which domain and text type it was trained on. The datasets in which these models have been tested can be seen in the description of the tables.

- In Table 4.4a, where the classifiers are evaluated on sentence level restaurant dataset, the combined classifier trained on both laptop and restaurant sentence level datasets performs best on all metrics except for precision, the classifier with the best precision is the sentiment classifier trained on both restaurant and laptop datasets in sentence level. The classifier with the best accuracy outperforms the baseline model XRCE with 1 point.

- In Table 4.4b, where the classifiers are evaluated on sentence level laptop dataset, the combined classifier trained on both laptop and restaurant sentence level datasets performs best on all metrics. This model barely outperforms the baseline model IIT-T.

- In Table 4.4c, where the classifiers are evaluated on sentence level hotels dataset, the combined classifier trained on both restaurant and laptop is the best on all metrics. It oupterforms the baseline model lsislif by 4 points.

Overall, the combined classifier performs better than the sentiment classifier.

Model	Domain	Level	F1	Precision	Recall	Accuracy
Combined	Both	Sentence	**89.547**	89.536	**89.771**	**89.771**
Sentiment	Both	Sentence	89.214	**89.605**	89.502	89.502
Combined	Restaurant	Sentence	86.961	86.585	87.752	87.752
Sentiment	Restaurant	Sentence	85.343	85.520	86.137	86.137
Combined	Both	Text	83.318	83.951	83.983	83.983
Sentiment	Both	Text	82.594	82.197	83.176	83.176
Combined	Restaurant	Text	80.994	82.593	81.157	81.157
Sentiment	Laptop	Sentence	81.645	83.808	81.157	81.157
Sentiment	Restaurant	Text	80.335	80.286	80.888	80.888
Combined	Laptop	Sentence	79.625	79.504	80.753	80.753
Sentiment	Laptop	Text	77.535	78.941	76.446	76.446
Combined	Laptop	Text	60.521	62.985	60.026	60.026
Baseline	Sentiment	(XRCE)	-	-	-	88.126

(a) Performance of classifiers in sentiment polarity task with dataset: Restaurant, sentence level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Combined	Both	Sentence	**83.171**	**83.610**	**82.824**	**82.824**
Sentiment	Laptop	Sentence	82.676	82.964	82.603	82.603
Combined	Laptop	Sentence	82.451	82.500	82.442	82.442
Sentiment	Both	Sentence	80.936	80.743	81.226	81.226
Combined	Restaurant	Sentence	77.086	75.696	79.007	79.007
Sentiment	Restaurant	Sentence	75.467	76.246	76.971	76.971
Combined	Both	Text	76.187	76.099	76.717	76.717
Sentiment	Restaurant	Text	74.403	74.378	75.594	75.594
Sentiment	Both	Text	75.509	76.428	75.219	75.219
Sentiment	Laptop	Text	76.377	79.491	74.593	74.593
Combined	Laptop	Text	72.958	74.282	72.391	72.391
Combined	Restaurant	Text	65.361	70.480	69.211	69.211
Baseline	Sentiment	(IIT-T)	-	-	-	82.772

(b) Performance of classifiers in sentiment polarity task with dataset: Laptop, sentence level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Combined	Both	Sentence	**89.997**	**90.953**	**89.473**	**89.473**
Sentiment	Both	Sentence	89.083	89.377	88.854	88.854
Sentiment	Restaurant	Sentence	86.955	86.852	87.306	87.306
Combined	Laptop	Sentence	86.164	86.082	86.996	86.996
Sentiment	Laptop	Sentence	86.526	87.050	86.377	86.377
Combined	Restaurant	Sentence	85.487	85.001	86.068	86.068
Combined	Both	Text	84.156	84.191	84.210	84.210
Sentiment	Restaurant	Text	81.761	81.790	81.733	81.733
Sentiment	Both	Text	80.967	81.611	81.114	81.114
Combined	Restaurant	Text	79.516	81.267	78.328	78.328
Sentiment	Laptop	Text	78.440	83.407	75.232	75.232
Combined	Laptop	Text	73.332	73.048	73.684	73.684
Baseline	Sentiment	(lsislif)	-	-	-	85.840

(c) Performance of classifiers in sentiment polarity task with dataset: Hotel, sentence level

Table 4.4: Evaluations of all sentiment classification models on sentence level datasets

4.3.2 Text Level Datasets

In Table 4.5, the evaluations of the classifiers for sentiment in text level are shown. In the tables, the 'Model' column represents which model type it is, 'Combined' is the combined model, 'Sentiment' is the sentiment classifier. The other two columns, Domain and Level, is which domain and text type it was trained on. The datasets in which these models have been tested can be seen in the description of the tables.

- In Table 4.5a, where the performance of the sentiment classifiers on the restaurant dataset in text level is measured, the classifier which performs best in text level restaurant dataset is the combined classifier trained on both restaurant and laptop datasets in text level. This classifier outperforms the baseline classifier UWB by 5 points.

- In Table 4.5b, which the performance of the sentiment classifiers on the laptop dataset in text level is measured, the combined classifier trained with both restaurant and laptop datasets in text level performed the best on all measurements. It performed better than the baseline model ECNU by 3 points.

- In Table 4.5c, where the performance of the sentiment classifiers on the hotels dataset in text level is measured, the combined model trained on both datasets performs best on all used measurements.

For out of scope, the classifiers trained on sentence level performs better than the ones trained on text level. Overall, the combined classifier performs better than the sentiment classifier.

Model	Domain	Level	F1	Precision	Recall	Accuracy
Combined	Both	Sentence	**86.255**	**86.188**	**87.531**	**87.531**
Combined	Restaurant	Sentence	84.094	81.813	87.022	87.022
Combined	Both	Text	84.729	84.102	86.633	86.633
Sentiment	Restaurant	Sentence	83.383	81.039	86.259	86.259
Sentiment	Both	Text	82.514	81.252	84.405	84.405
Combined	Restaurant	Text	81.244	80.903	82.673	82.673
Sentiment	Restaurant	Text	80.364	78.832	82.673	82.673
Combined	Laptop	Sentence	80.439	79.857	82.442	82.442
Sentiment	Both	Sentence	79.396	77.652	81.424	81.424
Sentiment	Laptop	Text	79.294	77.698	81.188	81.188
Sentiment	Laptop	Sentence	78.722	79.022	80.152	80.152
Combined	Laptop	Text	73.690	70.518	77.227	77.227
Baseline	Sentiment	(UWB)	-	-	-	81.931

(a) Performance of classifiers in sentiment polarity task with dataset: Restaurant, text level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Combined	Both	Sentence	**79.355**	**80.758**	**78.719**	**78.719**
Combined	Restaurant	Sentence	75.645	73.449	78.154	78.154
Sentiment	Both	Sentence	77.130	76.650	77.777	77.777
Sentiment	Laptop	Sentence	76.220	75.553	77.401	77.401
Combined	Laptop	Text	75.148	74.386	76.697	76.697
Combined	Laptop	Sentence	75.435	74.623	76.647	76.647
Combined	Both	Text	74.638	73.302	76.146	76.146
Sentiment	Restaurant	Sentence	73.414	71.258	75.706	75.706
Sentiment	Both	Text	75.132	76.170	74.678	74.678
Sentiment	Restaurant	Text	71.708	69.870	73.944	73.944
Sentiment	Laptop	Text	74.279	74.728	73.944	73.944
Combined	Restaurant	Text	69.151	69.053	71.926	71.926
Baseline	Sentiment	(ECNU)	-	-	-	75.046

(b) Performance of classifiers in sentiment polarity task with dataset: Laptop, text level

Model	Domain	Level	F1	Precision	Recall	Accuracy
Combined	Both	Sentence	**86.906**	**86.502**	**87.315**	**87.315**
Combined	Restaurant	Sentence	85.477	84.103	87.315	87.315
Combined	Both	Text	85.379	84.855	86.430	86.430
Combined	Laptop	Text	82.974	81.562	84.660	84.660
Sentiment	Both	Sentence	82.532	81.580	83.775	83.775
Combined	Laptop	Sentence	82.290	81.142	83.480	83.480
Sentiment	Restaurant	Sentence	81.924	80.433	83.480	83.480
Sentiment	Restaurant	Text	80.761	82.057	81.415	81.415
Sentiment	Laptop	Sentence	79.959	80.521	80.530	80.530
Combined	Restaurant	Text	78.887	81.546	76.401	76.401
Sentiment	Laptop	Text	78.600	81.600	76.401	76.401
Sentiment	Both	Text	77.194	78.321	76.401	76.401

(c) Performance of classifiers in sentiment polarity task with dataset: Hotel, text level

Table 4.5: Evaluations of all sentiment classification models on text level datasets

4.4 Results

For aspect classification, the text level datasets in Table 4.3 produces better results than the sentence level datasets in Table 4.2, in both of these tables, the aspect classifiers always outperforms the combined classifiers. In out-of-scope evaluations, aspect classification performs better with classifiers which have been trained on datasets with more unique aspects.

For sentiment classification, the combined classifiers always outperformed the sentiment classifiers. In out-of-scope scenarios, the classifiers which have been trained on sentence level datasets outperform the classifiers which have been trained on the text level datasets.

5
Discussion

This chapter analyzes the results of the experiments. Section 5.1 discusses the generation of unrelated data and how it affects the learning of the model to predict on out-of-domain tasks. Section 5.2 is about our thoughts on how the models performed in the evaluations. Lastly, the future work is treated in Section 5.3.

5.1 Out-of-domain

The SemEval-2015 had out-of-domain sentiment classification with the dataset Hotels but did not have any task for out-of-domain aspect classification. As shown in the results Chapter 4. We tried to achieve and evaluate for a generic model, which should not be restricted by the amount of aspects in the training dataset. To achieve this we experimented with the sentence pair classification from BERT, together with the contextual word representation to find semantic similarities between an aspect and a text. To teach a model whether an aspect was related or not to a text, we fine-tuned the model with additional unrelated generated data.

Our out-of-domain implementation performed with good results on the out-of-domain evaluation. In aspect category for hotels in Table 4.3c, which our aspect models have not been introduced to before, the model achieved a slightly higher F1 score than the in-domain baseline for laptop F1 score in Table 4.3b. This shows the potential of using semantic similarities to find features for the relation between aspect and a text input. However, to compare these models more in-depth, a better measurement would be to also look at both precision and recall, as the laptop domain has much more unique aspects, which in turn makes it more likely to predict more false-positives that gives it a lower precision. Unfortunately, the data from the baseline model does not include such information.

5.2 Models

For all the experiments and evaluations done in Chapter 4, we trained the models on each specific dataset and tested in the others. Our expectation was that the model would be able to improve the performance by using a combined dataset (restaurant and laptop) to have more features to use for the aspect classification task. This showed that it was not always the case, which we assume has to do with the difference between the number of unique aspects in the domains. The aspect classifiers seem not to work well at the sentence-level test dataset. We suspect that the reason

for this is due to each sentence not having necessarily have enough information to validate if an aspect is relevant for a text. A sentence-level text input example is “The thing wakes up super fast and is always ready to go.”, which is categorized as "LAPTOP#OPERATION_PERFORMANCE". In an out-of-domain and generalized model, this sentence does not provide the necessary information to make it clear that the aspect is related to the sentence and instead can be applied to a lot of other aspects from other domains.

The combined model is consistently better than the sentiment model in all domains, especially for in-domain where it slightly out-performed the Sentiment model. We believe the reason for this is that the combined model is trained on the vast volume of “unrelated” data, compared to the sentiment model, which allows it to learn to ignore redundant features when predicting the sentiment. However, the combined model performs worse than the aspect model in classifying relevant aspects. We conclude that the reason for this is that the combined model has to find what is “relevant”, where “relevant” for this model is related to the 4 sentiment polarity labels, compared to the aspects model that was trained specifically on whether or not the aspect is relevant to the text. Therefore the combined model has a harder time classifying an aspect as relevant, due to the increased complexity of classifying for additional labels and because of the lack of training data for each of the 4 related labels.

5.3 Future Work

Given additional time, there exist several additional adaptions of our system to make it perform better, which will be described in this section. In the field of NLP in general for the GLUE benchmark, new state-of-the-art solutions are continuously published at a fast pace. We assume there will be new and improved ideas of solving ABSA using pre-trained language models, as we did in this thesis.

One of the future adaptions to our work is to use the BERT-big model, which is a model with more hidden layers and has been trained on a larger corpus than the normal BERT we used in this thesis. In the original BERT papers [7], it has been shown that the big BERT model performs better than the original one.

Opinion target expression (OTE) might work well in BERT, by treating it like a question answering task Figure 2.1 (c), where the aspect target is extracted as a sequence, which is defined with a start and end offset within the input text. The additional fine-tuning of OTE might result in the models learning more features to reach better scores on the evaluations. Some of the previous state of the art described in Section 2.4 and Section 2.5 based their aspect classifiers on the OTE models.

6
Conclusion

The purpose of the thesis was to evaluate a pre-trained NLP model, based on unsupervised training on a large corpus to get contextual word representation features, to solve an NLP task, ABSA, with the dataset provided from SemEval-2016. There exists no ABSA task, from SemEval, that consists of a subtask where the models should be able to classify out-of-domain aspects, even though an out-of-domain test dataset was given (Hotel). The only out-of-domain subtask was to use the model to classify the sentiment polarity and not classify the aspects, as such we did not have a baseline to compare to. We developed a model that was able to classify out-of-domain aspects with the help of a pre-trained language model. The evaluation is based on the rules of SemEval-2016, where the aspect classifiers were measured using F1 metrics and the sentiment classifier was measured using accuracy. The models in this thesis were evaluated on two subtasks consisting of different level of input (Sentence Level, Text Level), three different domains (Laptop, Restaurant, Hotel) and two classifying tasks (Aspects, Sentiment).

We were able to implement a model that can predict the aspect of a text that is outside the domain that the model was trained on, by constructing the aspect classifier model as a sentence pair classification task, together with a method for training the model with generated data that consist of 'related' and 'unrelated' labels. The sentiment classifier was trained with supervised learning using the relation between an aspect and a text to learn when the contextual representation showed a sentiment context, in the same way as the aspect classifier. The models were able to achieve good evaluation score for some out-of-domain aspects in ABSA, surpassing all of the original submissions from SemEval-2016 in their in-domain classifying, with the exception of NLANGP [31] in the aspect category classification for laptop dataset on sentence level. This shows that our generic aspect classifying approach is at least as good as the in-domain trained models.

Four different models, based on sentence pair classification, were implemented and evaluated: one sentiment classifier model, two aspect classifier models and the last one which is a combined model that can be used to classify both aspect and sentiment polarity using only one model. In addition to the four sentence pair classification models, one Single Sentence classifier based model was implemented but is restricted to the aspects in the training data and thus could not be tested for out-of-domain, but was implemented to be compared with the other models on in-domain evaluations.

The results achieved in this thesis indicates that the sentence pair classification models are able to use the pre-trained contextual word representations as features to find relations between a text and an aspect but also between an aspect and sentiment with results that outperformed almost all of the baselines, which consisted of the previous State of the art models. The combined model appeared to perform well with classifying both aspect and sentiment using only one sentence pair classifier. The combined model performed better than the sentiment classifier model, but performed slightly worse on aspect classification compared to the aspect classifier models.

Bibliography

[1] M. V. Mäntylä, D. Graziotin, and M. Kuutila, "The evolution of sentiment analysis—a review of research topics, venues, and top cited papers", *Computer Science Review*, vol. 27, pp. 16 –32, 2018, ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2017.10.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574013717300606.

[2] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "Semeval-2014 task 4: Aspect based sentiment analysis", *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 27–35, Jan. 2014. DOI: 10.3115/v1/S14-2004.

[3] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis", in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 486–495. DOI: 10.18653/v1/S15-2082. [Online]. Available: https://www.aclweb.org/anthology/S15-2082.

[4] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit, "Semeval-2016 task 5 : Aspect based sentiment analysis", eng, in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, 2016, pp. 19–30, ISBN: 978-1-941643-95-2. [Online]. Available: http://www.aclweb.org/anthology/S16-1002.

[5] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. [Online]. Available: https://www.aclweb.org/anthology/N18-1202.

[6] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification", in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 328–339. [Online]. Available: https://www.aclweb.org/anthology/P18-1031.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2018. arXiv: 1810.04805 [cs.CL].

[8] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL].

[10] S. J. Pan and Q. Yang, "A survey on transfer learning", *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2009.191.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks", in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14, Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969173.

[13] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling", in *INTERSPEECH*, 2014, pp. 338–342.

[14] A. Graves, "Generating sequences with recurrent neural networks.", *CoRR*, vol. abs/1308.0850, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1308.html#Graves13.

[15] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *Trans. Neur. Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, ISSN: 1045-9227. DOI: 10.1109/72.279181. [Online]. Available: http://dx.doi.org/10.1109/72.279181.

[16] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks", in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML'13, Atlanta, GA, USA: JMLR.org, 2013, pp. III–1310–III–1318. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042817.3043083.

[17] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, *Google's neural machine translation system: Bridging the gap between human and machine translation*, 2016. arXiv: 1609.08144 [cs.CL].

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, ISSN: 0899-7667. DOI: 10.1162/neco.

1997.9.8.1735. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[19] J. Mitchell and M. Lapata, "Composition in distributional models of semantics", *Cognitive Science*, vol. 34, no. 8, pp. 1388–1429, 2010.

[20] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank", in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170.

[21] A. Warstadt, A. Singh, and S. R. Bowman, *Neural network acceptability judgments*, 2018. arXiv: 1805.12471 [cs.CL].

[22] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data", in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 670–680. DOI: 10.18653/v1/D17-1070. [Online]. Available: https://www.aclweb.org/anthology/D17-1070.

[23] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding", in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. [Online]. Available: https://www.aclweb.org/anthology/W18-5446.

[24] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1112–1122. DOI: 10.18653/v1/N18-1101. [Online]. Available: https://www.aclweb.org/anthology/N18-1101.

[25] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. DOI: 10.18653/v1/S17-2001. [Online]. Available: https://www.aclweb.org/anthology/S17-2001.

[26] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases", in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. [Online]. Available: https://www.aclweb.org/anthology/I05-5002.

[27] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features", in *Proceedings of the 10th European Conference on Machine Learning*, ser. ECML'98, Berlin, Heidelberg: Springer-Verlag, 1998, pp. 137–142, ISBN: 3-540-64417-2, 978-3-540-64417-0. DOI: 10.1007/BFb0026683. [Online]. Available: https://doi.org/10.1007/BFb0026683.

[28] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification", Department of Computer Science, National Taiwan University, Tech. Rep., 2003. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers.html.

[29] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289, ISBN: 1-55860-778-1. [Online]. Available: http://dl.acm.org/citation.cfm?id=645530.655813.

[30] Y. Kim, "Convolutional neural networks for sentence classification", in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. [Online]. Available: https://www.aclweb.org/anthology/D14-1181.

[31] Z. Toh and J. Su, "NLANGP at SemEval-2016 task 5: Improving aspect based sentiment analysis using neural network features", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 282–288. DOI: 10.18653/v1/S16-1045. [Online]. Available: https://www.aclweb.org/anthology/S16-1045.

[32] T. Hercig, T. Brychcín, L. Svoboda, and M. Konkol, "UWB at SemEval-2016 task 5: Aspect based sentiment analysis", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 342–349. DOI: 10.18653/v1/S16-1055. [Online]. Available: https://www.aclweb.org/anthology/S16-1055.

[33] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation", in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. [Online]. Available: https://www.aclweb.org/anthology/D14-1162.

[34] Q. V. Le and T. Mikolov, *Distributed representations of sentences and documents*, 2014. arXiv: 1405.4053 [cs.CL].

[35] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing", *Comput. Linguist.*, vol. 22, no. 1, pp. 39–71, Mar. 1996, ISSN: 0891-2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=234285.234289.

[36] T. Álvarez-López, J. Juncal-Martínez, M. F. Gavilanes, E. Costa-Montenegro, and F. J. González-Castaño, "Gti at semeval-2016 task 5: Svm and crf for aspect detection and unsupervised aspect-based sentiment analysis", in *SemEval@NAACL-HLT*, 2016.

[37] C. Brun, J. Perez, and C. Roux, "XRCE at SemEval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computa-

tional Linguistics, Jun. 2016, pp. 277–281. DOI: 10.18653/v1/S16-1044. [Online]. Available: https://www.aclweb.org/anthology/S16-1044.

[38] A. Kumar, S. Kohail, A. Kumar, A. Ekbal, and C. Biemann, "IIT-TUDA at SemEval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1129–1135. DOI: 10.18653/v1/S16-1174. [Online]. Available: https://www.aclweb.org/anthology/S16-1174.

[39] L. Smith, M. Gasser, L. Gleitman, and B. Landau, "The acquisition of the lexicon", *Language*, vol. 73, p. 160, Mar. 1997. DOI: 10.2307/416600.

[40] C. A. Thompson and R. J. Mooney, "Lexical acquisition: A novel machine learning problem", Artificial Intelligence Lab, University of Texas at Austin, Tech. Rep., 1996. [Online]. Available: http://www.cs.utexas.edu/users/ai-lab/?thompson:tech96.

[41] M. Jiang, Z. Zhang, and M. Lan, "ECNU at SemEval-2016 task 5: Extracting effective features from relevant fragments in sentence for aspect-based sentiment analysis in reviews", in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 361–366. DOI: 10.18653/v1/S16-1058. [Online]. Available: https://www.aclweb.org/anthology/S16-1058.

[42] H. Hamdan, P. Bellot, and F. Bechet, "Lsislif: CRF and logistic regression for opinion target extraction and sentiment polarity analysis", in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 753–758. DOI: 10.18653/v1/S15-2128. [Online]. Available: https://www.aclweb.org/anthology/S15-2128.

[43] H. Xu, B. Liu, L. Shu, and P. S. Yu, *Bert post-training for review reading comprehension and aspect-based sentiment analysis*, 2019. arXiv: 1904.02232 [cs.CL].