Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MASTERTHESIS

Knowledge Augmented Aspect Category Detection for Aspect-based Sentiment Analysis

Kai Martinen

01.12.2019

University of Hamburg
MIN-Faculty
Department of Computer Science
Language Technologies

Degree program	Master of Science - Informatics
Matriculation number	7093172
E-Mail	kai.martinen@studium.uni-hamburg.de
Supervisor	Dr-Ing. Gregor Wiedemann
Examiners	Prof. Dr. Chris Biemann
	Dr-Ing. Gregor Wiedemann

Abstract

With the increasing amount of data and especially text data that is available on the internet, an increasing demand has been crested to analyze this data automatically. For text data in particular we want to know what people talk about and what their opinion on these topics are. Aspect category detection helps us determine which topics come up in a text. Together with sentiment analysis this can support companies to find out how customers think about their products at scale by analyzing for example reviews and adjust them accordingly. This thesis proposes a novel way of approaching aspect category detection by introducing external knowledge from knowledge graphs into the classification process. The external knowledge is integrated into a neural network architecture, called Sentic Attention Network, that is tailored to the problem. This architecture is evaluated with different ways of collecting external knowledge as well as different knowledge sources. The results have shown a that the Sentic Attention Network cannot compete with state-of-the-art solutions. Compared to a model with a similar architecture and no external knowledge, it performs slightly better, when using the right knowledge source and method of collecting it. This means that external knowledge can augment aspect category detection to improve performance. However it depends heavily on the quality of the external knowledge.

Contents

List of Figures

List of Tables

List of Abbreviations

BERT Bidirectional Encoder Representations from Transformers

BFS breadth-first-search

CNN Convolutional Neural Network

CPU Central Processing Unit

DL Description Language

ELMo Embeddings from Language Models

GB Giga Bytes

GLoVe Global Vectors for Word Representation

GPU Graphical Processing Unit

GRU Gated Recurrent Unit

HTTP Hypertext Transfer Protocol

LSTM Long Short-Term Memory

MIT Massachusetts Institute of Technology

MLP multilayer perceptron

MS Microsoft

NLP Natural Language Processing

OMCS Open Mind Common Sense

OOV Out-Of-Vocabulary

OWL Web Ontologie Language

POS Part-of-Speech

RAM Random-Access Memory

RNN Recurrent Neural Network

TF-IDF Term Frequency - Inverse Document Frequency

Chapter 1

Introduction

In their daily life, people produce more and more data every day. A lot of this is done by writing simple text messages, comments, reviews or other forms of text using natural language. Certain natural language processing (NLP) tasks aim to extract information from these forms of text. This information can be an opinion about a certain topic, a distribution over all topics mentioned in a text or extracting manifestations of certain general topics.

Finding out which topics are written about in a text and how the writer views these topics is the task of aspect-based sentiment analysis. It basically extracts opinions from a given text. Therefore, it is also referred to as opinion mining. An opinion consists of an aspect, meaning a topic occurring in the text, and a corresponding sentiment value. This value indicates whether a topic has a positive or negative connotation in the text. Aspect category detection is one of the subtasks that have to be performed in doing aspect-based sentiment analysis. Its responsibility is to detect which aspect categories or general topics come up in a text. The most promising approaches in this task have used a neural network architecture [21, 25, 46].

This thesis presents an approach to solve aspect category detection by using external knowledge. External knowledge is hereby defined as an explicit collection of knowledge. This can be found for instance in knowledge graphs. They model knowledge explicitly by using concepts and their relations as nodes and edges. Therefore, it is possible to argument over a knowledge graph by traversing its edges. The approach presented here is inspired by the work of Ma et al. and their Sentic LSTM [21]. Ma et al. integrate external knowledge

in their extended form of the LSTM, a form of recurrent neuron, to do sentiment analysis and a simpler form of aspect category detection. They rely on already extracted aspect terms to map to aspect categories. Aspect terms are the words or phrases in a sentence signaling the presence of an aspect category. The approach presented in this thesis does not rely on already extracted aspect terms. This lifts the complexity from a single-label classification problem to a multi-label classification problem. The approach applies the Sentic LSTM using external knowledge to the multi-label classification problem that is aspect category detection without previously extracted aspect terms.
This raises the question, whether external knowledge can be used efficiently in a neural network architecture using the Sentic LSTM for aspect category detection without previously extracted aspect terms.

1.1 Related works

Aspect-based sentiment analysis is an active field of research [13, 38, 45]. Aspect category detection is one of the three sub-steps necessary to perform it and is therefore constantly improved. Researchers working on new approaches with neural networks often use techniques such as word embeddings, convolutional neural networks, recurrent neural networks or attention mechanisms. As the approach presented in this thesis is an end-to-end solution for aspect category detection, it performs the necessary aspect term extraction implicitly before categorizing the aspects.

1.1.1 Aspect Term Extraction

Aspect term extraction is a task that is related to aspect category detection. It is the first of three subtasks in aspect-based sentiment analysis and is also known as opinion target extraction. Its task is to extract aspect terms in a sentence. As described above aspect terms can be sentences or phrases signaling the presence of an aspect or in the broader sense an aspect category. The first more successful approaches were developed using conditional random fields [18, 8]. These approaches however are inferior to approaches using convolutional or recurrent neural networks [32, 35]. Recently Augustyniak et al. have published a new approach using bidirectional LSTMs [2]. Furthermore, they have conducted extensive research on how word embeddings and other factors

such as character embeddings and using a bidirectional LSTM instead of a normal LSTM influence the performance of their model. They have found that the best performance they achieved measured with the F1-Score, was with a model featuring a bidirectional LSTM, a conditional random field classifier and the GLoVe 42B word embeddings as a pretrained language model. The use of character embeddings has only proven useful if the language model has not captured enough information needed to perform well. The reasons for this could be a high out-of-vocabulary (OOV) ratio for the language model or that the embedding dimension is too small to capture all necessary information. On the other hand, if the language model is expressive enough, character embeddings can also be harmful to the performance.

1.1.2 Aspect Category Detection

There have been several promising approaches to aspect category detection in recent years. Although aspect category detection approaches aim to solve the same task, there is a distinction in how they are able to do this. Here we can distinguish two categories of approaches, that have been commonly seen. As stated in Section 1.1, aspect category detection is the second of three consecutive steps to do aspect-based sentiment analysis. Since the result of the previous step aspect term extraction might be available at this point, the first category of approaches focuses on categorizing aspect categories based on the original text sequence and the already extracted aspect terms. The second category aims to do the same but does not need the extracted aspect terms as an additional input. Therefore, the models in this category must implicitly determine the aspect terms to successfully categorize aspect terms.

Aspect category detection with given aspect terms The Sentic LSTM approach by Ma et al. [21] originally developed for the sentiment analysis part of aspect-based sentiment analysis is a promising way to integrate external knowledge into neural network to solve natural language processing tasks. The Sentic LSTM is an extension of the long short-term memory (LSTM) [15] by Hochreiter et al. and introduces a fourth gate into the LSTM cell, which takes external knowledge as an additional input and adds it to the Sentic LSTMs output if it helps in the downstream task. It has successfully been used in aspect-based sentiment analysis combined with attention mechanisms. They

also applied this architecture to aspect category detection with given aspect terms achieving a micro F1-Score of 73.82 on the SemEval Dataset of 2015 and 77.66 on the SentiHood Dataset.
Another promising approach is Google's BERT model [11], which set a new state of the art in many NLP tasks. Sun et al. [43] have developed a training scheme that trains BERT to be optimized for question answering and natural language inference. They define aspect pairs that consist of an entity and an aspect. From these pairs they generate sentences. This improves the understanding of language and leads to an F1-Score of 87.9 for aspect category detection on the SentiHood data set.

Aspect category detection without given aspect terms The SemEval Competition in 2016 raised the state of the art for this task to an F1-Score of 73.33 on its dataset. This was achieved with the NLANGP system proposed by Toh et al. [46]. It uses convolutional neural networks and general purpose as well as domain specific word embeddings in addition to POS-tags and n-grams to classify aspect categories.
Tamchyna et al. have tried to solve this problem by using multiple stacked LSTM layers in their submission for the SemEval competition in 2016 [44]. They classified the result of the LSTMs with independent binary classifiers for each aspect category. With a F1-Score of 59.3 they have performed slightly worse than the baseline, even though using superior technologies such as learned word embeddings instead of n-grams and a neural network instead of support vector machines.
Yanase et al. [52] have developed an approach using a bidirectional LSTM or GRU followed by a single layered perceptron SoftMax classifier. Since this only models the possibility of each aspect category's presence, they assume a category to be present when its value $y_i > \frac{1}{k}$ with $k = \#categories$. They surpass the baseline of this competition with a F1-Score of 60.14, but they specialize on the most frequent categories adding less frequent ones to a category *OTHERS*. This boosts their F1-Score compared to Tamchyna et al. by omitting less frequent data categories, because these are the categories with a low F1-Score.
Movahedi et al. [25] employed an architecture that has become very popular in recent years. They use a bidirectional recurrent layer, here a Gated Recurrent Unit (GRU), returning a sequence of context enriched vectors. They further pro-

cess these with an attention mechanism or multiple in this case. The novelty of this approach is that they model the architecture in a way that the attention mechanisms attends to different topics of aspects. This creates a topic aware embedding from which aspect categories can be classified independently. This way they aim to fit the model to the multi-label classification problem aspect category detection inherently is. They raised the state of the art with their approach to a F1-Score of 78.38 on the SemEval Dataset of 2016.

Chapter 2

Theoretical Concepts

This thesis evaluates a method on how external knowledge can be incorporated in a neural network architecture to improve aspect category detection without already extracted aspect terms. The standard neural network solutions to aspect category detection typically use several general techniques. Word embeddings [21, 46] and recently even contextualized word embeddings [43, 2] are the foundation, which serve as an input for different types of neural network architectures such as convolutional neural networks (CNN) [19, 31], long short-term memory (LSTM) [2, 21] and attention mechanisms [21, 43].

2.1 Aspect-based Sentiment Analysis

Aspect-based sentiment analysis extends common sentiment analysis to a more detailed level. While sentiment analysis classifies the sentiment of a whole given text, as stated in Chapter 1 aspect-based analysis extracts aspects and calculates sentiment values for them. Sentiment values can either be a category like positive, negative, neutral or a score typically between $[-1; 1]$. The most successful approaches used to involve Support Vector Machines and various preprocessing techniques like stemming and n-grams [13]. For about 5 years now, neural network approaches have started to outperform the traditional ones. Especially approaches applying CNNs and RNNs have produced new state-of-the art results [13]. Aspect-based sentiment analysis can be done on sentence level as well as on review level. Sentence level aspect-based sentiment analysis looks at a sentence as a closed unit and detects aspects and sentiments therein. It is

therefore limited by sentence boundaries and cannot recognize expressed sentiments for an aspect mentioned in a sentence before. On one hand aspect-based sentiment analysis on review level does not have this problem since a review is a contextually closed text unit and usually does not have any dependencies to other reviews. On the other hand, detecting aspects and sentiments for longer sequences can be more challenging since the approach needs to deal with more information at a time.

Service was divine, oysters where a sensual as they come, and the price can't be beat

Figure 2.1: From the *SemEval-2015 Task 12: Aspect Based Sentiment Analysis* restaurant data set

As mentioned in Chapter 1, aspect-based sentiment analysis is divided into three sub-tasks: object target extraction or aspect term extraction, aspect category detection and sentiment polarity detection. These subtasks of aspect-based sentiment analysis build upon each other to extract all necessary information in a sentence. The object target extraction identifies words that can indicate aspects. In the sentence displayed in Figure 2.1 the words *service*, *oyster* and *price* are object targets: In the case of *service* and *price* they directly indicate the aspects SERVICE#GENERAL and RESTAURANT#PRICE. *oyster* on the other hand is not an aspect itself but is mapped to the aspect FOOD#QUALITY, therefore only implying the presence of an aspect. The mapping of these object targets to aspect categories is done by the aspect category detection. In the following, for each aspect category a sentiment is detected from the sentence either as a value or a category.
In neural network approaches, aspect-based sentiment analysis is usually solved either as an end-to-end [37] or as a two-step solution [6, 20, 22]. An end-to-end solution implies having one network executing all 3 steps. A two-step solution hast a network for aspect extraction, which executes object target extraction and aspect category detection and a separate network for sentiment polarity detection.

2.1.1 Aspect Category Detection

Chapter 1 already pointed out the two different ways to solve this task. Depending on which one is used, a corresponding architecture is needed to solve a simple classification problem or a multi-label classification problem.
If the task is to detect aspect categories from a text sequence without given aspect terms, then the challenge is that there can be multiple categories in a text sequence making this a text book multi-label classification problem. A neural network attempting to solve this problem needs to address this by being able to detect each category independently. This means that as multiple categories can be detected at a time the network has to classify the presence of each category on its own. The architecture also needs to have a mechanism that focuses on certain points of the text sequence. As a result, network will learn to recognize aspect-category-related terms and use them to categorize aspects. This would be an implicit form of aspect term extraction.
If extracted aspect terms are already given, the challenge for an architecture is look at the aspect in context of the underlying text sequence. This is often done by pointing out the aspect terms positions in the text sequence to the architecture [21]. Because the aspect terms are already extracted, the architecture can focus on classifying each aspect alone in context of its sentence. This reduces the problem from a multi-label classification to a typical classification problem with only one correct category.

2.2 Neural Networks

Artificial neural networks are a technique in machine learning that have been inspired by biological neurons. As in biological neurons, artificial neurons are triggered by an activation function that determines if a neuron sends a signal or not. Each output is determined by the input and the learned weights on each incoming connection for a neuron. The simplest and most common neural networks are fully connected feed forward networks. These consist of stacked layers with hidden representations called multilayer perceptron (MLP) or in the single case a single layer perceptron. The weights on the incoming connections are trained via backpropagation [36]. Fully connected layers with hidden layers can be used as a non-linear classifier if the activation functions of the hidden layers are non-linear like $tanh$ [40].

Additionally, multiple variations of neural networks such as recurrent neural networks (RNN) and attention mechanisms have been developed for special use cases.

2.2.1 Recurrent Neural Networks

Recurrent Neural Networks consist of recurrent neurons, which have a connection from the output of a neuron to its own input. This way the previous output is factored into the result of the next calculation. Recurrent neurons can also be trained via backpropagation by unrolling them up to the first input. As this is a sequential task, training is slower because the potential for parallelism is limited.
A special form of RNNs is the Long-Short Term Memory (LSTM) [15]. It introduces a gating concept and an internal state as shown in Figure 2.2. The state reflects the acquired knowledge during the training period.

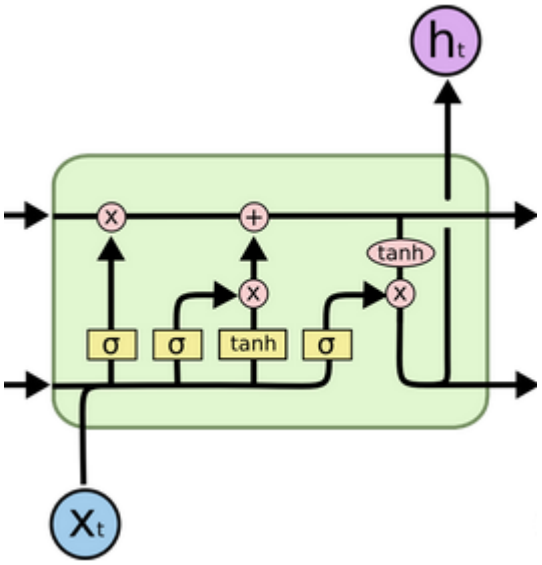


Figure 2.2: Architecture LSTM Cell[1]

A LSTM cell has 3 gates: input, output and forget. The forget gate determines, which information of the current state should be kept, and which should be removed. This is done by creating values between 0 and 1 for each dimension in f_i. The more a value tends to zero the less important this information seems

[1]Source: `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

to be. Correspondingly the closer the value is to 1 the more important is th information.

$$
\begin{aligned}
f_i &= \sigma(W_f[x_i, h_{i-1}] + b_f) \\
I_i &= \sigma(W_I[x_i, h_{i-1}] + b_I) \\
\widetilde{C}_i &= tanh(W_C[x_i, h_{i-1}] + b_C) \\
C_i &= f_i * C_{i-1} + I_i * \widetilde{C}_i \\
o_i &= \sigma(W_o[x_i, h_{i-1}] + b_o) \\
h_i &= o_i * tanh(C_i)
\end{aligned}
$$

Figure 2.3: LSTM: Computation of the three internal gates forget input and output as well as the internal state and the LSTM cell's output

The same principle holds for the input gate. It determines, which information from the input should be considered relevant and therefore be taken into the state C_i by calculating I_i. The actual information taken from the input is learned by W_C and then mapped to values between [-1; 1] to better reflect positive and negative effects on the state. The output gate determines, which information from the state and the input is used for the returned value h_i.
An LSTM layer consists of interconnected single LSTM cells, which pass on their internal state C_i and their last output h_i to the next cell. This is basically the recurrent element of this architecture. Each output of an LSTM cell is determined by the input and the previous cells' state.
Because of their recurrent connection LSTMs have proven to work well with sequences and extracting information from them. That is the main reason they are often used in natural language processing (NLP).

2.2.2 Attention Mechanisms

Attention mechanisms in neural networks have become increasingly popular in the last couple of years, setting a new state-of-the-art in machine translation and other sequence to sequence tasks [11, 47, 10]. In contrast to RNNs, attention-based architectures like transformer [47] are highly parallelizable, because they use a feed forward architecture and do not have to be unrolled, such as recurrent neural networks. Sequential information is processed by encoding the position of an input vector in the sequence and concatenating that encoding to the original

input sequence.
Attention basically works by computing an attention vector indicating, which parts of the original vector are most important. This vector is then added to or multiplied with the original vector highlighting the most important components and therefore information in the original vector. It can be written as:

$$e = M\alpha$$
$$\alpha = softmax(MV)$$

with e as the resulting embedding, M as the matrix attention is employed upon and $V \in \mathbb{R}^{m \times 1}$ as learned weights of the feed forward network. α is called the attention vector with $a_i \in [0, 1] \wedge \sum_{i=0}^{m-1} a_i = 1$ and a_i being the i-th component of α. It signifies, which columns in M are more relevant to the result than others. In NLP columns in a matrix often represent the words in a sentence. The embedding e is therefore made up by the most relevant words for the result of the task the attention mechanism is used for.
Combining attention mechanisms with RNNs has been a popular architecture to solve NLP tasks over recent years. As shown in Section 1.1 several approaches in aspect category detection have followed this general architecture pattern. The same goes for aspect-based sentiment analysis and other tasks [20, 6, 21].

2.3 Word Embeddings

Word embeddings are a manifestation of the vector space model, which represents words as vectors. By creating characteristic embeddings, it becomes easy to measure the difference of words by using the cosine distance or other distance metrics. The idea is that similar words like *big* and *large* each have a vector representation that is very close to one another since both words carry a similar meaning.
Approaches used to be based on word frequencies in documents. Examples for this are TF-IDF [49] or BM25 [34]. These approaches yield good representations when it comes to searching for words in large amounts of documents. As these approaches are based on word occurrences in a chosen number of documents, they model a distribution of them over this set of documents but do not manage to represent the meaning of them. With Word2Vec by Mikolov et al. [23] and especially the Skip-Gram algorithm, which for a given word predicts the t

surrounding words in a sentence, it became possible to capture the context of words using neural networks. The neural network basically learns a vector representation of a word and continues to improve this representation during training. This concept has been extended in FastText by Bojanowski et al. [5] to character n-grams. In addition to the word representations the algorithm learns vector representation for character n-grams. Therefore, contrary to Word2Vec FastText can find word embeddings for words it has not yet seen by using a combination of already known character n-gram representations to create an embedding for the unknown word. Another algorithm to learn general purpose embeddings is GLoVe by Pennington et al. [29]. They use a co-occurrence matrix, which is built at the beginning training and represents the probability of each word occurring with every other word in the trained-on corpus. During training, the model minimizes the distance between the logarithmic co-occurrence probability and the dot product of the two vector representations of the co-occurring words.

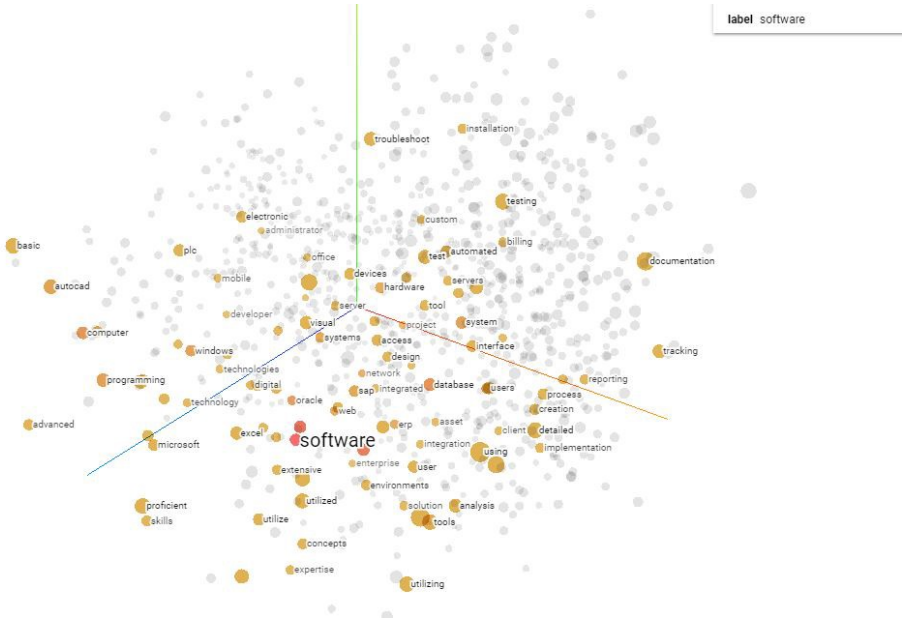


Figure 2.4: Word embeddings visualization[2]

All the above-mentioned models generate general purpose embeddings for words, by having been trained on large datasets featuring text from every available domain. The visualization in Figure 2.4 shows this in detail for software related

[2]Source: `https://medium.com/@aakashchotrani/visualizing-your-own-word-embeddings-using-tensorflow-688b3a7750ee`

words. Words like environment, solution or tools have a short distance to each other since in software development they come up in the same context. Words such as client and server for example have a longer distance, since they have different sometimes even opposing meaning in software development. However, the meaning of words can be dependent on the context. For example, the word *service* has a completely different meaning in the context of restaurants and mobile phones. To address this, Peters et al. developed ELMo [30], an algorithm that learns functions depending on the input sentence to extract contextual embeddings. To do this, ELMo has a general pretrained language model in the background and utilizes forwards and backwards working LSTMs as well as connecting hidden states of these LSTMs to generates its embeddings. Consequently, ELMo does not have a general matrix of embeddings any more but creates representation for each word of a sentences directly with its model.
Devlin et al. have improved contextual word embeddings with BERT [11] considerably. It utilizes a transformer architecture [47] with an encoder decoder architecture. The embeddings are created by extracting embedding layers from the model. As mentioned in the paper, the last four layers concatenated typically yield the best results, although taking the sum of the last four layers is not far off and results in less complexity due to a reduced dimensionality.
Augustyniak et al. [2] have conducted extensive research on the influence of the word embeddings' quality on the performance of an aspect term extraction model. They have found that apart from how the embedding model was trained, the dimensionality and OOV rate had the largest influence. They found that a dimensionality between 200 and 300 captured the most information, measured by the performance of the model. The OOV rate is a good indicator if the language model is suitable to the domain. The higher the OOV rate is, the more often word embeddings are missing, and a null vector must be inserted instead. This reduces the information that can be extracted from a sentence drastically and thus decreases classification rates.

2.4 Knowledge Graphs

Knowledge graphs are essentially ontologies. They represent real world knowledge using entities as nodes and relations between those entities as edges. By using this concept, knowledge graphs can represent every concept that can be

put into words with a triple representation (head entity, relation, tail entity) or (subject, predicate, object) [26].
Knowledge graphs are primarily used in information retrieval to model and extract related information. They only contain positive relations as for example *red is a color* or *an apple is a fruit*. However, negative relations like *an apple is not a vegatable* are not part of a knowledge graph. By containing only positive facts the knowledge graph can be reduced in size [12]. Since a knowledge graph is almost never complete, the absence of a triple must not mean that the concept is negated. This statement could be made under the closed world assumption, where the absence of a fact means that the opposite is true. For a knowledge graph this is not feasible however since it could have simply been forgotten or not yet added to the knowledge graph [12]. This is why typically knowledge graphs operate under the open world assumption, where the negative statements can only be extracted by reasoning over its knowledge, not just by their absence. For example, if we know from the knowledge graph that *an apple is a fruit* and *produce can either be a fruit or a vegetable* then we can deduce that *an apple is not a vegetable*. The graph structure of knowledge graphs makes it easier to reason over the present concepts by using graph inherent structures like transitivity. Reasoning is usually expressed by a using a Description Language (DL) like OWL [16]. The description language can be interpreted by a reasoner that performs the reasoning behind the DL's descriptions. These description languages' performance becomes slower the more expressive they are. For instance, OWL2 the follow up version of OWL has 4 different versions each tuned more towards performance or expressiveness for specific use cases [14].
There are a lot of projects that have collected large amounts of data in knowledge graphs. The most notable of these is the Google Knowledge Graph which contains over 70 billion statements about 500 entities [51]. It is used in Google's search engine to directly answer semantic queries as well as in the Google assistant. Companies such as Microsoft and Baidu have also created knowledge graphs for the same purpose [51]. However, there are also large open-source alternatives on the market such as WikiData, which has reached a respectable size approximately 52 Billion facts. Nevertheless, there is a distinction between different types of knowledge graphs. The aforementioned WikiData or Google Knowledge Graph are universal knowledge graphs modeling all knowledge avail-

able to them. Additionally, there are knowledge graphs such as WordNet [24], which have a narrower purpose such as model language specific data such as language translation, synonyms, etc. These more specific knowledge graphs have the distinct advantage that they are much smaller and thus a lot faster to query.

2.4.1 ConceptNet

ConceptNet is a knowledge graph that aims to represent common sense knowledge. It has been built on top of Open Mind Common Sense (OMCS) Database developed by the Massachusetts Institute of Technology (MIT) and used to only represent common sense concepts based on phrases collected in the OMCS Database. Since its release in 2002, it has been further developed to include other world knowledge and lexical information [42].

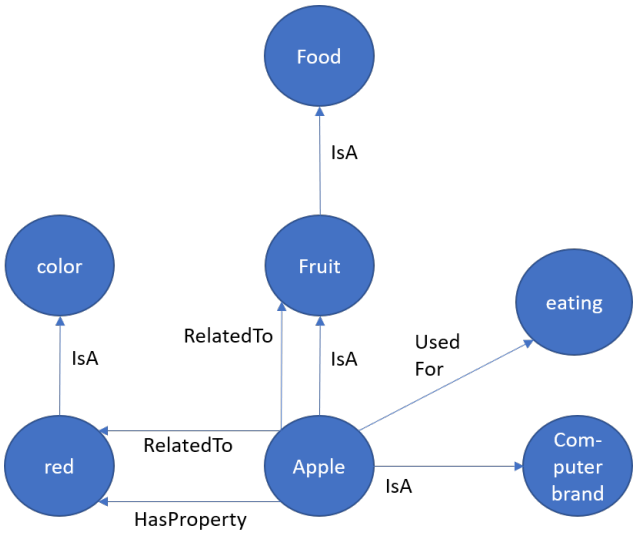


Figure 2.5: Partial Graph in ConceptNet

In contrast to other knowledge graphs such as DBPedia or the Google Knowledge Graph, ConceptNet largely focuses on the common sense meaning of words and does not contain named entities. In addition to the OMCS Database ConceptNet uses additional knowledge provided by projects such as DBPedia, WordNet and OpenCYC. These sources provide ConceptNet with over 8 million nodes and 21 million edges. Figure 2.5 shows a partial graph with selected nodes and edges. We can see here that the relations used in ConceptNet are rather gen-

eral. Although the IsA relation provides a clear hierarchy, focusing on categorical set and subset relations.
ConceptNet is only working with 36 different relations. By restricting the amount of possible relations, the graph is easier to use and query. There are relations such as *Hyponym*, which are imported from WordNet or other sources and are therefore often resource specific. As we see in Figure 2.5 there are also relations such as *RelatedTo*, which are more general and can be used for content-based queries. ConceptNet also preserves the relations from its nodes to nodes of external sources. This is done with a relation to a node containing a link to the external source. This instantly connects it to larger knowledge graphs such as DBPedia. These can have additional knowledge about concepts and words that were out of scope for or not yet inserted into ConceptNet.
ConceptNet also features a score for each edge. This score is especially important for concepts, which have a lot of neighbors with the same relation. For example, the concept *apple* has a lot of neighbors with the relation *RelatedTo*. Sorted by score however we see that the three most important concepts related to *apple* are *fruit*, *red* and *red fruit*. This is an accurate description of an apple. Concepts related to the brand apple such as *mac* are rated relatively low since compared to the fruit the brand is not dominant in the general use of the word in English. In addition to a ranking, these scores can also be used by developers in own applications as weights.
So, the goal of ConceptNet is to present a knowledge graph that is designed to support natural language processing applications with common sense knowledge and language inherent knowledge for multiple languages.

2.4.2 Microsoft Concept Graph

The Microsoft Concept Graph is Microsoft's equivalent to Google's Knowledge Graph an is used to augment their search engine Bing. It is built upon Probase, a general-purpose knowledge graph built by Microsoft crawling the web for all available information [50]. Probase is a universal ontology or knowledge graph that besides modeling the existence of knowledge also reflects the knowledge's probability or reliability. This means each relation or statement has a probability value attached to it representing the reliability of that statement.
It is built by an algorithm that extracts IsA-Pairs from natural language. These pairs can be concepts as well as named entities. Afterwards these pairs are

connected into an ontology. This ontology can then be used to successfully augment a search engine with a semantic search or used to extract the meaning and main point of short texts [50].
The Microsoft Concept Graph combines the Microsoft Concept Tagging model and Probase into a system that provides common sense knowledge as well as named entities. The Microsoft Concept Tagging model maps text inputs into concepts represented in Probase. These concepts can then be used for further queries, for example nearest neighbors or top-k neighbors. The Probase's probability score is an added bonus for developers since it can also be interpreted as measuring a level of connectedness between two concepts. This can be used by developers to interpret their query results.

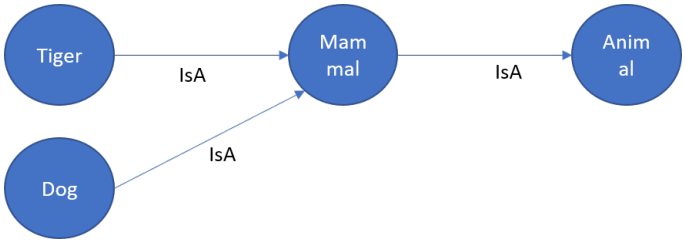


Figure 2.6: Partial Graph in Microsoft Concept Graph

Since Microsoft Concept Graph is restricted to IsA relations the kind of knowledge that can be captured by it, is limited to hierarchical subset relations like a *a mammal is an animal*. This also includes transitive relations such as *a tiger is an animal* if *a tiger is a mammal* as shown on Figure 2.6. It cannot however explicitly capture knowledge of any kind, that this is not mappable by an IsA relation. This includes for example properties of any kind like *a tiger has claws*. This limitation has to be considered when working with Microsoft Concept Graph.

Chapter 3

Approach

External knowledge can augment knowledge-based tasks and direct neural networks in making the right decisions [21]. The question arising however is, do these additional inputs and the information they provide, influence a neural network in its learning. Knowledge augmented aspect category detection in the context of aspect-based sentiment analysis has not yet been studied a lot as mentioned in Section 1.1. Nevertheless, knowledge augmented aspect-based sentiment analysis has, and the results have been promising. As mentioned in Subsection 2.1.1 there are two forms of aspect category detection. The form that this thesis aims to solve is the one without previously detected aspects terms. Meaning here aspect category detection can be defined as performing both object target extraction and aspect category detection. Solving both tasks together with one model is a popular way to approach this task [52, 46].
The neural network models presented in this chapter are inspired by the work of Ma et al. [21], and specifically their Sentic LSTM. Ma et al. have developed a neural network approach for aspect-based sentiment analysis using external knowledge, an attention mechanism and a modified version of the LSTM in their model. With this they augment their information by using SenticNet [7] and classify sentiments in text sequences for given aspect terms as well as aspect categories for given aspect terms. Nevertheless, they only solve the aspect category detection with extracted aspect terms. This thesis applies their Sentic LSTM to aspect category detection without already extracted aspect terms. As explained in Subsection 2.1.1 this makes the problem a lot more difficult to solve and the results cannot be compared.

3.1 Sentic LSTM

The Sentic LSTM is an extension to the basic LSTM Architecture. It features a second input μ and an additional output gate that can be called concept gate. The idea behind the Sentic LSTM is that in addition to the traditional input

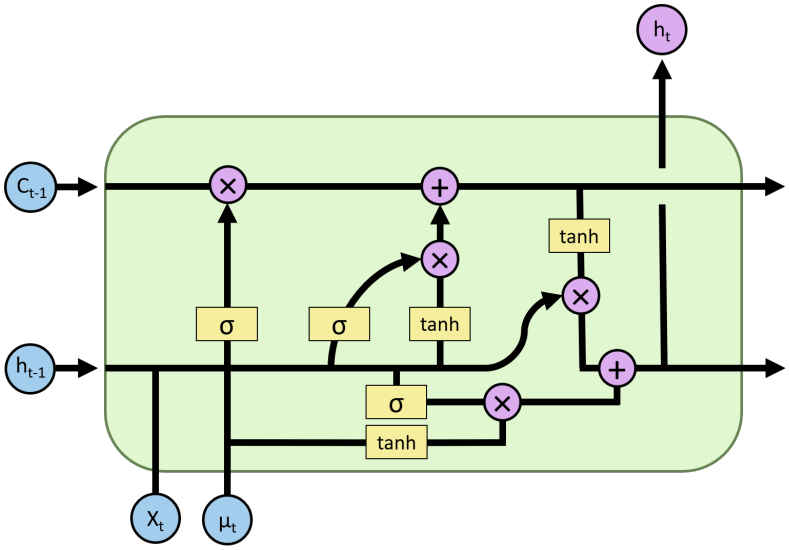


Figure 3.1: Architecture Sentic LSTM Cell

sequence x there is another input sequence μ, whose elements provide additional knowledge for their corresponding elements in the original input sequence x. The additional information passed to the Sentic LSTM by μ does not enter the state C, as it is only supposed to help the LSTM make better decisions about which parts of the input sequence x are important and which are not. The additional information is however added to the output, since it can help to find and augment information that has been found in the original input sequence x. This adaption results in the following changes to the LSTM equations from

Subsection 2.2.1:

$$
\begin{aligned}
f_i &= \sigma(W_f[x_i, h_{i-1}, \mu_i] + b_f) \\
I_i &= \sigma(W_I[x_i, h_{i-1}, \mu_i] + b_I) \\
\widetilde{C}_i &= tanh(W_C[x_i, h_{i-1}] + b_C) \\
C_i &= f_i * C_{i-1} + I_i * \widetilde{C}_i \\
o_i &= \sigma(W_o[x_i, h_{i-1}, \mu_i] + b_o) \\
o_i^c &= \sigma(W_{co}[x_i, h_{i-1}, \mu_i] + b_{co}) \\
h_i &= o_i * tanh(C_i) + o_i^c * tanh(W_c\mu_i)
\end{aligned}
$$

Figure 3.2: Sentic LSTM: Adjusted equations for the forget, input and output gates as well as the forth added knowledge gate and the adjusted output h_i

Input μ_i represents the external knowledge that is fed into the Sentic LSTM. It is added to the input and forget gate and thus plays a factor in which parts of input x_i are added to the state. This is also the case for the output h_i. The additional gate o_i^c determines, which parts from the external knowledge can be used to augment the output to improve the overall result. If there is no helpful information in μ_i, o_i^c would become a null vector reducing the impact of any unhelpful information from μ on h to zero. This is also depicted in Figure 3.1. The main difference to the basic LSTM in Figure 2.2 is shown in the bottom of the picture. This visualizes the fourth gate and the integration of its values into the output of the Sentic LSTM cell.
So, the Sentic LSTM uses the additional information provided by μ only in the cases where it is helpful but not if it would make no difference or might even be harmful.

3.2 Sentic Attention Network

The Sentic Attention Network is a neural network architecture that aims to perform aspect category detection without given aspect terms. It takes two inputs, a sentence and sequence of augmenting information where each element in the sequence augments the word with the same index in the input sequence. The sentence is represented in the vector space model by converting each word

into its corresponding embedding vector. This embedding comes from a previously chosen language model. Figure 3.3 shows that apart from the inputs and

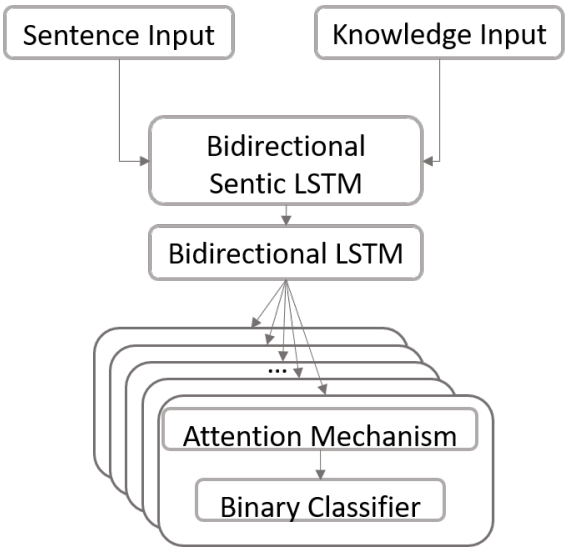


Figure 3.3: Architecture Sentic Attention Network

outputs we have 3 key components present in this architecture:

- Sentic LSTM
- Attention Mechanism
- Classifier

The general architecture of this model obviously corresponds to the properties of aspect category detection without aspect terms. This means that the network needs to be able to classify multiple aspect categories from input text sequences. It does this by mapping the inputs to the right form for the Sentic LSTM. The resulting output sequence is then fed to the attention mechanism, which is followed by an MLP classifier. As it is not uncommon to talk about multiple aspects in a sentence, the network needs to be able to independently classify if an aspect category is present in a sentence or not. This makes classifying aspect categories harder as explained in Subsection 2.1.1. The network manages the independent classification by having independent classification components for each possible category. A classification component consists of an attention mechanism and an attached MLP classifier.

Sentic LSTM The Sentic LSTM as described in Section 3.1 combines external knowledge gathered for a text sequence with that same text sequence. It enriches the information found in this text sequence improving the overall capability to extract its information [21].
In the Sentic Attention Network the Sentic LSTM is used as a Bidirectional Sentic LSTM. The traditional LSTM has a direction in which it processes sequences. This would be left to right (LTR) ascending the indices or right to left (RTL) descending the indices. The result of an LTR-LSTM and an RTL-LSTM will be different, and its quality depends on the language of the written text sequence. A language such as English, which is read left to right will have better results with an LTR-LSTM, while Arabic a right to left language will have better ones with the RTL-LSTM. Nevertheless, studies have found that using both directions for the same text has delivered better results, since not all dependencies in a language follow the read directions [39]. The output of a bidirectional LSTM is therefore defined as

$$H = \begin{bmatrix} \overrightarrow{h_1}\overrightarrow{h_2}\dots\overrightarrow{h_m} \\ \overleftarrow{h_1}\overleftarrow{h_2}\dots\overleftarrow{h_m} \end{bmatrix}$$

which is the concatenation of each the RTL and LTR cell's output. This principle also applies to the Sentic LSTM.
Within the Sentic Attention Network each h1 reflects the information in the sequence coming from left and right enhanced by external knowledge. Other approaches have shown that two to three stacked LSTM layers can enhance performance [44]. Therefore, an LSTM layer is connected to the Sentic LSTM further processing its output. Connecting a second Sentic LSTM layer might not be feasible since the first layer should have already integrated the external knowledge. Integrating it a second time has shown to be harmful to the overall performance.

Attention Mechanism The attention mechanism used here is a self-attention mechanism also employed by Ma et al. and others [21, 17]. It slightly differs from the standard attention mechanism explained in Section 2.2.2 in using two fully connected layers instead of one resulting in:

$$v = H\alpha$$
$$\alpha_i = softmax(W_1 tanh(W_2 h_i))$$

with $W_1 \in \mathbb{R}^{m \times n}; W_2 \in \mathbb{R}^{n \times 1}; m, n \in \mathbb{R}$ and $m > n$, where α is a vector of learned parameters containing weights between 0 and 1 for each h_i in H. Since the parameter is learned from H, it is supposed to reflect the impact of each h_i on the result. The weights are then multiplied with H and the results summed up creating an embedding v of the previously extracted information.
The Sentic LSTM creates context enriched vectors for each word embedding in a text sequence. Since attention mechanisms basically highlight information in a sequence that can be beneficial to the end result, the Sentic Attention Network provides an attention mechanism for each possible category. The beneficial information in this context are the context enriched embeddings of the aspect terms provided by the Sentic LSTM. The task of each attention mechanism is therefore to recognize the aspect terms relevant to its category and highlight them in the embedding of H. In theory this should help the downstream classifier to makes a more certain prediction for each category.

Classifier The classifier is a simple two-layer MLP with a scalar output. There is an independent classifier for each category attached to its attention mechanism as explained in the general architecture. The output value is determined by the $sigmoid$ function which maps its values to the interval $[0; 1]$. Each classifier can be described as

$$y_i = sigmoid(V_1 ReLU(V_2 v))$$

with $V_1 \in \mathbb{R}^{m \times n}; V_2 \in \mathbb{R}^{m \times 1}; m, n \in \mathbb{R}$ and $m > n$ and y is the output vector of the Sentic Attention Network and each $y_i \in [0; 1]$. Since we expect a binary value meaning 1 if an aspect category is present and 0 if it is not, we can round the output values y_i and receive the expected binary values.

3.3 External knowledge

External knowledge in the context of aspect category detection can be defined as every input the neural network gets in addition to the text it is supposed to analyze. Already extracted aspect terms do not count as external knowledge since they were determined by external sources. The questions arising in this context must be, what kind of information can be beneficial to a neural network and what form does this information need to have.

To start with, external knowledge can be every form of knowledge that does not come from the input text itself. One needs to distinguish, however, between knowledge provided by for example domain-specific language models, which have a different distribution of their words than a general-purpose model and an explicit knowledge base. An additional domain-specific language model might help with the quality of the word embedding and thus provide more knowledge about an input text. It does not however represent explicit knowledge. An explicit knowledge base on the other hand and knowledge graphs in particular represent knowledge explicitly and can be queried for this knowledge as described in Section 2.4. Therefore, the external knowledge in this thesis comes from knowledge graphs as they represent explicit knowledge in a convenient way.
The following ways to collect knowledge from knowledge graphs will be used to feed this external knowledge to the model described above.

3.3.1 Weighted Potential Opinion Targets

Aspect category detection in the form described in this chapter performs opinion target extraction in addition to its own task. So, any model solving aspect category detection must consequently implicitly solve opinion target extraction as well. The idea behind this method to gather external knowledge is to go through a sentence and extract all nouns, verbs, adjectives and adverbs as they are the only words that can represent opinion targets. For these words we look at the knowledge base and have a look at the top k related words via a relation named RelatedTo. These words must have a strong connection to the looked-up words. Furthermore, we extract the weights for the connecting edge in the knowledge graph to have a measurement of how much the looked-up word is influence by the found one. This is demonstrated in Figure 3.4. Then we transform the found words into embeddings with the same model that was used to embed words of the text sequence. At the end we take the embeddings and weights for each a word extracted from the sentence and calculate the weighted sum. This gives us a vector that represents the word by concepts most related to it. This embedding should in theory be closer to an embedding of an aspect than using the embeddings of opinion targets and therefore augment the knowledge present in the text. So, for each extracted word from the text follows that the

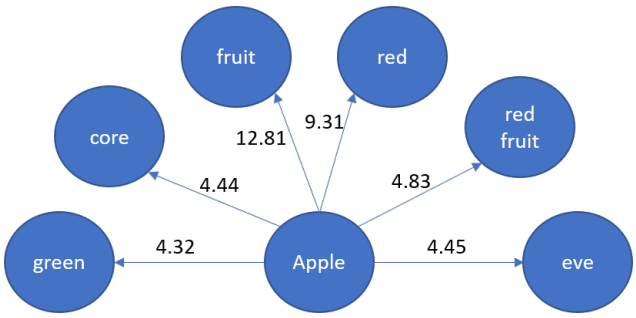


Figure 3.4: potential weighted opinion targets

augmented knowledge vector w is describes as:

$$w = \begin{cases} \beta E & \text{if the word is a noun, verb, adjective or adverb} \\ \overrightarrow{0} & \text{if no related word could be found in the knowledge graph} \\ \overrightarrow{0} & else \end{cases}$$

where E is the concatenation of the top k words as embeddings and β the vector consisting of the weights in the knowledge graph as described above.

3.3.2 Potential Aspect Categories

Aspects are often super categories such as *food* and *drinks*. These categories come up in natural language via many different potential opinion targets. In an extensive knowledge base these potential opinion targets are represented with a relation or transitive relation via subcategories to a super category. Using this explicit knowledge in detecting aspect categories could potentially help to improve classification rates. This is shown in a constraint form in Figure 3.5.

This way of collecting external knowledge resembles the approach described in the previous subsection. We filter a sentence for nouns, verbs, adjectives and adverbs. The main difference is that we do not describe the weighted sum of related words but try to utilize its inherent categorical relations. For example, if we have a potential aspect term such as *apple* for aspect *FOOD#Quality* the goal is to look up the word apple in a knowledge base and traverse it via IsA relations until the transitive relation *apple* IsA *food* is found. If it is found, the word *food* is extracted, and its embedding returned as a result. If no relation is found, we take the weighted sum of the direct neighbours with an IsA relation. This should lead to an embedding that resembles the aspect.

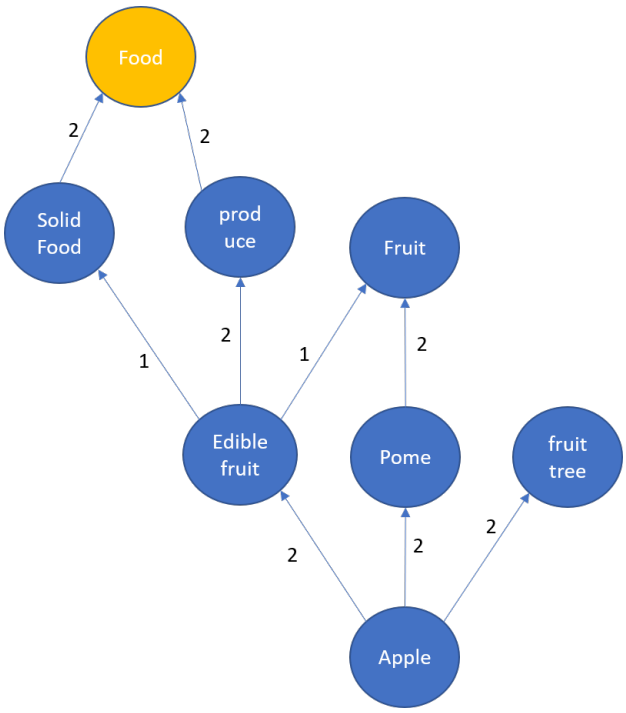


Figure 3.5: potential aspect categories excerpt

Since exploring the transitive relations in a graph can be quite costly, the algorithm is restrained by a maximum depth of 3. It also does not look at all the children of a node, but only at the 6 most relevant ones. This is again determined by the weight of the IsA relation.

Chapter 4

Experiments and Discussion

External knowledge can augment aspect category detection on various levels. As much as it has the potential to improve the overall detection of aspect categories, it might also help training the classifier in case the amount of data is insufficient.

In this chapter the architectural components from Chapter 3 will be combined and used to show what kind of an impact external knowledge can have for aspect category detection without having already extracted aspect terms. Furthermore, two different sources for external knowledge will be explored and two approaches for collecting external knowledge from a source compared.

The experiments are programmed in Python. It offers a lot of libraries for the machine learning context such as tensorflow [1], Keras [9] or spacy [41]. The Sentic Attention Network is programmed with Keras on top of tensorflow and the language model was used with spacy as it offers a lot of functionality for NLP tasks. Additionally, it enables you to use pretrained word embedding models without considerable effort and conveniently downloads them for you. As it is not a neuron provided by Keras itself the Sentic LSTM had to be implemented as an extension of the LSTM implementation by Keras. For the rest of the network, components provided by Keras were used. The knowledge gathering was implemented using Numpy [28], simple HTTP requests from the requests library and spacy for the vector representation of words.

4.1 Data

The experiments will be conducted with the SemEval Dataset from 2016 and will be performed against the *restaurants* domain on sentence level. Each sentence is annotated with a set of opinions that occur within the sentence. Each sentence may have $0 \ldots n$ opinions. Each has a sentiment value s indicated by the polarity attribute with $s \in \{$positive, negative, neutral, conflicted$\}$ and an aspect category defined by two words separated with a $\#$, as depicted in Figure 4.1. The first word is an entity such as *restaurant* or *food* while the second one is an aspect of the corresponding entity. Aspects are specific to the entities they belong to, meaning that an aspect for restaurants must not automatically be an aspect for food as well. The *restaurants* data set has twelve aspect categories, which can occur in every one of the 2000 sentences present. The test set is 808 samples big and features the same categories.

```
<sentence id="1014458:3">
<text>The wine list is interesting and has many good values.</
   text>
<Opinions>
<Opinion target="wine list" category="DRINKS#STYLE_OPTIONS"
   polarity="positive" from="4" to="13"/>
<Opinion target="wine list" category="DRINKS#PRICES" polarity=
   "positive" from="4" to="13"/>
</Opinions>
</sentence>
```

Figure 4.1: Data: Format of the SemEval data set 2016, Example Restaurants Domain

The SemEval data set from 2016 is one of the most popular data sets for aspect-based sentiment analysis and has been used in many publications even after the competition finished [19, 25, 6].

4.2 Experiments

The experiments are conducted on a cluster with four GeForce GTX 1080 Ti GPUs, 32 Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz CPUs and 93.1 GB RAM. All experiments presented here have been executed k-times setting $k = 5$ for the restaurants domain to even out random peaks and valleys in the calculated scores indicating the experiments success. Since the Sentic Attention Network used for all following experiments has a binary classifier for each possible category, the scores used in these experiments need to be adapted to sequences of binary values per category. To capture and compare the performance of the network as best as possible, the experiments will return the mean classification accuracy as well as the mean F1-Score over all categories. Additionally, it will return the mean precision and mean recall as this gives valuable insight on why the F1-Score behaves in a certain way

$$
\begin{aligned}
acc &= \frac{1}{n}\sum_{i=0}^{n} acc_i \\
acc_i &= \frac{tp_i + fn_i}{tp_i + fn_i + tn_i + fp_i} \\
f1 &= \frac{1}{n}\sum_{i=0}^{n} bin\_f1_i \\
f1_i &= 2 * (\frac{precision_i * recall_i}{precision_i + recall_i}) \\
precision_i &= \frac{TP}{TP + FP} \\
precision &= \frac{1}{n}\sum_{i=0}^{n} precision_i \\
recall_i &= \frac{TP}{TP + FN} \\
recall &= \frac{1}{n}\sum_{i=0}^{n} recall_i
\end{aligned}
$$

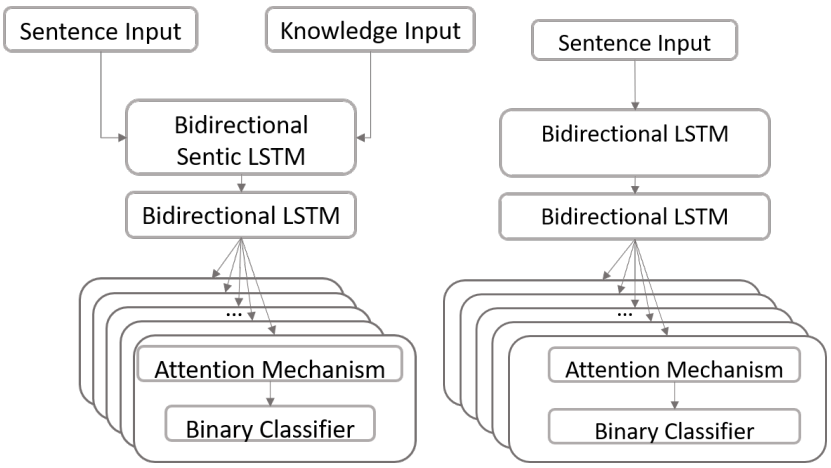
Figure 4.2: Scores

with $n \in \mathbb{N}$ and n equals the number of categories. $precision_i$ measures the rate of correctly classified samples of all the samples that have been predicted as value 1 within a category. The $recall_i$ returns a measurement for the rate of all samples that have correctly been classified 1 within all samples that have

been predicted as 1, correctly or not. Since we have multiple binary classifiers here plus on average there are about 2 aspect categories per sentence, a large amount of the data will be labeled 0. In fact, by classifying all categories a 0 on the test set returns an accuracy of slightly over 90. However, accuracy is a valid metric here but needs to be viewed considering this fact. All in all, the F1-Score might be a more accurate metric to measure the experiments' success. In the following experiment the mean accuracy, F1-Score, precision and recall are treated as actual accuracy, F1-Score, precision and recall.
Regarding the research on the effect of word embedding hyperparameters on aspect term extraction by Augustyniak et al. [2], the embedding dimension is chosen to be 300 and the spacy model used here is related to GLoVe 42B, which featured the lowest OOV rate in the SemEval Dataset of 2014. The vectors chosen for the spacy model are trained by GLoVe on the Wikipedia corpus and the common crawl dataset. This dataset also has a *restaurants* domain. Accordingly, the results by Augustyniak et al. are transferable to the dataset from 2016.
The hyperparameters of the Sentic Attention Network have been optimized using the hyperas framework[33]. Hyperas is a python library that offers a Keras specific interface for the optimization framework hyperopt [4]. Hyperopt optimizes hyperparameters not by Grid- or Random-Search but minimizes a chosen metric using an interface function that is interchangeable. The best algorithm for Keras has proven to be the Tree-of-Parzen-Estimator [3], which is used in this paper. For the hyperparameter optimization 10% of the training data was set aside as a validation set. Thereby, the best model was determined on how well the model performed on the validation set, which had not been seen during training.

4.2.1 Performance with external knowledge

In order to evaluate the performance of the Sentic Attention Network proposed in Subsection 3.2, a baseline model was developed to evaluate against. The performance of the baseline model shows the performance of an equivalent model without external knowledge. Therefore, the baseline model has the same architecture as the Sentic Attention Network. The similarity is shown by the comparison in Figure 4.3. Contrary to the Sentic Attention Network the baseline model does not have an additional input for external knowledge. Since there



(a) Architecture Sentic Attention Network

(b) Architecture Baseline Model

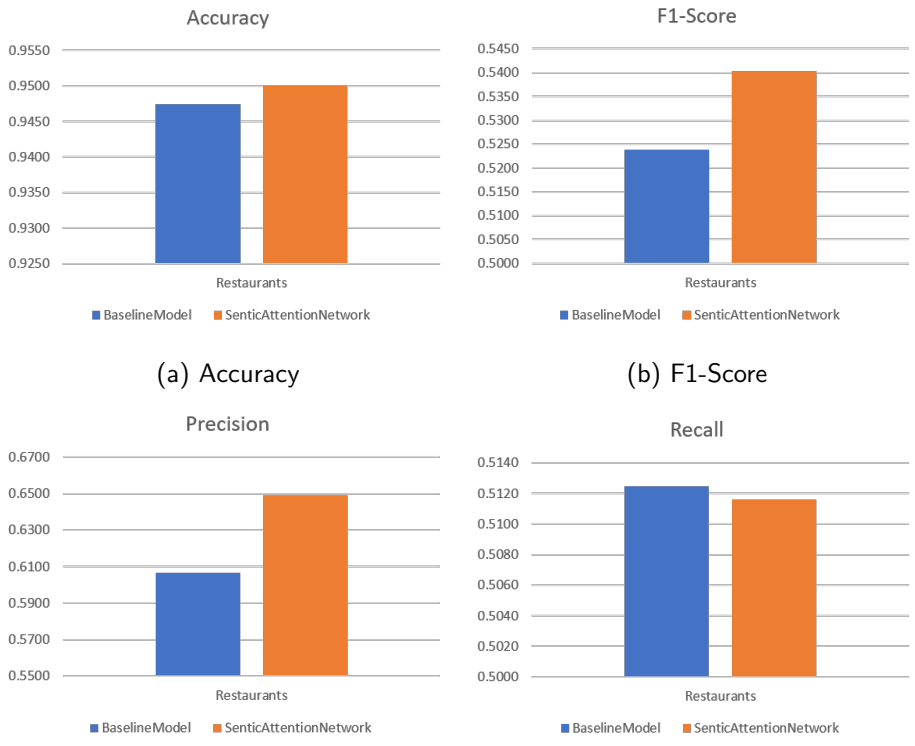
Figure 4.3: Architecture Comparison

is no external knowledge involved, it does not use a bidirectional Sentic LSTM but a regular bidirectional LSTM to process the input sequences. From this point on the baseline model's architecture is exactly the same. Comparing it to the Sentic Attention Network, the model has been implemented with the same technologies and the same hyperparameter settings.

This model's performance will be compared to the NLANGP model [46], the UFA-L model [44] and the Topic-Attention Network proposed by Movahedi et al. [25]. The NLANGP system was chosen as the winner of task 5 sub-task 1 slot 1 in the SemEval Challenge 2016, which was aspect category detection without previously extracted aspect terms. UFA-L was the leading approach in this competition with a comparable architecture featuring a LSTM layer. The system of Movahedi et al. [25] achieved the best performance of these three systems and thus was added.

Figure 4.4 shows the model's performance on the restaurants domain. It can be seen that for the restaurants domain the Sentic Attention Network performs better than the baseline model in terms of accuracy, F1-Score and precision. However it's performance is slightly lower than the baseline model when it comes to the recall. Consequently for the restaurants domain adding additional knowledge helps to classify the aspect categories. This becomes even more clear when we look at the precision in Figure 4.4c. The precision of the Sentic

Attention Network is around four points higher than the baseline model's. It follows that with external knowledge we can classify a lot more positive examples correctly. However, since the F1-Score does not reflect this large gap between the baseline model and the Sentic Attention Network, we have to look at the recall shown in Figure 4.4d. Since the recall indicates how many of the occurring positive samples were classified correctly, it shows that the baseline model suffers from less misclassifications of positive samples. This leads to the conclusion that external knowledge helps the Sentic Attention Network in making right decisions, therefore increasing the number of correctly classified aspect categories. Nevertheless, by looking at the recall we must also deduce that the external knowledge also confuses it. Otherwise there would be less falsely detected aspect categories.



(a) Accuracy

(b) F1-Score

(c) Precision

(d) Recall

Figure 4.4: Evaluation of the Models performance

Compared to the approaches described above, the Sentic Attention Network and the baseline approach do not reach the same level regarding the F1-Score.

This can be observed in Table 4.1. With 54.01 F1-Score, the Sentic Attention Network does not even come close to the nearest comparable approach of the SemEval Competition 2016 by Tamchyna et al. [44]. When comparing the technical foundation of the approaches by Movahedi et al., Yanase et al., Tamchyna et al. with the approaches presented in this thesis, you will find very similar architectures, especially compared to the baseline model. Each of the above described approaches use a form of recurrent neural network, here a LSTM or a GRU. On top Movahedi et al. also use multiple attention mechanisms. However, they outperform the approaches of this paper by a lot. As a result, it is not exactly clear at this point, why the results here do not really match the selected comparable approaches.

Approach	Restaurants
Movahedi et al. [25]	78.38
Toh et al. [46]	73.33
Yanase et al. [52]	60.15
Tamchyna et al. [44]	59.30
own work SAN	54.01
own work baseline	52.38

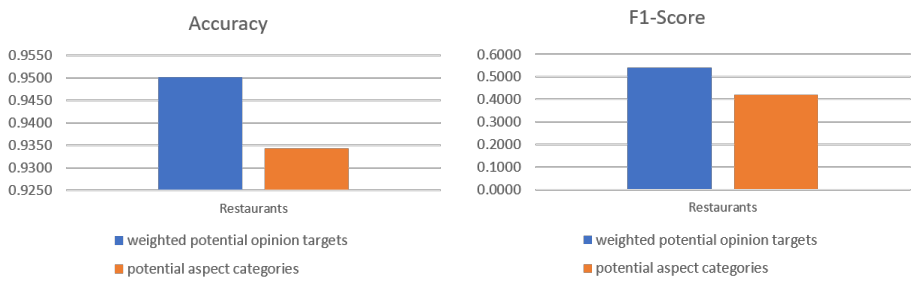
Table 4.1: Comparison to State of the Art

4.2.2 Comparison of knowledge collection methods

The Performance of a model using external knowledge depends heavily on the quality of the gathered knowledge. As there are several possible methods to gather external knowledge, this thesis restricts itself to two of them. These methods use the explicit knowledge presented in knowledge graphs by traversing it. However, their operational method and the result they collect differs as explained in Subsections 3.3.1 and 3.3.2.

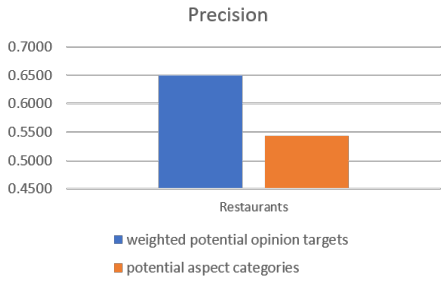
This experiment compares the results the Sentic Attention Network achieves with the two different knowledge collection methods proposed in Section 3.3. The weighted potential opinion targets method uses related words to create an augmenting knowledge embedding for a word. The potential aspect categories method tries to match the aspect categories it has seen during training to super

categories of a word and gathers its augmenting information by using the first aspect category it has seen. Both methods leverage the same language model that is used to represent words in sentences as vectors for the Sentic Attention Network and both methods are employed on the same knowledge graph, which is ConceptNet [42].

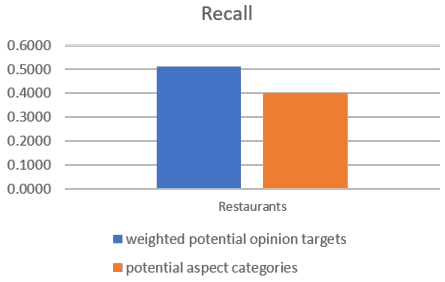


(a) Accuracy

(b) F1-Score



(c) Precision



(d) Recall

Figure 4.5: Evaluation of the methods for collecting external knowledge

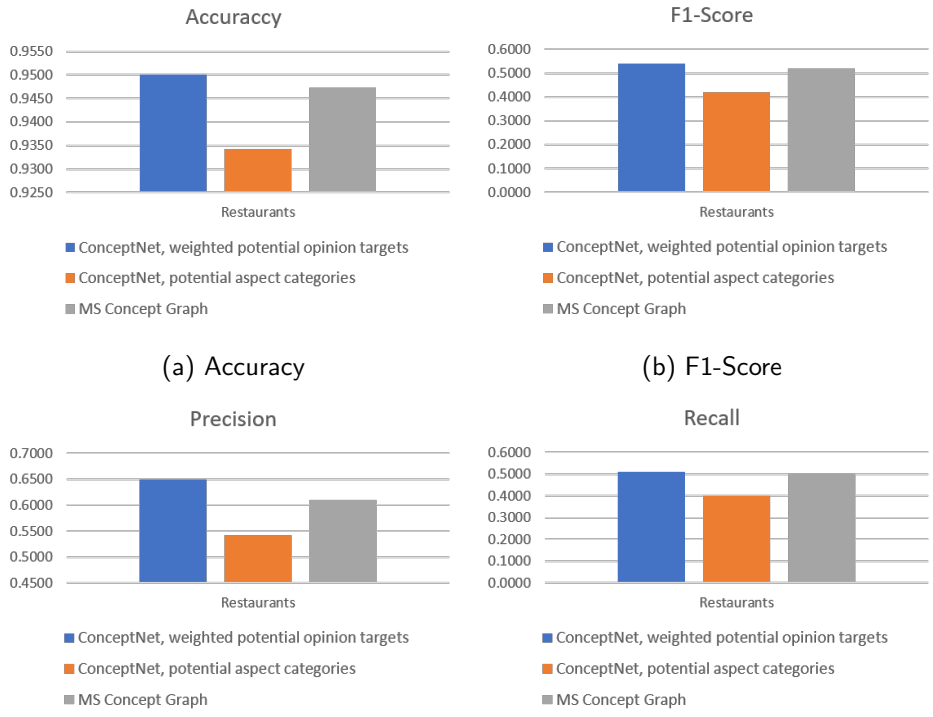
The results in Figure 4.5 indicate that the weighted potential opinion targets method produces superior results in every way. Looking at the F1-Score and accuracy in Figures 4.5b and 4.5a, we can see that the weighted potential opinion targets method beats the potential aspect categories method with an F1-Score of 54.03 to 42.03 and an accuracy of 94.99 to 93.43. The same goes for precision and recall. It could be due to the fact that the weighted potential opinion targets method more often produces meaningful embeddings. Moreover, it does not have to be restricted in depth as it only takes the k most related neighbours and usually finds matches in ConceptNet. The potential aspect categories method searches specific IsA-Relations for super categories by using breadth-first-search (BFS). This must of necessity be restricted in

depth, since otherwise for certain queries the duration of the query could grow exponentially, especially if no match concept can be found in the knowledge graph. Consequently, it is either due to this restriction or insufficient information in ConceptNet that the method might not find a matching super category and thus returns a null vector as the augmenting information. The loss of this relevant information could be a reason for the performance gap between these methods.

4.2.3 Comparison of external knowledge sources

ConceptNet is not the only knowledge graph that can provide augmenting knowledge to the Sentic Attention Network. There are a lot of proprietary and open source knowledge graphs available that can be used to generate augmenting external knowledge. Its quality depends of cause on the completeness of the knowledge graph regarding the domain on which aspect category detection is employed on. To evaluate the knowledge provided by ConceptNet, its result with the Sentic Attention Network will be compared with the result achieved with external knowledge from Microsoft (MS) Concept Graph. The method to extract knowledge from ConceptNet is the weighted potential opinion targets method as it has provided better results in the previous experiment. As MS Concept Graph is built on Probase, the graph is made up of IsA relations. Therefore, traversing this graph for information is similar to the way the weighted aspect categories method searches for potential aspect categories in the previous experiment. As it has achieved poor results compared to the weighted potential opinion targets method, we do not look for potential aspect categories again but create an embedding by taking the weights of the edges to the super categories, their word embeddings and calculate their weighted sum. This resembles the weighted potential opinion targets method with the difference, that we do not use RelatedTo but IsA relations.

By comparing the results of MS Concept Graph in Figure 4.6 with the potential aspect categories method from the previous experiment, we can definitely see an improvement in all the collected metrics. This comparison is interesting because both methods use IsA relations, but the knowledge source and the method, with which the embedding is aggregated, are different. While MS Concept Graph achieves an F1-Score of 51.81, the F1-Score of the potential aspect categories method with ConceptNet with 42.03 is significantly smaller. The performance



(a) Accuracy (b) F1-Score

(c) Precision (d) Recall

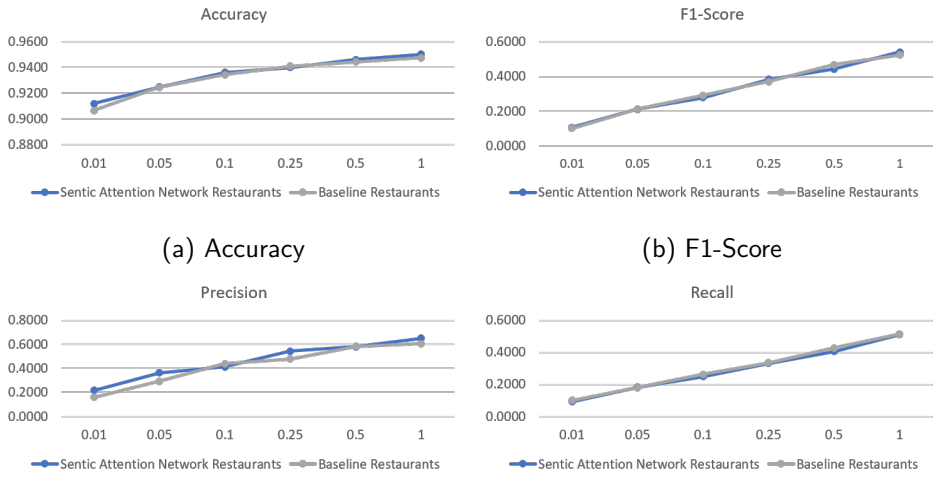
Figure 4.6: Evaluation of the external knowledge sources

gap can either be due to the different knowledge source or the methods themselves. However, if the results of MS Concept Graph are compared to the weighted potential opinion targets method with ConceptNet, we still see a better performance looking at the latter option. The weighted potential opinion targets method with ConceptNet performs a lot better. With an F1-Score of 54.01, this value exceeds the F1-Score for both previous methods. By looking at Figure 4.6 we can also observe this behaviour for the accuracy, precision and recall. However, it is curious to see that for the recall the weighted potential opinion targets method with ConceptNet and the method with MS Concept Graph are not far apart. This shows that in terms of falsely detected aspect categories the knowledge of both methods almost introduces the same amount of confusion into the neural network. The precision of the MS Concept Graph method is about four points lower than the precision of the weighted potential opinion targets method. This makes the latter's knowledge embeddings more suitable for aspect category detection without previously extracted aspect terms.

4.2.4 Influence of external knowledge on training

Adding augmenting knowledge to the input cannot only improve the overall performance of a system but also decrease the need for large amounts of training data to achieve an adequate performance. Since we add additional knowledge to the input, it might be assumed, that the neural network might not need as much training data to learn patterns from the input. As a result, it might achieve a good performance with significantly less data. For this reason, an experiment was constructed, where additionally to the whole data set, $0.01, 0.05, 0.1, 0.25$ and 0.5 of the data was used to train the Sentic Attention Network. Additionally the baseline model was also trained on these subsets of data to compare, how much less training data is needed for the Sentic Attention Network to perform similarly well.

Looking at the F1-Score and accuracy shown in Figure 4.7, we can see the baseline model's and the Sentic Attention Network's result indicate the same tendencies. For $0.01, 0.05, 0.25$ and 1 the Sentic Attention Network has a slightly better result, while for 0.1 and 0.5 the baseline model is a little superior. This leads to the conclusion that the models develop similarly the more data they are trained on. Here we can also see the phenomenon observed in the previous experiment that the Sentic Attention Network is a lot better in precision while



(a) Accuracy

(b) F1-Score

(c) Precision

(d) Recall

Figure 4.7: Performance of Baseline Model and Sentic Attention Network on a percentage of data

the baseline model has the better recall most of the time. This supports the suggestion mentioned before that, while improving the classification of positive samples, it also produces a higher rate of falsely classified aspect categories. Seeing this phenomenon repeated with various amounts of training data, concludes that the confusion entered into the network does not increase or decrease with the amount of data the network has been trained on.

4.3 Discussion

The experiments conducted in the previous section have evaluated many different sides that play a role in the Sentic Attention Network. The most important one is without a doubt the performance of the model. From the experiment in Subsection 4.2.1 we can see that for the Sentic Attention Network, external knowledge does actually help in improving the networks performance by comparing it to the baseline model. The performance of the model depends of cause on many external factors.

One of the most important factors in NLP tasks is the quality of the word embeddings used to represent the input sentences in a vector space. The spacy

model used in all of these evaluations consist of GLoVe vectors. The GLoVe 42B is the model that was used by Augustyniak et al. [2] in their evaluations of recent word embedding models for aspect category detection. As these models are trained on the same dataset by the same algorithm and the GLoVe 42B's embeddings achieved the best results, the smaller spacy model has to be evaluated for it suitability compared to the GLoVe 42B model. The spacy model is only half the size of the original GLoVe model measured by the number of tokens it can represent. Augustyniak et al. have argued that the with every word that is not represented by a language model potential knowledge about the context where such a word occurs gets lost. They have found a correlation between the out-of-vocabulary (OOV) rate for a model and a dataset with the performance of aspect category detection on that dataset. As a result, the best performance they could manage was with an OOV rate of 3.46% on the SemEval restaurants dataset of 2014. Accordingly, the OOV rate for the spacy model and the *restaurants* data set was determined with 3.3%. The difference of 0.16% is negligible. Hence, the spacy model is as well suited for the SemEval resturants data set of 2016 as the GLoVe 42B model is for the SemEval restaurants dataset of 2014. Since even slightly less words are missing from the vocabulary there than in GLoVe 42B with the data set from 2014, it might to some extend even be better.

In this thesis two experiments were conducted in Subsections 4.2.2 and 4.2.3 that concern external knowledge. The first one dealt with the ways relevant knowledge can be collected from a knowledge graph, in this case ConceptNet. The second one evaluated the impact of using different knowledge sources. As the quality of the external knowledge is crucial for the performance of this task, its embeddings need to reflect the demanded information as accurately as possible.
In both experiments the representation of knowledge was conducted by using the same spacy language model as for the representations of sentences in vector space. The experiment comparing the external sources ConceptNet and MS Concept Graph has shown that the knowledge provided by ConceptNet has proven to yield better results with the Sentic Attention Network. The results achieved with MS Concept Graph are inferior especially by looking at the precision. Even though the vector representations are created in a similar fashion,

there can still be multiple reasons for this performance disparity. The first reason is presented by the knowledge bases themselves. While ConceptNet offers only common-sense knowledge, MS Concept Graph also contains the manifestation of concepts such as a specific restaurant. While this might help in some cases, where restaurant names are mentioned in a review and that name is popular enough to have made it into the MS Concept Graph, the structure of MS Concept Graph constraints it, delivering better information to this task. Even though it can potentially contain an unlimited number of concepts, restricting it to IsA relations limits the way these concepts can be related. Therefore, by comparing the results, it seems to be that the concept hierarchy in MS Concept Graph does not contain enough information for the Sentic Attention Network to perform better than with ConceptNet. Therefore, having more general relations such as RelatedTo helps in collecting more dimensions of knowledge than just its categorical hierarchy.
Another big influence on the performance is of cause the way the external knowledge is collected in addition to which source it is taken from. Although this is very similar for both knowledge sources, it might not be the best solution for the kind knowledge that is captured in their sources. Taking the weighted sum of the nearest k neighbours is a good solution if you have a general relation such as RelatedTo but this method does not take advantage of a hierarchical structure that is implied by the IsA Relation. To use the categorical knowledge of MS Concept Graph optimally this should be considered but is neglected here.

This leads us to the knowledge collection methods. The experiment in Subsection 4.2.2 has compared the weighted potential opinion targets method and the potential aspect categories method. The potential aspect categories method similar to the MS Concept Graph method operates on IsA relations and tries, contrary to the collection method for MS Concept Graph, to leverage the categorical hierarchy built by IsA relations in ConceptNet. To primarily use this hierarchy when working with IsA relations makes sense. Most of the words coming up in a sentence belong to a category, such as lasagne is food and a beer being a drink in the restaurant domain. If aspect categories are defined thus, leveraging the hierarchy is a good idea to find the aspect category implied by a word. However, the experiment has demonstrated that using this categorical hierarchy does not improve the performance. This might be due to the depth

constraint of the BFS or the fact the embeddings of the words making up the aspect categories do not really reflect the category itself in the vector space made up by the language model. Also, there is the fact that ConceptNet might not have enough IsA relations so that for more uncommon words no categories can be found due to missing relations between nodes. Consequently, it restricts the effectiveness of this approach further. It was really surprising that this method performed to poorly because in theory for each aspect term it is possible to trace it back to a super category, which incidentally is also an entity or aspect in an aspect category. Handing this information to a neural network so that it can considering the context of a sentence to determine which aspect categories come up, seemed to be a plausible idea. But due to the facts mentioned above this method did not perform as good as was hoped for.
Even though using the weighted potential opinion targets method with ConceptNet has proven to be the best one, compared with state-of-the-art models this approach with the Sentic Attention Network has failed to perform as well. One reason for this could be that the coverage of relevant words by ConceptNet with 66.14% is relatively low for the *restaurants* domain. Relevant words are defined by a word having a potential impact on the aspect category. In this context only nouns, verbs, adjectives or adverbs are considered relevant. Among these not covered words are critical terms for the *restaurants* domain such as different dishes indicating food or composite words like over-priced. These words have a clear indication for a specific aspect category that cannot be captured. For MS Concept Graph the coverage with 61.1% is even lower. Thus, sometimes crucial information to augment the knowledge for the aspect category detection is missed. That can explain why this approach has performed worse than the comparable approached listed in Table 4.1.

Compared to these approaches except NPLANG by Toh et al. [46], which used CNNs, similar or inferior technologies were used. Yanase et al. [52] and Tamchyna et al. [44] have not even utilized an attention mechanism following their RNN to direct attention to specific parts of a sentence. Using attention mechanisms can boost the performance by a long-shot as Movahedi et al. [25] demonstrated in comparison. Especially compared to Movahedi et al. the results of the Sentic Attention Network seem insignificant. The two approaches apply similar technologies in the architecture of their neural networks. Movahedi et

al. adopt a bidirectional GRU instead of a bidirectional LSTM and the way they apply the attention mechanism is slightly different. They use a fixed number of attention mechanism following the bidirectional GRU and concatenate their results. The classification is performed with n fully connected layers, producing a binary output with n being the number of categories. The output is mapped to $[0; 1]$ by a squash function acting as an activation function and the L2 norm. This architectural difference was also tested with the Sentic Attention Network but yielded no performance increase compared to the architecture presented in Section 3.2. Therefore, it remains unclear why this network is not performing to the standards set by these comparable approaches on the same dataset, since minor architectural differences such as using a GRU instead of an LSTM typically do not result in a performance gap of around 20. Especially compared to the baseline model, the only larger architectural difference is that an attention mechanism is employed for each category. Following this the presence of that category is classified by the output of the attention mechanism. However, this does not constitute a big architectural difference specifically when compared to Movahedi et al.

By looking at just the baseline model and the Sentic Attention Network we can discover that in the case of the Sentic Attention Network external knowledge does actually help in improving the performance. As the experiment in Subsection 4.2.1 has demonstrated external knowledge can help in classifying aspect categories. Especially looking at the precision shows this is the case. However the confusion that comes with the external knowledge is a problem. It should help the neural network to make better decisions but not increase its wrong classifications at the same time. This could be due to some categories not being clearly distinguishable. For example, talking about saying *„The desert was nice.“* could be considered as the aspect food quality or also food styling options. In this case external knowledge, the way it is used in this thesis, cannot help because the sentence does not contain an indicative word whose meaning could be made plainer by using a knowledge base. All the knowledge augmentation would find in this case are related concepts to desert, which would in turn be related to food and concepts related to nice, which indicates the sentiment of the statement more than it indicates anything aspect category related. Therefore, some forms of misclassification cannot be helped even with external knowledge, while others can. Clear misclassifications need to be avoided by using external

knowledge. By looking at the recall however this is not the case here as the baseline model's recall is higher than the Sentic Attention Network's. This could be due to insufficient or even confusing external knowledge. For example, the word *roast*, which comes up in the restaurant domain with for example meat also has a meaning in the entertainment industry where words such as sarcasm or joke would be related to it. While this is perfectly valid in a general-purpose context, in a predefined domain these facts introduce misleading knowledge into the embedding. This results in a lower quality embedding for that predefined domain, here restaurants. Furthermore it can confuse the network, which then leads to a larger amount of wrong predictions.
On the other hand, it needs to be taken into account that the more a model gets crafted for a certain domain, it loses its general applicability. Right now, the Sentic Attention Network with the weighted potential opinion targets method and ConceptNet works on knowledge that was not designed for a specific domain. Therefore, the approach should be applicable to more domains apart from restaurants without altering it. This of cause is a vital quality of a model if it is considered in practical application.

In training, external knowledge does not really support in scoring the same performance with less data. The experiment in Subsection 4.2.4 has not validated the previous assumption. The reason for this might again be the quality of the external knowledge embeddings, influenced by missing or misleading knowledge. Even though the precision is constantly higher than the recall, we discover that this is the case for both models. The Sentic Attention Network's precision is typically higher than the baseline model's but there is no clear pattern here. Since at four out of six measurements the Sentic Attention Network has better F1-Sore. Thus, we can say that the performance measurement from the experiment in Subsection 4.2.1 was not just a single occurrence but a trend. Consequently, we cannot derive a quote how much less training data is needed when using external knowledge in this case. The two measurement points, where the baseline model was better, contradict the previously assumed hypothesis. Thus, we can only say that Sentic Attention Model tends to be better but not conclude this as a general rule.

4.3.1 Limitations

The general setup of the experiments has a couple of weak point where errors can be introduced to these experiments. The first obvious source for errors derives from the connection to the external knowledge sources. During the development of the Sentic Attention Network and deployment on the machine where the experiments were run, a problem came up. The connection of the requests for knowledge from MS Concept Graph and ConceptNet was timed out multiple times and the knowledge could not be collected. The implementation was altered to introduce a persistent cache for every knowledge source and knowledge gathering method. This alternation was made to collect the necessary information beforehand and append it to the cache if that information was not already saved. There is still the possibility that at the time of the evaluation certain knowledge could not be collected beforehand or during the preprocessing of the training data. This would impact the evaluation in a negative way.
As mentioned before the external knowledge embeddings can only be as good as the knowledge provided by the knowledge sources. As there are ambiguities in every language, words can have a different meaning in different contexts. Consequently, this helps in some cases where you need general understanding of a language and thus a general representation of it. However, this does not help when dealing with a special domain such as restaurants. There you only want the meaning of the word in that domain reflected in the embedding. All other related concepts from other domains pull your knowledge embedding in a direction that might be contra productive in a downstream task such as aspect category detection. Therefore, a general-purpose knowledge base such as ConceptNet or MS Concept Graph might not be as effective as a knowledge base tailored to that specific domain. However, a domain specific knowledge base would restrict the applicability of an approach. So, we would have the trade-off between better performance and wide applicability.

Chapter 5

Conclusion

This thesis' aim was to find out if solving aspect category detection without already extracted aspect terms can be improved by adding external knowledge about the context from an explicit knowledge base. To achieve this the Sentic Attention Network was proposed to integrate external knowledge into a neural network architecture that solves this problem. Its architecture is built on the Sentic LSTM by Ma et al. [21] to incorporate external knowledge into the neural network to help in solving the task of aspect category detection without already extracted aspect terms. As Ma et al. have used this technique successfully for aspect category detection with already extracted aspect terms, the plan was to apply parts of their approach to the problem at hand. For these two methods of collecting external knowledge from knowledge bases were created.

The results have clearly demonstrated that the Sentic Attention Network cannot compete with state-of-the-art models. However, it has presented that external knowledge can augment a task like aspect category detection without already extracted aspect terms. This is underlined by comparing the Sentic Attention Network's performance on metrics such as the F1-Score and accuracy to the baseline model. We can clearly see a performance improvement on the test set by about 3 points. This fact itself is encouraging since is shows that in an architecture such as the baseline model, which does not differ that far from other state of the art networks, introducing external knowledge overall has benefit.

This thesis has also established, that the way external knowledge is collected, plays a major role in the performance of a model. In their approach Ma et al. have used knowledge graph embeddings for concepts as their augmenting

knowledge. However, in this thesis the approaches were rather based on collecting knowledge from knowledge graphs by using the graph structure explicitly. The weighted potential opinion target method performed best by operating on ConceptNet. Its performance with the Sentic Attention Network surpassed the potential aspect categories method by a lot. Additionally, Microsoft Concept Graph was evaluated as another knowledge base but did not provide knowledge embeddings that were as good as the ones provided by the weighted potential opinion target method with ConceptNet.
An astonishing observation was however that adding external knowledge did not help in reducing the amount of training data needed to achieve the same result as the baseline model that was trained without external knowledge. The experiment that was conducted to test this hypothesis was not able to determine a general rule or rate about how much less training data is needed with external knowledge to perform as well as the same model without external knowledge.
All in all, it can be said that while helping in the task of aspect category detection without already extracted aspect terms, external knowledge needs to be examined from multiple point. The quality of the external knowledge and the way it is integrated in a neural network architecture play huge role in whether the external knowledge is actually beneficial. This thesis has only evaluated one neural network architecture and a few ways of collecting external knowledge. Therefore, the answer to the question raised at the beginning of this thesis is, external knowledge can help in aspect category detection without already extracted aspect terms but there is still a lot of potential for making it more efficient by optimizing the gathering of external knowledge and way it is integrated into a neural network architecture further. The Sentic LSTM is a good start for integrating external knowledge but the challenge now must be to get such an architecture to perform up to the state-of-th- art or even to surpass it.

5.1 Future Work

Since this thesis has established that external knowledge can help in categorizing aspect categories, we take a look at other techniques to gather external knowledge to provide better knowledge embeddings. One of these techniques are knowledge graph embeddings, which were used by Ma et al. [21] in their original paper on the Sentic LSTM. These embeddings are generated by algo-

rithms such as TransH [48] or Hole [27]. They learn embeddings for entities and relations in the knowledge graphs by utilizing a neural network architecture called associative memory to predict a part of a knowledge graph triple. Based on what kind of embeddings are needed, during training either entities or relations are predicted. First approaches with the Sentic Attention Network and knowledge graph embeddings for Wikidata have not yielded good results. However, employed on other knowledge graphs these techniques might lead to better knowledge embeddings and thus boost the performance.

Other improvements that can be made of cause are to have a look at other knowledge graphs that might provide better knowledge graph embeddings or even just better results with the existing knowledge gathering approaches. Also changing the architecture of the Sentic Attention Network is a possibility. The successful application of the transformer architecture to various NLP tasks opens the question whether such an architecture can successfully be augmented by external knowledge and applied to aspect category detection without already extracted aspect terms.

It would also be interesting how this approach works on other data sets out there. This would give a more complete view about the wide range applicability of this general approach.

Bibliography

[1] M. Abadi. TensorFlow: learning functions at scale. *ACM SIGPLAN Notices*, 51(9):1–1, 2016.

[2] Ł. Augustyniak, T. Kajdanowicz, and P. Kazienko. Comprehensive Analysis of Aspect Term Extraction Methods using Various Text Embeddings. Technical report, 2019.

[3] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, Granada, Spain, 2011.

[4] J. Bergstra, D. Yamins, and D. D. Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. *12th PYTHON IN SCIENCE CONF*, (Scipy):13–20, 2013.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[6] N. Cai, C. Ma, W. Wang, and D. Meng. Effective self attention modeling for aspect based sentiment analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11540 LNCS, pages 3–14. 2019.

[7] E. Cambria, S. Poria, D. Hazarika, and K. Kwok. SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embed-

dings. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1795–1802, New Orleans, Louisiana, USA, 2018.

[8] M. Chernyshevich. IHS R & D Belarus : Cross-domain Extraction of Product Features using Conditional Random Fields. In *SemEval@COLING.*, number SemEval, pages 309–313. Dublin, Ireland, 2014.

[9] F. Chollet. Keras: The python deep learning library. Astrophysics Source Code Library., 2018.

[10] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser. Universal transformers. In *7th International Conference on Learning Representations, ICLR 2019*, pages 1–23, New Orleans, Louisiana, USA, 2019.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, USA, 2019.

[12] B. Ding, Q. Wang, B. Wang, and L. Guo. Improving knowledge graph embedding using simple constraints. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 110–121, Florence, Italy, 2018.

[13] H. H. Do, P. W. Prasad, A. Maag, and A. Alsadoon. Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review. *Expert Systems with Applications*, 118:272–299, 2019.

[14] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Web Semantics*, 6(4):309–322, 2008.

[15] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[16] I. Horrocks, P. F. Patel-Schneider, and F. Van Harmelen. From SHIQ and RDF to OWL: The making of a Web Ontology Language. *Web Semantics*, 1(1):7–26, 2003.

[17] D. Hu. An introductory survey on attention mechanisms in NLP problems. *Advances in Intelligent Systems and Computing*, 1038:432–448, 2020.

[18] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of the National Conference on Artificial Intelligence*, pages 755–760, San Jose, California, USA, 2004.

[19] N. Jihan, Y. Senarath, and S. Ranathunga. Aspect extraction from customer reviews using convolutional neural networks. In *18th International Conference on Advances in ICT for Emerging Regions, ICTer 2018 - Proceedings*, pages 215–220, Colombo, Sri Lanka, 2019.

[20] Q. Liu, H. Zhang, Y. Zeng, Z. Huang, and Z. Wu. Content Attention Model for Aspect Based Sentiment Analysis. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pages 1023–1032, New York, New York, USA, 2018.

[21] Y. Ma, H. Peng, and E. Cambria. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 5876–5883, New Orleans, Louisiana, USA, 2018.

[22] M. Mat, L. Huangt, B. Xiang, and B. Zhou. Dependency-based convolutional neural networks for sentence embedding. In *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, volume 2, pages 174–179, Beijing, China, 2015.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, Scottsdale, Arizona, USA, 2013.

[24] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[25] S. Movahedi, E. Ghadery, H. Faili, and A. Shakery. Aspect Category Detection via Topic-Attention Network. *CoRR*, abs/1901.0, 2019.

[26] K. Munir and M. Sheraz Anjum. The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*, 14(2):116–126, 2018.

[27] M. Nickel, L. Rosasco, and T. Poggio. Holographic embeddings of knowledge graphs. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, abs/1510.0:1955–1961, 2016.

[28] T. E. Oliphant. Python for scientific computing. *Computing in Science and Engineering*, 9(3):10–20, 2007.

[29] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1532–1543, 2014.

[30] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep Contextualized Word Representations. In *Proc. of NAACL*, pages 2227–2237, New Orleans, Louisiana, USA, 2018.

[31] S. Poria, E. Cambria, and A. Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49, 2016.

[32] S. Poria, N. Ofek, A. Gelbukh, A. Hussain, and L. Rokach. Dependency tree-based rules for concept-level aspect-based sentiment analysis. *Communications in Computer and Information Science*, 475:41–47, 2014.

[33] M. Pumperla. Hyperas, 2015. https://github.com/maxpumperla/hyperas.

[34] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[35] S. Ruder, P. Ghaffari, and J. G. Breslin. INSIGHT-1 at SemEval-2016 Task 5: Deep learning for multilingual aspect-based sentiment analysis. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, pages 330–336, San Diego, California, USA, 2016.

[36] J. Schmidhuber. Deep Learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[37] M. Schmitt, S. Steinheber, K. Schreiber, and B. Roth. Joint Aspect and Polarity Classification for Aspect-based Sentiment Analysis with End-to-End Neural Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1114, Brussels, Belgium, 2019.

[38] K. Schouten and F. Frasincar. Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830, 2016.

[39] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[40] M. J. Somers and J. C. Casal. Using artificial neural networks to model nonlinearity: The case of the job satisfaction-job performance relationship. *Organizational Research Methods*, 12(3):403–417, 2009.

[41] SpaCy. spaCy, 2017. https://spacy.io/.

[42] R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451, San Francisco, California, USA, 2017.

[43] C. Sun, L. Huang, and X. Qiu. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. *CoRR*, abs/1903.0, 2019.

[44] A. Tamchyna and K. Veselovská. UFAL at SemEval-2016 task 5: Recurrent neural networks for sentence classification. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, pages 367–371, San Diego, California, USA, 2016.

[45] C. Thellaamudhan, R. Suresh, and P. Raghavi. A Comprehensive Survey on Aspect Based Sentiment Analysis. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(4):442–447, 2016.

[46] Z. Toh and J. Su. NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using neural network features. In *SemEval 2016*

- *10th International Workshop on Semantic Evaluation, Proceedings*, pages 282–288, San Diego, California, USA, 2016.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009, Long Beach, California, USA, 2017.

[48] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, pages 1112–1119, Québec City, Québec, Canada, 2014.

[49] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems*, 26(3):1–37, 2008.

[50] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 481–492, New York, New York, USA, 2012. ACM Press.

[51] J. Yan, C. Wang, W. Cheng, M. Gao, and A. Zhou. A retrospective of knowledge graphs. *Frontiers of Computer Science*, 12(1):55–74, 2018.

[52] T. Yanase, K. Yanai, M. Sato, T. Miyoshi, and Y. Niwa. Bunji at SemEval-2016 Task 5: Neural and syntactic models of entity-attribute relationship for aspect-based sentiment analysis. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, pages 289–295, San Diego, California, USA, 2016.

Hiermit versichere ich, Kai Martinen, an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Informatik mit dem Titel: "Knowledge Augmented Aspect Category Detection for Aspect-based Sentiment Analysis" selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, den 01.12.2019

. . . . . .

Kai Martinen