A Memory-Driven Neural Attention Model for Aspect-Based Sentiment Classification

Jonathan van de Ruitenbeek, Flavius Frasincar and Gianni Brauwers*

Erasmus School of Economics, Erasmus University Rotterdam, 3062 PA Rotterdam, the Netherlands
E-mail: jjvanderuitenbeek@gmail.com; frasincar@ese.eur.nl; gianni.brauwers@gmail.com
**Corresponding Author*

Abstract

Sentiment analysis techniques are becoming more and more important as the number of reviews on the World Wide Web keeps increasing. Aspect-based sentiment analysis (ABSA) entails the automatic analysis of sentiments at the highly fine-grained aspect level. One of the challenges of ABSA is to identify the correct sentiment expressed towards every aspect in a sentence. In this paper, a neural attention model is discussed and three extensions are proposed to this model. First, the strengths and weaknesses of the highly successful CABASC model are discussed, and three shortcomings are identified: the aspect-representation is poor, the current attention mechanism can be extended for dealing with polysemy in natural language, and the design of the aspect-specific sentence representation is upheld by a weak construction. We propose the Extended CABASC (E-CABASC) model, which aims to solve all three of these problems. The model incorporates a context-aware aspect representation, a multi-dimensional attention mechanism, and an aspect-specific sentence representation. The main contribution of this work is that it is shown that attention models can be improved upon using some relatively simple extensions, such as fusion gates and multi-dimensional attention, which can be implemented in many state-of-the-art models. Additionally, an analysis of the parameters and attention weights is provided.

River Journal, 1–36.

Keywords: aspect sentiment classification, sentiment analysis, deep learning, attention models.

1 Introduction

Applications of Natural Language Processing (NLP) have never been studied more intensively than in the last decade. One of the most prominent reasons for this is the ever-increasing number of texts available on the World Wide Web [14]. With this abundance of online documents, there is a need for the development of NLP techniques that can be used to automatically process these texts. Popular applications of NLP include information extraction, question answering, text summarization, speech recognition, entity recognition, and machine translation [20]. Yet, one of the most prominent branches of NLP is the field of sentiment analysis [25, 29, 39]. This field of research uses NLP, statistics, and machine learning techniques to extract the sentiment of a text. With the rise of the social Web, people are sharing more and more opinions online on a wide-ranging variety of topics. The constantly growing number of online opinions allows consumers to make more informed decisions when purchasing products or services. Additionally, these sentiments can also be used to improve said products and services. A common medium used for sharing sentiments online is through Web reviews. Yet, it is highly impractical to manually read and filter through the abundance of Web reviews for a specific product. As such, there is a need for sentiment analysis techniques that allow for the automatic extraction of sentiments from Web reviews.

Aspect-based sentiment analysis (ABSA), also known as feature-based sentiment classification [13], considers one of the most fine-grained applications of sentiment analysis. Rather than classifying the polarity of a complete Web review, like in classical sentiment analysis, the goal of ABSA is to identify the sentiment expressed for every aspect in the Web review. This task can be performed at the review level and the sentence level. The focus of this paper is on sentence-level ABSA. Consider the following Web review sample sentence from the SemEval 2016 task 5 restaurant data set:

'I liked the atmosphere very much but the food was not worth the price.'

The aspects in this sentence are given by 'atmosphere' and 'food'. Positive sentiment is expressed towards 'atmosphere', while negative sentiment is expressed towards 'food'. As such, the polarity of the first aspect is positive, whereas the polarity of the second aspect is negative.

Due to the explosive growth of the World Wide Web, ABSA has garnered more and more attention and has become a clearer and more defined field over the years [11]. The applications of ABSA are numerous. The social Web yields a tremendous amount of opinions on products, policies, people, companies, and so on. These opinions will often be directed at certain aspects of the topics. From a company's point of view, knowing the sentiment expressed towards features of their products can help them to improve on their production and marketing. For a service provider, being able to extract the opinions of their customers on the specific aspects of their services can be of great help to improve their business. ABSA is not only beneficial from the company's point of view, but also from the customer's point of view. Knowing the positive and negative aspects of a product or service can help potential customers in their decision-making. For example, when considering phone reviews, some customers may value screen quality more than battery life.

The task of ABSA is defined by two sub-tasks: the extraction, or identification, of the aspects, and the sentiment classification of the aspects. In this paper, the focus is on the sentiment classification step of ABSA, also known as aspect-based sentiment classification (ABSC). As for the aspect detection step, one can use techniques like the one presented in [41]. A highly successful model for ABSC is the content attention-based aspect-based sentiment classification (CABASC) model [26]. The CABASC neural attention model is chosen as a starting point because of its good performance in ABSA [26]. This neural attention model has the ability to represent text features without feature engineering and the authors show that the model outperforms feature-based classifiers [26]. However, three shortcomings of this model can be identified. First, it can be argued that the aspects are not well represented, because a simple average of the aspect embeddings is taken. Second, the current content attention mechanism can be extended by a construct that deals with polysemy in natural language. Third, the CABASC model features a weak design choice, because it combines two vector representations via addition. As such, the research question of this paper is defined as follows:

How can the limitations of the CABASC model be addressed to obtain a more effective method?

We propose the Extended-CABASC (E-CABASC) model, which extends the CABASC model with solutions for each of the identified problems. The proposed model contains a context-aware aspect representation, which is a better aspect representation. Furthermore, the new model contains a multi-dimensional content attention mechanism that weights the sentence feature-wise and deals with polysemy in natural language. Last, an aspect-specific sentence representation is constructed by fusing two sentence representations via a dimension-wise fusion gate. It is shown that every extension improves the model and that the resulting E-CABASC model outperforms the CABASC model. Additionally, an analysis is provided of the attention weights and the effect that the number of parameters has on the performance of the CABASC and E-CABASC models. To summarize, the main contributions of this work are as follows:

- Three general mechanisms are discussed that can be implemented in a wide variety of models and are shown to individually improve model performance.
- A parameter analysis is provided to evaluate the parameter efficiency of attention models.
- An attention weights analysis is provided to compare the attention mechanisms of the models.

The setup of this paper is as follows. In Section 2, related work is discussed. In Section 3, the CABASC model and several baseline models are explained. In Section 4, the proposed E-CABASC model is presented. In Section 5, the training settings of the models are discussed. In Section 6, the results are presented and the model parameters are evaluated. Last, In Section 7, conclusions and recommendations for further research are given.

2 Related Work

There are many different approaches to ABSC. A clear distinction can be made between standard feature-based models, which rely on input such as syntactic information, and neural network models. Neural network models for ABSC do not rely on external input features but solely rely on low-level features from the text representation. Popular supervised feature-based machine learning methods for the ABSC task include support vector machines [22], maximum entropy [5], naive Bayes [32], and even ensembles of these models [43]. Classifiers of this kind are often trained on sparse, high-dimensional feature vectors, requiring hand-crafted feature engineering work which relies on massive linguistic sources. The performance of these methods depends heavily on the quality of the input features. Using different features results in different performances of the method. Contrary to this, neural networks are trained on dense vector representations of text, yielding excellent results for various NLP tasks. The number of works published on deep learning for NLP tasks has been rapidly growing in the last few years [57]. Along with the success of neural network applications for NLP tasks in general, neural network models have been successfully applied to the task of ABSC, frequently outperforming feature-based sentiment classifiers. A variety of deep learning models, like recursive, convolutional, and capsule neural networks, can be used for sentiment analysis [10, 54, 9]. An overview of different deep learning algorithms along with applications for sentiment analysis is given by [59].

Convolutional neural network (CNN) models are sometimes used for ABSC [7, 54]. CNN models are particularly powerful in capturing high-level representations of images but are now used for NLP as well. An advantage of these types of networks is that computation time is much lower compared to, for example, long short-term memory (LSTM) models because it has fewer parameters to learn. In practice, however, LSTM based networks are more often used for ABSC because of their recurrent nature and ability to capture long-term dependencies [17]. Other approaches combine a CNN and recurrent neural network (RNN), producing a convolutional recurrent neural network (CRNN) [19].

Recursive neural network (RecNN) models can be used for ABSC as well. These types of networks take as input the entire sequence of text encoded by vectors and process the input sequence via a binary tree. [10] adopts an adaptive recursive neural network (AdaRNN) for ABSC. In [33], the AdaRNN is adapted by taking into account both the dependency tree and the constituent tree of the sentence. One disadvantage of these methods might be that the methods require grammatical parsing, which might be ineffective on non-standard texts [6].

An important development in ABSC with neural networks is the application of attention-based memory networks [3, 4]. [46] proposes the memory network (MemNet) model, which computes attention weights for each context word. First, the context is stored in a memory, after which the importance of each word in the context is quantified by generating a probability distribution over all words in the sequence. The MemNet model further adds some computational layers to the network which results in an aspect-specific sentence representation based on a deep memory representation. This approach is computationally faster than LSTM networks. In [46], it is shown that a memory-based network with 9 layers is 15 times faster than an LSTM, because of the complex operations within each LSTM unit along the input sequence. After the proposal of the MemNet model, many other researchers have adopted the idea of external memory. The content attention mechanism of the CABASC model described in this paper also uses this idea. In [47], the memory network is adapted via a more advanced mechanism to compute the attention weights. Rather than a simple dot-product or a small feed-forward network, the authors propose a dyadic memory network. This network uses neural tensor compositions or holographic compositions, which uses complex-valued computations based on a fast Fourier transform for learning the attention weights. This advanced attention mechanism is able to capture rich interactions between the aspect and its context.

An extension to the attention mechanism is multi-dimensional attention [42]. Multi-dimensional attention allows the model to properly define the importance of each feature. While this technique has shown promising results in other fields [3], to our knowledge, it has not yet been properly explored in the field of sentiment analysis. As such, the effectiveness of a multi-dimensional attention mechanism for ABSC is investigated in this paper.

Many developments in ABSC with neural networks consider LSTM networks, because of their great ability to represent sequential information. [45] proposes the target-dependent (TD)-LSTM and the target-connection (TC)-LSTM. These models aim to model the connection between the target words and the sentence context by using a left LSTM to model context preceding the target and a right LSTM for modeling context following the target. One drawback of the TD-LSTM and TC-LSTM is that if an important context word is far away from the target, the information contained in that word might be lost while it is propagated word by word to the target [6]. To overcome this problem for the TD-LSTM, [6] proposes a recurrent attention network on memory (RAN) model. This model uses a bidirectional LSTM to obtain a memory, which is then weighted according to its importance towards the target. This solves the problem of the TD-LSTM because context words are weighted according to their relative position to the target, which explicitly links a target to certain important context words.

In [50], an attention-based aspect embedding (ATAE)-LSTM is proposed, which takes into account target information as well. As such, this model takes the aspect embedding into account twice. First, the word embedding of every context word is extended with the aspect embedding. Second, the hidden states that are produced by the LSTM are extended with the aspect embedding. The resulting model performs well in cases when a sentence contains multiple aspects. In [27], the interactive attention network (IAN) is proposed, which is an attention-based LSTM. Contrary to many other models, this model does not only use an LSTM for the context, but it uses an LSTM for the target as well such that the model interactively learns attentions for both the contexts and the targets. The authors define that the model is interactive in the sense that the target and the context are related and make up the attention weights together. Similar co-attention models have been explored extensively recently [12, 58, 56, 55].

While the attention mechanism has mostly been used as an extension to recurrent and convolutional neural networks, transformer-based architectures [49] that solely rely on the attention mechanism have been proposed lately. These models rely on a specific attention mechanism known as self-attention [24]. Transformer-based models, such as BERT [8], are able to accurately capture the linguistic structure and have produced state-of-the-art results for a variety of NLP tasks. As such, the use of BERT for ABSC has become more popular recently [44, 52, 2]. In [21], the BERT model for ABSC is extended by using adversarial training [15]. Nevertheless, traditional self-attention has also shown limitations when modeling natural language [16].

3 Content Attention-Based Model

In this section, the CABASC model and three baseline models are discussed. In Section 4, the proposed E-CABASC model is discussed. This model features three extensions as compared to the standard CABASC model. After, the training details of the used neural networks are discussed. The following notation is used for all methods. A sentence of length N is denoted as $S = \{s_1, ...s_i, ..., s_{i+L}, ...s_N\}$, where s_j denotes word j and where the aspect is given by L words, $S_a = \{s_i, ..., s_{i+L}\}$. We assume that all sentences have length N. Shorter sentences are padded with zeros to length N for computational efficiency. If a sentence contains more than one aspect, separate instances are considered for each aspect, where each instance consists of the same sentence with a different aspect. The goal of all models discussed in this paper is to model the sentiment expressed in sentence S towards aspect S_a. The set of sentiment categories is denoted as C. This contains the three sentiment categories ‘negative’, ‘neutral’, and ‘positive’, which are encoded as 0, 1, and 2, respectively.

First, the CABASC model [26] is introduced. This neural attention model produces excellent performances for ABSC, obtaining a higher accuracy than many other attention models. The model performs particularly well due to its context attention mechanism. This mechanism takes into account the correlations between the context words and the target word, resulting in a customized

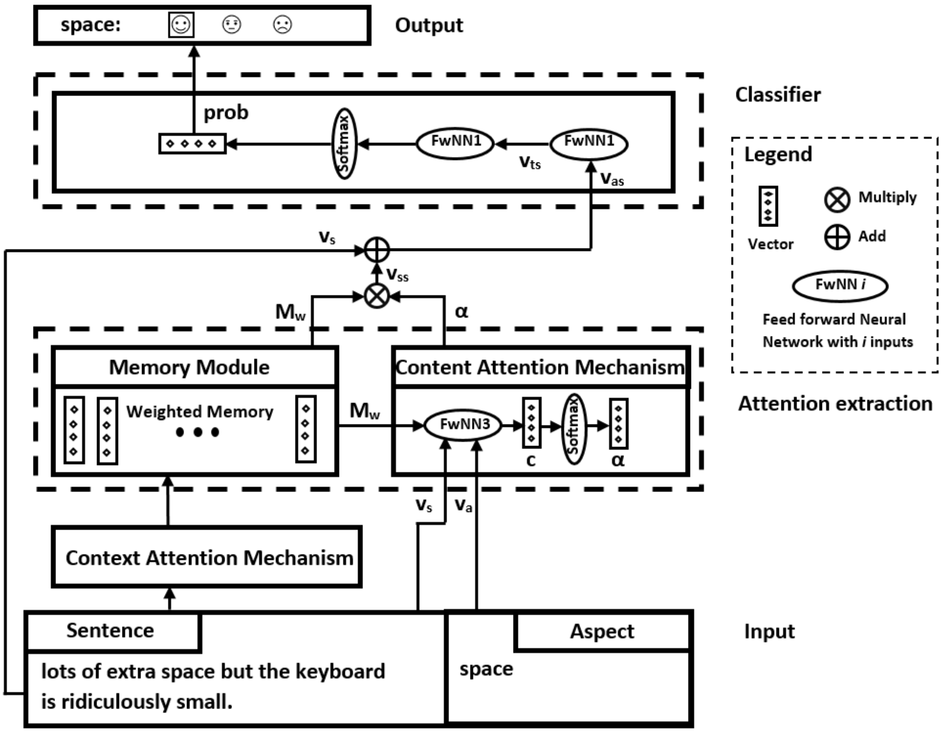

Figure 1: CABASC Framework.

attention-based aspect-specific memory. Furthermore, the model takes into account sentence-level information. A general overview of the CABASC framework is displayed in Figure 1. The example sentence contains the two aspects ‘space’ and ‘keyboard’. Only one aspect is processed at a time, which is why the aspect in the example is given by ‘space’.

The model consists of four different components: the input component, the attention extraction component, the classifier component, and the output component. The input component takes as input the sentence and the aspect and embeds these into a sentence embedding and an aspect embedding. The attention extraction component generates a memory of the context words, computes attention weights for every context word, and outputs an aspect-specific sentence representation. The classifier component applies a non-linear transformation to the aspect-specific sentence representation to increase the flexibility of the model. The result is passed through a linear layer and the softmax function. This generates a probability distribution over all sentiment categories. Last, the output component classifies the sentiment expressed towards the aspect based on the highest probability.

CABASC can be seen as a combination of several models that extend each other. To properly define and explain these extensions, two separate models are defined which are called baseline model A and baseline model B. Baseline model A is the simplest model, baseline model B extends baseline model A, and, lastly, CABASC further extends baseline model B. This section first discusses the word embedding component of the models, after which the two baseline models and the CABASC model are presented.

3.1 Word Embedding

The first step in the analysis is to embed the text data into vectors of real numbers. The collective name for mapping text to vectors is called word embedding. One solution for this is to represent every word by a one-hot encoding vector [34]. However, one-hot representations do not capture the complex linguistic characteristics of the words. Some effective word embedding solutions that map words to dense vector representations are the continuous bag-of-words model (CBOW) [31], the skip-gram model [31], and global vectors for word representation (GloVe) [35].

The different word embedding methods tend to perform similarly for different NLP tasks. [35] argues that the solution provided through the CBOW model and the skip-gram model are sub-optimal because they do not exploit all statistical information regarding co-occurrences of words. Also, the authors argue that GloVe performs best as compared to all other related models for similarity tasks and Named Entity Recognition. Given the excellent results of GloVe, this technique is used for the embeddings. The 300-dimensional GloVe vectors are used, with a vocabulary size of 1.9M[1].

Let $\mathbb{L} \in \mathbb{R}^{d \times |V|}$ denote a word embedding matrix of the vocabulary generated by GloVe or another unsupervised method. Here d denotes the dimension of a word vector and $|V|$ denotes the size of vocabulary V. The words in sentence S are mapped to a real-valued sentence embedding matrix $\mathbf{E} = [\mathbf{e}_1, ..., \mathbf{e}_N]$, where word s_j is mapped to vector $\mathbf{e}_j$ using $\mathbb{L}$. Note that the sentence embedding matrix contains the aspect-terms. Let $\mathbf{v}_a$ denote the $d \times 1$ embedding of aspect term S_a. If the aspect consists of multiple words, for example 'battery life', then $\mathbf{v}_a$ is computed as the mean over these em-

[1] Available at: https://nlp.stanford.edu/projects/glove/

beddings [26, 46]. Words that are not present in the vocabulary are initialized with a vector containing d zeros.

3.2 Baseline Model A

The first model under consideration is baseline model A. This model generates importance weights for all context words in the sentence, after which the result is processed through a non-linear and a linear layer. The different components of the model are defined as follows. The input component of the model takes as input a sentence and an aspect and maps these to word vectors. This results in sentence embedding matrix $\mathbf{E}$ and aspect embedding vector $\mathbf{v}_a$. The attention extraction component computes an attention weight for every context word. A schematic overview of the attention extraction component of baseline model A is displayed in Figure 2.

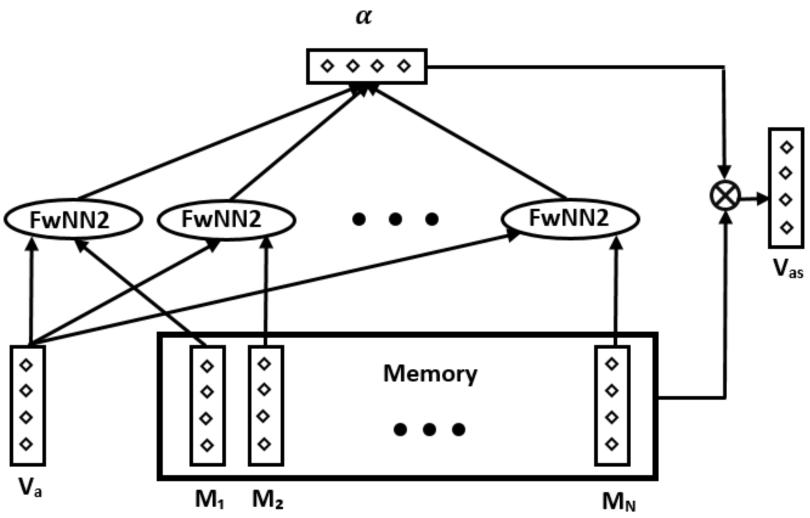

Figure 2: Attention Extractor Baseline Model A.

First, a long-term memory $\mathbf{M} = [\mathbf{m}_1, ..., \mathbf{m}_N]$ is constructed, which is a $d \times N$ matrix equal to $\mathbf{E}$. The memory module stores the sentence representation such that the complete context is used for prediction. The next module determines attention weights for each word in the context.

The **content attention module** determines the contribution of each word in the sentence towards the polarity of the aspect using an attention mechanism. First, the importance of each context word j is computed using a feed-forward neural network with as input the aspect representation $\mathbf{v}_a$ and the context word representation $\mathbf{m}_j$. By including the aspect representation in the computation of the attention weights, the model ensures that the attention weights are aspect-specific. This means that a sentence with multiple aspects generates a different set of attentions weights depending on the aspect. After this, the computed values are transformed to a distribution between zero and one through a softmax function.

For all words j in the sentence an importance score c_j is computed through a feed-forward neural network with two inputs $\mathbf{m}_j$ and $\mathbf{v}_a$ (FwNN2) as follows:

$$\underset{1\times 1}{c_j} = \underset{1\times m}{\mathbf{w}_1} \tanh(\underset{m\times d}{\mathbf{W}_2} \underset{d\times 1}{\mathbf{m}_j} + \underset{m\times d}{\mathbf{W}_3} \underset{d\times 1}{\mathbf{v}_a} + \underset{m\times 1}{\mathbf{b}_1}), \tag{1}$$

where the model parameters are given by the $1 \times m$ weight vector $\mathbf{w}_1$, $m \times d$ weight matrices $\mathbf{W}_2$ and $\mathbf{W}_3$, and $m \times 1$ bias vector $\mathbf{b}_1$. All the feed-forward networks share their parameters. The attention weights are processed through a softmax, where attention weight α_j is computed as follows:

$$\alpha_j = \frac{\exp(c_j)}{\sum_{i=1}^{N} \exp(c_i)} \in [0, 1]. \tag{2}$$

The output of the attention extraction component is given by the *aspect-specific sentence representation* $\mathbf{v}_{as}$, which is a weighted memory representation computed as follows:

$$\underset{d\times 1}{\mathbf{v}_{as}} = \underset{d\times N}{\mathbf{M}} \underset{N\times 1}{\boldsymbol{\alpha}}, \tag{3}$$

where $\mathbf{v}_{as}$ is a $d \times 1$ vector and where $\boldsymbol{\alpha}$ is an $N \times 1$ vector that contains the individual attention weights for all words in the sentence.

The classifier component transforms the aspect-specific sentence representation $\mathbf{v}_{as}$ to a probability distribution over all possible sentiment categories. First, the depth of the model is increased by processing $\mathbf{v}_{as}$ through a non-linear layer. This increases the flexibility of the model such that more abstract information from the aspect-specific sentence representation can be obtained. This *transformed sentence representation* is computed by a standard feed-forward network with 1 input (FwNN1) as follows:

$$\underset{m\times 1}{\mathbf{v}_{ts}} = \tanh(\underset{m\times d}{\mathbf{W}_4} \underset{d\times 1}{\mathbf{v}_{as}} + \underset{m\times 1}{\mathbf{b}_2}), \tag{4}$$

where $\mathbf{v}_{ts}$ is the resulting $m \times 1$ vector, $\mathbf{W}_4$ is an $m \times d$ weight matrix, and $\mathbf{b}_2$ a $m \times 1$ bias vector. In this way, the features in $\mathbf{v}_{as}$ are combined and m different representations are constructed such that every element in $\mathbf{v}_{ts}$ is a weighted sum of all features of $\mathbf{v}_{as}$. The transformed sentence representation is processed through a linear layer and a softmax layer to a probability distribution over all sentiment categories $c \in C$, as follows:

$$\underset{|C|\times 1}{prob} = \text{softmax}(\underset{|C|\times m}{\mathbf{W}_5}\ \underset{m\times 1}{\mathbf{v}_{ts}} + \underset{|C|\times 1}{\mathbf{b}_3}), \tag{5}$$

where $\mathbf{W}_5$ denotes a $|C| \times m$ weight matrix, $\mathbf{b}_3$ denotes a $|C| \times 1$ bias vector, and $|C|$ denotes the number of possible sentiment categories. The softmax function has the same functional form as in Equation 2 and transforms the input values to a probability distribution with values between 0 and 1. The resulting conditional probability distribution is given by a $|C| \times 1$ vector. The output component predicts the sentiment as expressed towards the aspect by choosing the sentiment value with the highest probability.

3.3 Baseline Model B

The model described above considers the contribution of each word towards the aspect polarity. However, the model does not take into account the meaning of a sentence as a whole. The second baseline model B considers an improvement over baseline model A to deal with this problem. The attention extractor of model A is extended by taking into account sentence-level information for the computation of the attention weights, and sentence-level information is added to the final weighted sentence representation. A schematic overview of the attention extraction component of baseline model B is displayed in Figure 3.

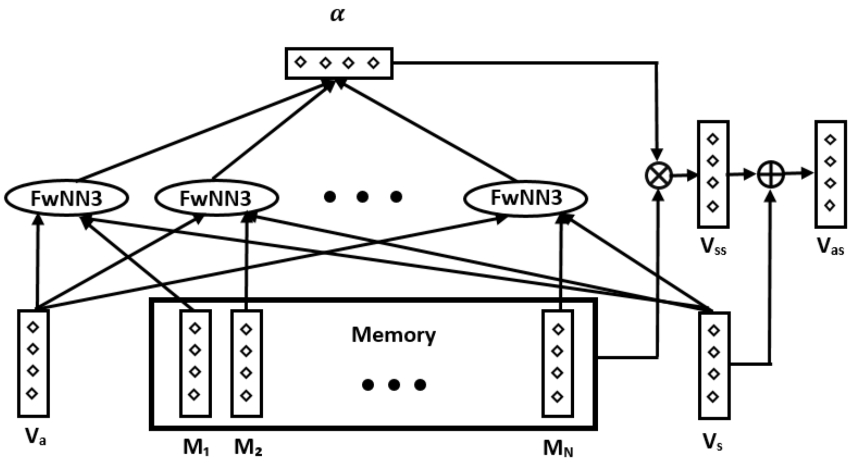

Figure 3: Attention Extractor Baseline Model B.

The content attention module is extended by adding the complete sentence representation of S to the model to obtain the **sentence-level CAM** (SAM). The feed-forward layer in Equation 1 is augmented to a feed-forward network with three inputs (FwNN3), formally denoted as follows:

$$\underset{1\times 1}{c_j} = \underset{1\times m}{\mathbf{w}_6} \tanh(\underset{m\times d}{\mathbf{W}_7}\ \underset{d\times 1}{\mathbf{m}_j} + \underset{m\times d}{\mathbf{W}_8}\ \underset{d\times 1}{\mathbf{v}_a} + \underset{m\times d}{\mathbf{W}_9}\ \underset{d\times 1}{\mathbf{v}_s} + \underset{m\times 1}{\mathbf{b}_4}), \tag{6}$$

where the dimensions of $\mathbf{w}_6$, $\mathbf{W}_7$, $\mathbf{W}_8$, and $\mathbf{b}_4$ are equal to the dimensions of $\mathbf{w}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, and $\mathbf{b}_4$, respectively, and where $\mathbf{W}_9$ denotes an $m \times d$ weight matrix. The sentence-level representation of sentence S is denoted as the $d \times 1$ vector $\mathbf{v}_s$, which is computed as the average of the embedded words in the sentence. [1] investigates the effects of different sentence embeddings on prediction tasks and finds that the average of the word embeddings serves as an effective sentence representation. All attention weights are then computed following Equation 6, after which the $d \times 1$ *aspect-specific sentence representation with sentence information* $\mathbf{v}_{ss}$ is computed as follows:

$$\underset{d\times 1}{\mathbf{v}_{ss}} = \underset{d\times N}{\mathbf{M}}\ \underset{N\times 1}{\boldsymbol{\alpha}}, \tag{7}$$

where $\mathbf{M}$ denotes the memory and where $\boldsymbol{\alpha}$ denotes the $N \times 1$ vector of attention weights. The output of the attention extraction component is computed

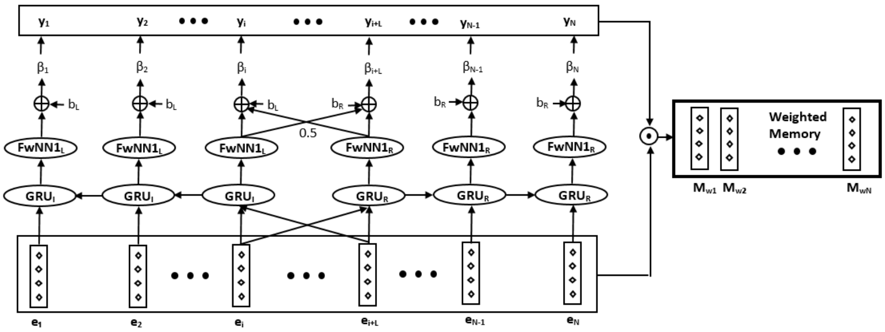

Figure 4: CABASC Context Attention Mechanism.

as follows:

$$\underset{d\times 1}{\mathbf{v}_{as}} = \underset{d\times 1}{\mathbf{v}_{ss}} + \underset{d\times 1}{\mathbf{v}_{s}}\,, \tag{8}$$

where $\mathbf{v}_{as}$ denotes a $d \times 1$ vector with the aspect-specific sentence representation that contains the output of the sentence-level content attention module. The interpretation of this addition is as follows. The first element of $\mathbf{v}_{as}$ combines the first feature of the aspect-specific sentence representation $\mathbf{v}_{ss}$ and the sentence-level representation $\mathbf{v}_s$. The result is processed through the classifier in the same way as for model A, so that the desired output is reached.

3.4 CABASC Model

This section describes the CABASC model. It extends the previously defined baseline model B. The CABASC model hypothesizes that the correlation between context words and the aspect can be used to improve the memory, which is why a **context attention-based memory module** (CAM) is defined. A schematic overview of the CAM module of the CABASC model is displayed in Figure 4. The context attention mechanism is defined as follows. First, the sentence S is divided into a left context and a right context. The left context is denoted as $S_l = \{s_1, ...s_i, ..., s_{i+L}\}$ and consists of the words left of the aspect and the aspect itself. Similarly, the right context is defined as $S_r = \{s_i, ..., s_{i+L}, ...s_N\}$. Their corresponding embeddings are denoted as $\mathbf{E}_l = [\mathbf{e}_1, ..., \mathbf{e}_i, ..., \mathbf{e}_{i+L}]$ and $\mathbf{E}_r = [\mathbf{e}_i, ..., \mathbf{e}_{i+L}, ..., \mathbf{e}_N]$.

The correlation between the context words and the aspect is modeled by two Gated Recurrent Unit (GRU) neural networks. A left GRU$_l$ models the correlations of the context preceding the aspect and a right GRU$_r$ models the correlations of the context following the aspect. GRU$_l$ runs from right to left with input $\mathbf{E}_l$, while GRU$_r$ runs from left to right with input $\mathbf{E}_r$. A GRU consists of an update gate, a reset gate, a candidate hidden state, and a hidden state. At time step t, these four components can be defined by Equations 9, 10, 11, and 12, respectively.

$$\underset{m\times 1}{\mathbf{z}_t} = \sigma(\underset{m\times d}{\mathbf{W}_z}\ \underset{d\times 1}{\mathbf{e}_t} + \underset{m\times m}{\mathbf{U}_z}\ \underset{m\times 1}{\mathbf{h}_{\mathbf{t-1}}} + \underset{m\times 1}{\mathbf{b}_z}), \tag{9}$$

$$\underset{m\times 1}{\mathbf{r}_t} = \sigma(\underset{m\times d}{\mathbf{W}_r}\ \underset{d\times 1}{\mathbf{e}_t} + \underset{m\times m}{\mathbf{U}_r}\ \underset{m\times 1}{\mathbf{h}_{\mathbf{t-1}}} + \underset{m\times 1}{\mathbf{b}_r}), \tag{10}$$

$$\underset{m\times 1}{\tilde{\mathbf{h}}_t} = \tanh(\underset{m\times d}{\mathbf{W}_h}\ \underset{d\times 1}{\mathbf{e}_t} + \underset{m\times m}{\mathbf{U}_h}\ (\underset{m\times 1}{\mathbf{r}_t} \odot \underset{m\times 1}{\mathbf{h}_{\mathbf{t-1}}}) + \underset{m\times 1}{\mathbf{b}_h}), \tag{11}$$

$$\underset{m\times 1}{\mathbf{h}_t} = (\underset{m\times 1}{\mathbf{1}} - \underset{m\times 1}{\mathbf{z}_t}) \odot \underset{m\times 1}{\mathbf{h}_{t-1}} + \underset{m\times 1}{\mathbf{z}_t} \odot \underset{m\times 1}{\tilde{\mathbf{h}}_t}, \tag{12}$$

where $\mathbf{W}_r$, $\mathbf{W}_z$, and $\mathbf{W}_h$ are $m \times d$ weight matrices, $\mathbf{U}_r$, $\mathbf{U}_z$, and $\mathbf{U}_h$ are $m \times m$ parameter matrices, $\mathbf{b}_z$, $\mathbf{b}_r$, and $\mathbf{b}_h$ are $m \times 1$ bias vectors, σ denotes the sigmoid logistic function, $\mathbf{e}_t$ denotes the word embedding of the word at step t, where t denotes the position in the sentence, and $\odot$ denotes element-wise multiplication. The output of the left GRU$_l$ is given by $\mathbf{H}_l = \{\mathbf{h}_{i+L}, ..., \mathbf{h}_i, ..., \mathbf{h}_1\}$ and the right GRU$_r$ produces $\mathbf{H}_r = \{\mathbf{h}_i, ..., \mathbf{h}_{i+L}, ..., \mathbf{h}_N\}$. The attention weights for the left context are computed by a small feed-forward network. Formally, the attention weight for word k in the left context is computed as:

$$\underset{1\times 1}{\beta_{kl}} = \sigma(\underset{1\times m}{\mathbf{w}_{10}}\ \underset{m\times 1}{\mathbf{h}_k} + \underset{1\times 1}{b_5}) + \underset{1\times 1}{b_l}, \tag{13}$$

where $\mathbf{w}_{10}$ denotes a $1 \times m$ weight vector, b_5 a scalar bias, and b_l a basic attention weight, which is regarded as a super parameter. This super parameter is a fixed constant that defines the minimum weight that each word receives from the context attention mechanism. The resulting reversed left attention list is given by $\beta_l = \{\beta_{1l}, ..., \beta_{il}, ..., \beta_{i+L,l}\}$. Similarly, a set of right attentions is computed, where the attention weight of context word g, in the right context is computed as:

$$\underset{1\times 1}{\beta_{gr}} = \sigma(\underset{1\times m}{\mathbf{w}_{11}}\ \underset{m\times 1}{\mathbf{h}_g} + \underset{1\times 1}{b_6}) + \underset{1\times 1}{b_r}, \tag{14}$$

where $\mathbf{w}_{11}$ denotes a $1 \times m$ weight vector, b_6 a scalar bias, and b_r a basic attention weight, which is a super parameter. The full set of attentions weights is obtained by concatenating the left and right attention weights, resulting in $\boldsymbol{\beta} = \{\beta_{1l}, ..., \beta_i, ...\beta_{i+L}, ..., \beta_{Nr}\}$, where $\beta_i, ...\beta_{i+L}$ are the average of the left and right attention weights, computed as follows:

$$\beta_a = \frac{(\beta_{al} + \beta_{ar})}{2}, \tag{15}$$

for $a = i, ..., i + L$. Finally, every memory slice in weighted memory $\mathbf{M}_w = [\mathbf{m}_{w1}, ..., \mathbf{m}_{wN}]$ is computed as follows:

$$\underset{d\times 1}{\mathbf{m}_{wj}} = \underset{d\times 1}{\mathbf{y}_j} \odot \underset{d\times 1}{\mathbf{m}_j}, \tag{16}$$

for $j = 1..., N$ and where $1 \times d$ vector $\mathbf{y}_j$ is obtained by tiling β_j, d times, and $\odot$ denotes element-wise multiplication.

4 Extended CABASC Model

The authors of the CABASC model show that the model achieves outstanding performance on various data sets. With two well-defined mechanisms, the content attention mechanism and the context attention mechanism (the last presented one), the model can discriminate between the importance of different context words. First, the context attention mechanism plays an important role in identifying the relations between the context words and the aspect by taking into account the recurrence in the text. After this, the content attention mechanism generates a probability distribution over all context words, where more important context words receive a higher attention weight.

While the model contains two important and well-defined mechanisms, it also has some limitations. In this section, three shortcomings of the model are identified and solutions are proposed. Three extensions to the CABASC model are introduced, resulting in the CABASC-A, CABASC-B, and CABASC-C models. Each of these three models features an extension that deals with one of the identified shortcomings of the CABASC model. The three new models do not built upon each other, but extend the CABASC model in a particular way and are regarded as separate models. With this setup, the effects of each of the newly proposed mechanisms as compared to the standard CABASC model can be identified. Last, all three extensions are combined into the Extended-CABASC (E-CABASC) model.

Section 4.1 discusses the CABASC-A model, which extends the CABASC model with a context-aware aspect representation to replace the simple average of the word embeddings. Section 4.2 discusses the CABASC-B model, which extends the CABASC model by modifying the content attention mechanism into a multi-dimensional content attention mechanism. Section 4.3 discusses the CABASC-C model, which extends the standard CABASC model with a dimension-wise fusion gate to generate the aspect-specific sentence representation. Last, in Section 4.4, the E-CABASC model is presented, which features all extensions simultaneously.

4.1 CABASC-A Model

In the last few years, many researchers have adopted the idea of an aspect-specific sentence representation, as a solution to sentences with multiple aspects [26, 46]. For example, [45] proposes the target-dependent LSTM, where the aspect embedding is concatenated to the word embeddings. The CABASC model generates an aspect-specific sentence representation by splitting the context into a left context and a right context, which both contain the aspect. Also, the aspect embedding is incorporated in the content attention mechanism. For this, the average of the word embeddings is used in cases where the aspect consists of multiple words. However, we argue that the aspect phrases are not well represented in the CABASC model. Consider the following example sentence:

The chicken and falafel platters were nondescript combinations with fresh leaf salad.

This sentence contains two aspects, being ‘chicken and falafel platters’ and ‘fresh leaf salad’. Taking the average of the word embeddings of these aspects might result in a false representation that has nothing to do with its original meaning. Remarkable is the fact that most researchers do not consider this problem and simply use the average aspect embedding for the aspect representation [26, 46]. 26% and 28% of the aspects consist of multiple words for the 2014 restaurant data set and the 2016 restaurant data set, respectively. Thus, a significant proportion of targets may not be well-represented in many models proposed in the literature, including the CABASC model.

Contrary to the mainstream, the model proposed in [60] contains a rich aspect representation. The authors show that this increases the performance of the model by 3-5%. Their model features a context-aware aspect representation, where the aspect is modeled with a bi-directional LSTM and depends on the left and right contexts. Inspired by this idea and the shortcoming of the current aspect representation in the CABASC model, we propose to replace the aspect representation $\mathbf{v}_a$ with a *context-aware aspect representation* $\mathbf{v}_{ca}$. This approach is different from the one presented in [60] because a bidirectional LSTM is not used in this work. In this way, the number of model parameters is greatly reduced.

A sentence of length N is denoted as $S = \{s_1, ...s_i, ..., s_{i+L}, ...s_N\}$, where s_j denotes word j and where the aspect is given by L words, $S_a = \{s_i, ..., s_{i+L}\}$. Following the notation of the CABASC model, the word embeddings of S_a are given by $\mathbf{E}^a = [\mathbf{e}_i, ..., \mathbf{e}_{i+L}]$. This is copied to memory $\mathbf{M}^a$ such that $\mathbf{E}^a = \mathbf{M}^a$. Importance scores for every word in the aspect are computed using a bi-linear scoring function. The reason for this is that it introduces only one weighting matrix, resulting in fewer parameters compared to an additive scoring function. An importance score c_j^a of word j in aspect S_a is computed for every word j in the aspect as follows:

$$\underset{1\times 1}{c_j^a} = \tanh(\underset{1\times d}{\mathbf{e}_j^T}\ \underset{d\times d}{\mathbf{W}_{12}}\ \underset{d\times 1}{\mathbf{v}_s} + \underset{1\times 1}{b_7}), \tag{17}$$

where $\mathbf{v}_s$ denotes the average sentence embedding, $\mathbf{W}_{12}$ denotes a $d \times d$ weight matrix, and b_7 denotes a scalar bias. The importance score depends on the aspect word itself and the average sentence representation $\mathbf{v}_s$. With this construction, the attention mechanism is context-aware because the attention mechanism takes into account the context of the aspect. The importance scores for every word in S_a are processed through a softmax as follows:

$$\alpha_j^a = \frac{\exp(c_j^a)}{\sum_{i=1}^{N} \exp(c_i^a)} \in [0, 1], \tag{18}$$

where α_j^a denotes the attention weight of word j in aspect S_a. If the aspect consists of a single word, the attention weight receives a weight of 1 by definition. If the aspect consists of multiple words, each word is weighted such that the sum of all attention weights equals 1. The context-aware aspect representation $\mathbf{v}_{ca}$ is computed as follows:

$$\underset{d\times 1}{\mathbf{v}_{ca}} = \underset{d\times L}{\mathbf{M}^a}\ \underset{L\times 1}{\boldsymbol{\alpha}^a}, \tag{19}$$

where $\mathbf{M}^a$ is the vector representation of the aspect, and $\boldsymbol{\alpha}^a$ contains all attention scores α_j^a. Here, L denotes the length of the aspect. The advantage of the above approach is two-fold. First, the aspect representation is context-aware, in the sense that the context influences the aspect representation. Second, the attention mechanism generates a weighted aspect representation, which is a richer representation than a simple average. The context-aware aspect representation $\mathbf{v}_{ca}$ directly replaces the aspect representation $\mathbf{v}_a$ and is processed in exactly the same way as $\mathbf{v}_a$ in the CABASC model.

4.2 CABASC-B Model

The second extension that we propose considers a change to the content attention mechanism. The CABASC model contains a content attention mechanism that produces attention scores $\boldsymbol{\alpha}$ that weight every context word by its importance towards the aspect. The currently defined mechanism works as follows. Given a sequence of words $S = \{s_1, ...s_i, ..., s_{i+L}, ...s_N\}$ encoded in a $d \times N$ (weighted) memory matrix $\mathbf{M}_w = [\mathbf{m}_{w1}, ..., \mathbf{m}_{wi}, ..., \mathbf{m}_{w,i+L}, ..., \mathbf{m}_{wN}]$, an importance score is calculated for every memory slice $\mathbf{m}_{wj}$ in $\mathbf{M}_w$ as follows:

$$\underset{1\times1}{c_j} = \underset{1\times m}{\mathbf{w}_6} \tanh(\underset{m\times d}{\mathbf{W}_7}\ \underset{d\times1}{\mathbf{m}_{wj}} + \underset{m\times d}{\mathbf{W}_8}\ \underset{d\times1}{\mathbf{v}_a} + \underset{m\times d}{\mathbf{W}_9}\ \underset{d\times1}{\mathbf{v}_s} + \underset{m\times1}{\mathbf{b}_4}). \tag{20}$$

The scores c_j are processed using a softmax function as follows:

$$\alpha_j = \frac{\exp(c_j)}{\sum_{i=1}^{N} \exp(c_i)} \in [0, 1]. \tag{21}$$

After this, the aspect-specific sentence representation is computed as follows:

$$\underset{d\times1}{\mathbf{v}_{as}} = \underset{d\times N}{\mathbf{M}}\ \underset{N\times1}{\boldsymbol{\alpha}}\ . \tag{22}$$

This content attention mechanism produces an aspect-specific sentence representation $\mathbf{v}_{as}$, which is a weighted sum of all words in the sentence. One particular feature of this attention mechanism is that it weights the entire word embedding of a word j by a single weight α_j while not discriminating between different features of the word. This shortcoming can be directly derived from how the sentence representation is computed, as shown in Equation 22. As a result, the current content attention mechanism does not discriminate between important and less important features of a word. As a solution to this, the content attention mechanism is extended to a *multi-dimensional* content attention mechanism.

[42] proposes the multi-dimensional attention mechanism, where words are weighted at the feature level. Rather than computing a single scalar score for word j, a feature-wise score vector is computed for word j as follows:

$$\underset{d\times1}{\mathbf{c}_j^*} = \underset{d\times m}{\mathbf{W}_{13}} \tanh(\underset{m\times d}{\mathbf{W}_7}\ \underset{d\times1}{\mathbf{m}_j} + \underset{m\times d}{\mathbf{W}_8}\ \underset{d\times1}{\mathbf{v}_a} + \underset{m\times d}{\mathbf{W}_9}\ \underset{d\times1}{\mathbf{v}_s} + \underset{m\times1}{\mathbf{b}_4}) + \underset{d\times1}{\mathbf{b}_8}, \tag{23}$$

where $\mathbf{W}_{13}$ replaces weight vector $\mathbf{w}_6$ in Equation 20 by a $d \times m$ weight matrix and where $\mathbf{b}_8$ is an additional $d \times 1$ bias term. The resulting importance scores are now given by the $d \times 1$ vector $\mathbf{c}_j^*$, where element c_{jk}^* of vector $\mathbf{c}_j^*$ defines the weight of feature k of word j. The attention weight of feature k of word j is then computed as follows:

$$\alpha_{jk}^* = \frac{\exp(c_{jk}^*)}{\sum_{i=1}^{N} \exp(c_{ik}^*)} \in [0,1], \tag{24}$$

for $k = 1, ..., d$ and $j = 1, ..., N$. This softmax operation normalizes every feature in the sentence such that the attention weights of all features k for a word in sentence S add up to one. Let $d \times 1$ vector $\boldsymbol{\alpha}_j^*$ denote the vector that contains the attention weights of word j. The aspect-specific sentence representation of $\mathbf{M}_w$ can be computed as follows:

$$\underset{d\times 1}{\mathbf{v}_{ss}} = \sum_{j=1}^{N} \underset{d\times 1}{\boldsymbol{\alpha}_j^*} \odot \underset{d\times 1}{\mathbf{m}_{wj}}. \tag{25}$$

$\mathbf{v}_{ss}$ is the result of element-wise multiplication (denoted by $\odot$) of the attention weights and the features, summed over all words in the sentence. $\mathbf{v}_{ss}$ is then processed in the same way as in the CABASC model.

An additional feature of multi-dimensional attention is that it deals with polysemy in natural language. Standard word embeddings suffer from the problem that the different word senses of a word with multiple meanings are not identified. The standard attention mechanism identifies the importance of a word by considering the aspect and the full sentence representation. Multi-dimensional attention deals even better with polysemy in natural language, because the importance of a word is weighted at the feature level. Since the attention mechanism takes into account the aspect-representation $\mathbf{v}_a$ and the average sentence representation $\mathbf{v}_s$, the features which describe the word best in the context are selected.

4.3 CABASC-C Model

The last modification of the CABASC model that is introduced in this work considers a more advanced and suitable mechanism to compute the aspect-specific sentence representation $\mathbf{v}_{as}$. The aspect-specific sentence representation of the CABASC model is computed as follows. First, a sentence-level weighted memory representation is obtained:

$$\underset{d\times 1}{\mathbf{v}_{ss}} = \underset{d\times N}{\mathbf{M}} \; \underset{N\times 1}{\boldsymbol{\alpha}} \, . \tag{26}$$

After this, the average sentence embedding $\mathbf{v}_s$ is added to the sentence representation:

$$\underset{d\times 1}{\mathbf{v}_{as}} = \underset{d\times 1}{\mathbf{v}_{ss}} + \underset{d\times 1}{\mathbf{v}_{s}} \, , \tag{27}$$

which results in the aspect-specific sentence representation $\mathbf{v}_{as}$. The choice of adding $\mathbf{v}_{ss}$ and $\mathbf{v}_s$ seems quite arbitrary. A clear disadvantage of this choice is that the sentence-level weighted memory representation $\mathbf{v}_{ss}$ and the average sentence embedding $\mathbf{v}_s$ are equally weighted. We argue that this is a very poor choice because the sentence-level weighted memory representation $\mathbf{v}_{ss}$ is a very detailed representation of the sentence after it has been processed through the context attention mechanism and the content attention mechanism. On the other hand, the average sentence embedding is a simple average that is not weighted in any way. Still, both representations receive the same weight and are considered to be equally important.

It is for this reason that we propose to combine the two vectors using a gated neural network that fuses the two representations, rather than via a simple addition. The dimension-wise fusion is inspired by [42] and can be defined by the following two equations:

$$\underset{d\times 1}{\mathbf{F}} = \sigma(\underset{d\times d}{\mathbf{W}_{14}} \underset{d\times 1}{\mathbf{v}_{ss}} + \underset{d\times d}{\mathbf{W}_{15}} \underset{d\times 1}{\mathbf{v}_{s}} + \underset{d\times 1}{\mathbf{b}_{9}}), \tag{28}$$

$$\underset{d\times 1}{\mathbf{v}_{as}} = \underset{d\times 1}{\mathbf{F}} \odot \underset{d\times 1}{\mathbf{v}_{ss}} + (\underset{d\times 1}{\mathbf{1}} - \underset{d\times 1}{\mathbf{F}}) \odot \underset{d\times 1}{\mathbf{v}_{s}} \, , \tag{29}$$

where $\mathbf{W}_{14}$ and $\mathbf{W}_{15}$ are $d \times d$ weight matrices, and $\mathbf{b}_9$ is a $d \times 1$ bias vector. σ denotes the logistic sigmoid function and $\odot$ denotes element-wise multiplication. The fusion gate $\mathbf{F}$ computes a score that is scaled between 0 and 1 by the sigmoid function, which denotes the relative importance of the features of $\mathbf{v}_{ss}$ as compared to the features of $\mathbf{v}_s$. Following Equation 28, one can see that the importance of feature k depends on all other features, hence the term dimension-wise fusion.

4.4 E-CABASC Model

This section summarizes the Extended CABASC model. This new model extends the CABASC model with the three previously discussed extensions. Figure 5 displays an overview of the E-CABASC model. All extensions are colored red (light grey in black and white printing).

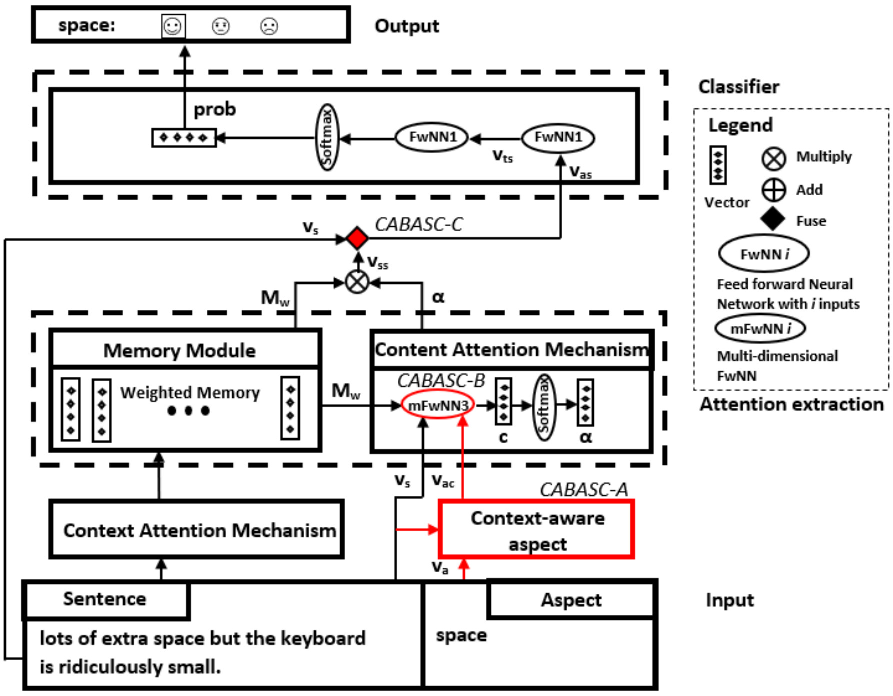

Figure 5: Extended CABASC Framework.

The framework is identical to the standard CABASC framework, except for the context-aware aspect representation, the multi-dimensional content attention mechanism, and the fusion gate. The CABASC-A extension replaces the aspect representation with a context-aware aspect representation. The CABASC-B extension uses multi-dimensional attention. The CABASC-C attention uses a fusion gate to select the importance of the aspect-specific sentence representation and the complete sentence representation.

5 Training Settings Neural Networks

This section describes the training details of the neural attention models. Every model is trained by minimizing the cross-entropy loss-function:

$$L = -\sum_{(s,a)\in T}\sum_{c\in C} P_c^g(s,a)\log(P_c(s,a)), \tag{30}$$

where T denotes the set of all the sentence-aspect pairs (s, a) in the training sample, C denotes the set of all sentiment categories c, $P_c(s, a)$ denotes the probability of predicting sentiment category c for sentence-aspect pair (s, a), and where $P_c^g(s, a)$ is equal to 1 if c is the correct sentiment and 0 if not. This loss function minimizes the difference between the estimated probability distribution and the true labels (g denotes the gold standard).

The parameters of the model are optimized using back-propagation with the Gradient Descent Optimizer. Hyper-parameters are tuned using a validation set with a random 20% sample of the training data. To avoid over-fitting, the dropout technique is used. We perform a grid search over the following parameter values: learning rate of 0.001 or 0.01 and keep-probability of 0.5 or 0.7, as suggested in [26]. The parameters that are found best on the validation set are a learning rate of 0.001 and a keep-probability of 0.7. Dropout is applied to all inputs in every network, including the biases. Although performing drop-out on the biases is not common, this provided a noticeable improvement to the performance of the networks. Similarly as in [26], the super-parameters b_l and b_r are set equal to 0.5. All initial parameter values are randomized with a zero mean normal distribution with a variance of 0.05, as suggested in [26]. The 300-dimensional pre-trained GloVe word embeddings are used with a vocabulary size of 1.9M words. Words that are not in the vocabulary are initialized with a zero vector, as suggested in [26]. During training, the network is updated via mini-batches of size 128 [18]. The number of neurons in baseline models A, B, C, and CABASC is set to d, so $m = d$, as proposed in [26]. All models are implemented in Python using Tensorflow for which the source code can be found online[2].

6 Results and Evaluation

This section evaluates the models discussed in this paper. The data used in this paper considers the widely used restaurant data sets from the SemEval 2014 task 4 [37] and the SemEval 2016 task 5 [36]. Since the SemEval 2015 data is a subset of the SemEval 2016 data, the models have been evaluated only on the 2016 data set only. The data sets consider data for sentence-level ABSC. Subsection 6.1 discusses the main results of this paper. Subsection 6.2 provides an evaluation of the parameters of the proposed E-CABASC model. Lastly, in Subsection 6.3, an evaluation is provided of the attention weights of the CABASC and E-CABASC models.

[2] https://github.com/jjvanderuitenbeek/CABASC-Model

6.1 Main Results

In this section, the main results of the neural attention models are discussed. First, the baseline A and B models and the CABASC model are compared. Then, the results of the Extended CABASC (E-CABASC) model are evaluated. The separate components of this model are included in the models CABASC-A, CABASC-B, and CABASC-C. Table 1 displays the accuracies of the different models for the restaurant 2014 and restaurant 2016 data set. Similar accuracies as in [26] are obtained for the baseline models and the CABASC model on the 2014 data set. Minor differences are most likely due to different pre-processing of the data, which is not specified by the authors. The results for the 2016 data set were not reported in [26].

Table 1: Accuracies of Neural Attention Models.

Model	Rest. 2014	Rest. 2016
Majority vote	65.0	74.3
Baseline model A	78.5	83.5
Baseline model B	78.9	84.0
CABASC	80.4	85.1
CABASC-A	80.7	85.3
CABASC-B	80.7	85.2
CABASC-C	80.6	85.3
E-CABASC	**80.9**	**85.4**

A few observations can be made from Table 1. As compared to the majority vote (positive), the overall results of the other models are better on the 2014 data set than on the 2016 data set. This is expected, since the 2014 data set contains more training instances, making it easier to train deep learning models. However, the relative number of positive labels in the test set is much higher for the 2016 data set than the 2014 data set. Because of this, all models perform much better on this data set in terms of absolute accuracy. Notable is the fact that the number of training instances for 2016 is about half the number of training instances for 2014, but much better results are achieved. This fact is remarkable since generally the performance of neural networks tends to increase when the number of training instances increases. However, since the model predicts positive with the default initialization, the results must be compared with the majority vote (positive). Looking at this difference, it can be observed that the differences are much higher on the 2014 data set.

Baseline model B yields around 0.4 percentage points improvement over baseline model A. From this, it can be concluded that adding the sentence-level information in baseline model B improves the model because the information of the sentence as a whole contains information that is not captured in baseline model A. The CABASC model outperforms all baseline models. The results affirm the effectiveness of the context attention mechanism in the CABASC model as it causes an increase in accuracy of around 1.1-1.5 percentage points as compared to baseline model B. These results indicate that modeling the correlation between context words and the aspect via the context attention mechanism adds much value to the model because an aspect-specific customized memory is created for every aspect in the sentence. Experiments have been run with LSTM units instead of GRU units in the CABASC model, but a minor decrease in accuracy of 0.1 percentage points was observed. Hence, the choice of GRU units is justified.

It can be observed from Table 1 that all proposed extensions to the CABASC model yield improvement on both data sets. The CABASC-A model performs around 0.2-0.3 percentage points better than the CABASC model, indicating the need for a rich aspect representation. Simply taking the average over the word embeddings of multi-word aspects is not enough and taking a weighted average based on the average sentence embedding improves the results. The CABASC-B model yields an improvement of around 0.1-0.3 percentage points over the CABASC model. This result indicates that multi-dimensional attention, where the words are weighted feature-wise rather than per word, is a better attention mechanism for this task. It can also be observed that the CABASC-C model yields an improvement of 0.2 percentage points over the CABASC model. It helps the model distinguish better between the importance of the aspect-specific sentence representation and the average sentence representation.

The improvements of the E-CABASC model compared to the standard CABASC model are larger on the 2014 data set than on the 2016 data set. The size of the data set most likely plays an important role in the effectiveness of the extensions. Because the 2014 data set is much larger, better relative results are reported. Since the extensions lead to more parameters, they are less effective on small data sets. Yet, we conclude that the E-CABASC model outperforms all other models in this paper for both data sets. These results show the effectiveness of all three extensions.

As presented in [26], the CABASC model outperforms several notable deep learning and attention models. For example, deep memory networks [46] and interactive attention networks [27]. By extension, the E-CABASC model also outperforms all of these. Yet, other types of models have been shown to provide even better results [4]. One example of this is the hybrid deep learning models that combine deep learning models with knowledge bases, such as dictionaries, ontologies, or discourse trees [4]. The usage of such knowledge bases has been shown to provide improved performance, but knowledge bases are not always available for every problem in every language [4]. Another example is the transformer model [49]. This type of model produces state-of-the-art results in most fields where deep learning is popular, including ABSA [23]. Yet, transformer models use vast amounts of training parameters, requiring significant amounts of memory and computation. The E-CABASC model has an advantage here, as explained in the next subsection.

6.2 Parameter Evaluation

For a more in-depth analysis of the CABASC and E-CABASC models, in this section, the number of parameters and the effect of the number of neurons in the hidden layers are evaluated. Table 2 displays the number of parameters for the baseline models, CABASC model, and E-CABASC model.

Table 2: Number of Parameters of Neural Attention Models.

Model	Attention mechanism	CAM	Deep layers	Fusion	Total
Baseline A	180,600	-	91,203	-	271,803
Baseline B	270,600	-	91,203	-	361,803
CABASC	270,600	541,502	91,203	-	903,305
E-CABASC	450,601	541,502	91,203	180,300	1,263,606

It can be observed that the number of parameters for the models is relatively small as compared to other ABSC models. For example, a standard LSTM unit with biases and input embeddings of size 300 contains 721,200 parameters. A bi-directional LSTM model already has more parameters than the CABASC and E-CABASC models. The highly popular transformer models can even use hundreds of millions of parameters [53]. This makes the CABASC and E-CABASC models easier to train than other models.

As suggested in [26], $m = d = 300$ neurons were used in the hidden layers of both CABASC and E-CABASC. To show that this is the optimal number of neurons, Table 3 displays the accuracies for different numbers of neurons in the hidden layers of the CABASC model.

Table 3: Accuracy vs. Number of Neurons for CABASC.

No. of Neurons:	$m = 100$	$m = 200$	$m = 300$
2014 data set	80.0	80.3	80.4
2016 data set	85.1	85.2	85.1

It can be observed that the number of neurons in the hidden layers affects the performance of the model. Fine-tuning the number of neurons can make a difference of around 0.5 percentage points for the accuracy. Choosing too many neurons results in overfitting the data, so it is not always recommended to choose the number of neurons equal to the dimensions of the word vectors. However, in this case, it can be confirmed that the number of neurons is chosen correctly for CABASC ($m = d = 300$). For fair comparison purposes, this parameter was therefore also used for E-CABASC.

6.3 Attention

In this section we analyze the performance of the different attention mechanisms of the neural attention models by considering some example cases. We evaluate an example sentence for the CABASC model and the E-CABASC model. The following figures display the attention weights and context attention weights obtained from the CABASC model.

Attention Weights CABASC

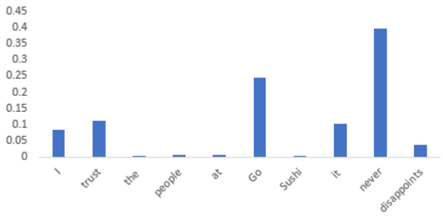

Figure 6: Content Attention

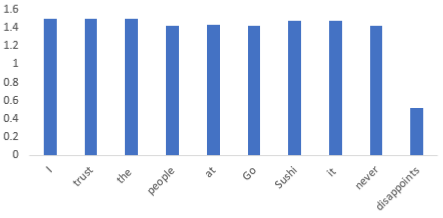

Figure 7: Context Attention

An interesting observation can be made from the visualized attention weights. We observe that the most focused words are not given by sentiment expressions anymore, but that other words are focused on as well. The most probable cause for this is that the model is overfitting, such that the final results in terms of accuracy are good, but not always completely interpretable. Ideally, the focused words would have a clear interpretation, but this is not the case in this example. Another observation that we make is that the sentiment word ‘disappoint’ receives a very low attention. While this word carries a lot of sentiment, we observe that the context attention mechanism of the model learns that after the word ‘never’ the word ‘disappoints’ no longer has a negative influence, giving it a low attention weight.

Figure 8 and Figure 9 display the content attention weights and context attention weights for the E-CABASC model. To obtain the content attention weight of a word of the E-CABASC model, we sum the weights of all features of that word. The scale of the weights obtained by the E-CABASC model is different than the scale of the weights obtained with the CABASC, because of the multi-dimensional attention mechanism. In this case the attention weights sum up to one per feature, which means that the sum of the multi-dimensional attention weights sums up to 300, because of the 300-dimensional word vectors.

Attention Weights E-CABASC

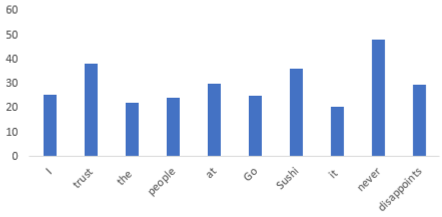

Figure 8: Content Attention

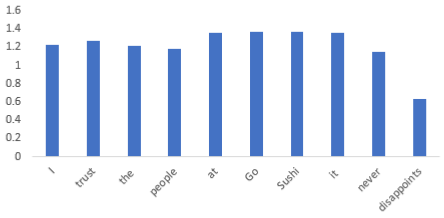

Figure 9: Context Attention

Comparing the content attention weights obtained by the E-CABASC model with the weights obtained by the CABASC model, we observe that the content attention weights of the E-CABASC model have a more equal distribution, indicating that the model does give every word some importance. Similar to the CABASC model, the word ‘never’ receives a high attention weight. Different from the CABASC model, the word ‘trust’ receives a higher attention weight than other words. This word is very important for determining the sentiment and the E-CABASC model identifies this better than the CABASC model. Comparing the context attention weights of the CABASC and E-CABASC model, we see the same pattern for both models. The word 'disappoints' receives a low weight, after the word 'never'.

7 Conclusion

Since the explosive growth of the Word Wide Web, a large amount of information has become publicly available [14]. Sentiment analysis allows for the automatic analysis of large numbers of opinions shared online through, for example, Web reviews. ABSC considers sentiment analysis at a highly fine-grained level. The goal of ABSC is to predict the sentiment expressed towards every aspect in a sentence. In this paper, the CABASC model [26] has been discussed, which is a highly successful neural attention model that features a customized aspect-specific memory where every context word is weighted with respect to its importance towards the sentiment of the aspect.

Three shortcomings of the CABASC model have been identified and the Extended CABASC (E-CABASC) model has been proposed, which attempts to solve these problems. The model incorporates a context-aware aspect representation, which explicitly models the aspect using context features. In addition, the content attention mechanism is adapted to a multi-dimensional attention mechanism. Multi-dimensional attention weights the sentence feature-wise and deals with polysemy in natural language better than standard attention. Last, an aspect-specific sentence representation is constructed by fusing two sentence representations using a dimension-wise fusion gate that flexibly combines both representations. The resulting E-CABASC model is shown to outperform the CABASC model.

One of the main contributions of this paper is that the three proposed model extensions have been shown to all improve the model individually. As such, we would recommend researching the implementation of the three extensions into other attention models. The context-aware aspect representation and aspect-specific sentence representation can be implemented in most ABSA models. Furthermore, the multi-dimensional attention mechanism is a technique that can be applied in almost any attention model, even for different problems outside the field of ABSA and NLP, such as computer vision and time series analysis. These extensions can also be implemented in the previously mentioned state-of-the-art hybrid and transformer models.

Directions for future research with regards to the E-CABASC model include modeling all aspects in a sentence simultaneously. This could enhance the model to optimally link words together. Also, we suggest to investigate if the GloVe word embeddings are optimal for this problem or results can be improved using different word embeddings like sentiment-oriented word embeddings [28] or IWV embeddings [40]. Furthermore, in this paper we ignored the implicit targets, because the attention model explicitly models the aspect. An interesting direction to investigate is how implicit targets can be modeled using proxy words in sentences and neural attention models. As mentioned, the inclusion of knowledge bases in deep learning models is a technique that is gaining popularity due to improvements in performance [4, 51, 61, 48]. As such, another interesting subject to explore would be combining the E-CABASC model with external knowledge bases.

We would further recommend that future research in ABSA focuses on dealing with complex sentence structures and transition words. One of the main challenges of ABSA is to correctly link sentiment expressions and aspects in sentences with multiple aspects and multiple sentiment expressions. Two other complex linguistic phenomena that are important to consider for future research are thwarting and sarcasm. These two concepts that introduce contradictions of sentiments in texts pose significant challenges for sentiment analysis models [38, 30]. Yet, sarcasm and thwarting are highly prevalent on social media and any other platform on the Internet where opinions and sentiments are expressed.

References

[1] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, 2017. https://openreview.net/forum?id=BJh6Ztuxl.

[2] W. Ansar, S. Goswami, A. Chakrabarti, and B. Chakraborty. An efficient methodology for aspect-based sentiment analysis using BERT through refined aspect extraction. *Journal of Intelligent & Fuzzy Systems*, 40(5):9627–9644, 2021.

[3] G. Brauwers and F. Frasincar. A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[4] G. Brauwers and F. Frasincar. A survey on aspect-based sentiment classification. *ACM Computing Surveys*, 2022.

[5] T. Brychcín, M. Konkol, and J. Steinberger. UWB: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 817–822. ACL, 2014. https://www.aclweb.org/anthology/S14-2145.

[6] P. Chen, Z. Sun, L. Bing, and W. Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 452–461. ACL, 2017. https://www.aclweb.org/anthology/D17-1047.

[7] P. Chen, B. Xu, M. Yang, and S. Li. Clause sentiment identification based on convolutional neural network with context embedding. In *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2016)*, pages 1532–1538. IEEE, 2016.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 4171–4186. ACL, 2019. https://www.aclweb.org/anthology/N19-1423.

[9] V. Diviya Prabha and R. Rathipriya. Sentimental analysis using capsule network with gravitational search algorithm. *Journal of Web Engineering*, pages 762–778, 2020.

[10] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, volume 2, pages 49–54. ACL, 2014. https://www.aclweb.org/anthology/P14-2009.

[11] G. D'Aniello, M. Gaeta, and I. La Rocca. KnowMIS-ABSA: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis. *Artificial Intelligence Review*, pages 1–32, 2022.

[12] F. Fan, Y. Feng, and D. Zhao. Multi-grained attention network for aspect-level sentiment classification. In *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 3433–3442. ACL, 2018. https://www.aclweb.org/anthology/D18-1380.

[13] R. Feldman. Techniques and applications for sentiment analysis. In *Communications of the ACM*, volume 56, pages 82–89, 2013.

[14] J. Gantz and D. Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. In *IDC iView: IDC Analyze the Future*, volume 2007, pages 1–16. EMC, 2012.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *28th Annual Conference on Neural Information Processing Systems (NIPS 2014)*, pages 2672–2680. Curran Associates, Inc., 2014. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[16] M. Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, volume 37, pages 448–456. JMLR, 2015. http://proceedings.mlr.press/v37/ioffe15.html.

[19] S. Jebbara and P. Cimiano. Aspect-based relational sentiment analysis using a stacked neural network architecture. arXiv preprint arXiv:1709.06309, 2017.

[20] D. Jurafsky and J. H. Martin. *Speech and Language Processing*, volume 3. Pearson, 2014.

[21] A. Karimi, L. Rossi, and A. Prati. Adversarial training for aspect-based sentiment analysis with BERT. In *25th International Conference on Pattern Recognition (ICPR 2020)*, pages 8797–8803. IEEE, 2021.

[22] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442. ACL, 2014. https://www.aclweb.org/anthology/S14-2076.

[23] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems*, 235:107643, 2022.

[24] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations (ICLR 2017)*, 2017.

[25] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer, 2012.

[26] 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 2018)*, pages 1023–1032. ACM, 2018.

[27] D. Ma, S. Li, X. Zhang, and H. Wang. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 4068–4074. AAAI Press, 2017. https://doi.org/10.24963/ijcai.2017/568.

[28] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2011)*, volume 1, pages 142–150. ACL, 2011.

[29] M. V. Mäntylä, D. Graziotin, and M. Kuutila. The evolution of sentiment analysis - a review of research topics, venues, and top cited papers. In *Computer Science Review*, volume 27, pages 16–32. Elsevier, 2018. http://www.sciencedirect.com/science/article/pii/S1574013717300606.

[30] D. Maynard and M. Greenwood. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *9th International Conference on Language Resources and Evaluation (LREC 14)*, pages 4238–4243. ELRA, 2014.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013)*, 2013.

[32] M. S. Mubarok, Adiwijaya, and M. D. Aldhi. Aspect-based sentiment analysis to review products using naïve bayes. In *AIP Conference Proceedings*, volume 1867, page 020060. AIP Publishing, 2017. https://aip.scitation.org/doi/abs/10.1063/1.4994463.

[33] T. H. Nguyen and K. Shirai. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2509–2514. ACL, 2015.

[34] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods*

in Natural Language Processing (EMNLP 2002), volume 10, pages 76–86. ACL, 2002. https://www.aclweb.org/anthology/W02-1011.

[35] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543. ACL, 2014. https://www.aclweb.org/anthology/D14-1162.

[36] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, et al. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, pages 19–30. ACL, 2016. https://www.aclweb.org/anthology/S16-1002.

[37] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35. ACL, 2014. https://www.aclweb.org/anthology/S14-2004.

[38] A. Ramteke, A. Malu, P. Bhattacharyya, and J. S. Nath. Detecting turnarounds in sentiment analysis: Thwarting. In *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 860–865. ACL, 2013.

[39] K. Ravi and V. Ravi. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. In *Knowledge-Based Systems*, volume 89, pages 14–46. Elsevier, 2015. http://www.sciencedirect.com/science/article/pii/S0950705115002336.

[40] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi. Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117:139–147, 2019.

[41] K. Schouten, O. Weijde, F. Frasincar, and R. Dekker. Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data. *IEEE Transactions on Cybernetics*, 48(4):1263–1275, 2017.

[42] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang. DiSAN: Directional self-attention network for RNN/CNN-free language understanding. In *32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 5446–5455. AAAI Press, 2018. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16126.

[43] A. Shoukry and A. Rafea. Machine learning and semantic orientation ensemble methods for Egyptian telecom tweets sentiment analysis. *Journal of Web Engineering*, pages 195–214, 2020.

[44] C. Sun, L. Huang, and X. Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 380–385. ACL, 2019. https://www.aclweb.org/anthology/N19-1035.

[45] D. Tang, B. Qin, X. Feng, and T. Liu. Effective LSTMs for target-dependent sentiment classification. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pages 3298–3307. ACL, 2016. https://www.aclweb.org/anthology/C16-1311.

[46] D. Tang, B. Qin, and T. Liu. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. ACL, 2016.

[47] Y. Tay, L. A. Tuan, and S. C. Hui. Dyadic memory networks for aspect-based sentiment analysis. In *Proceedings of the 26h Conference on Information and Knowledge Management (CIKM 2017)*, pages 107–116. ACM, 2017. https://doi.org/10.1145/3132847.3132936.

[48] M. M. Truşcă, D. Wassenberg, F. Frasincar, and R. Dekker. A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention. In *International Conference on Web Engineering*, volume 12128 of *LNCS*, pages 365–380. Springer, 2020.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, pages 5998–6008. Curran Associates, Inc., 2017. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[50] Y. Wang, M. Huang, X. Zhu, and L. Zhao. Attention-based LSTM for aspect-level sentiment classification. In *2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 606–615. ACL, 2016. https://www.aclweb.org/anthology/D16-1058.

[51] H. Wu, Z. Zhang, S. Shi, Q. Wu, and H. Song. Phrase dependency relational graph attention network for aspect-based sentiment analysis. *Knowledge-Based Systems*, 236:107736, 2022.

[52] H. Xu, B. Liu, L. Shu, and P. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 2324–2335. ACL, 2019. https://www.aclweb.org/anthology/N19-1242.

[53] H. Xu, B. Liu, L. Shu, and P. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 2324–2335. ACL, 2019.

[54] L. Xu, J. Lin, L. Wang, C. Yin, and J. Wang. Deep convolutional neural network based approach for aspect-based sentiment analysis. In *Advanced Science and Technology Letters*, volume 143, pages 199–204. AST, 2017.

[55] Q. Xu, L. Zhu, T. Dai, and C. Yan. Aspect-based sentiment classification with multi-attention network. *Neurocomputing*, 388:135–143, 2020.

[56] C. Yang, H. Zhang, B. Jiang, and K. Li. Aspect-based sentiment analysis with alternating coattention networks. *Information Processing Management*, 56(3):463–478, 2019. http://www.sciencedirect.com/science/article/pii/S0306457318306344.

[57] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.

[58] B. Zhang, D. Xiong, J. Su, and Y. Qin. Alignment-supervised bidimensional attention-based recursive autoencoders for bilingual phrase representation. *IEEE Transactions on Cybernetics*, 50(2):503–513, 2020.

[59] L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4), 2018. https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1253.

[60] S. Zheng and R. Xia. Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention. arXiv preprint arXiv:1802.00892, 2018.

[61] Q. Zhong, L. Ding, J. Liu, B. Du, H. Jin, and D. Tao. Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. *arXiv preprint arXiv:2201.04831*, 2022.

Biographies

Jonathan van de Ruitenbeek received the B.S. degree in econometrics and operations research in 2017 and the M.S. degree in econometrics and management science in 2018 from Erasmus University Rotterdam, Rotterdam, the Netherlands. He presently works as a Data Specialist at ABN AMRO Verzekeringen, where he is responsible for the data delivery to the risk management.

Flavius Frasincar received the M.S. degree in computer science, in 1996, and the M.Phil. degree in computer science, in 1997, from Politehnica University of Bucharest, Bucharest, Romania, and the P.D.Eng. degree in computer science, in 2000, and the Ph.D. degree in computer science, in 2005, from Eindhoven University of Technology, Eindhoven, the Netherlands.

Since 2005, he has been an Assistant Professor in computer science at Erasmus University Rotterdam, Rotterdam, the Netherlands. He has published in numerous conferences and journals in the areas of databases, Web information systems, personalization, machine learning, and the Semantic Web. He is a member of the editorial boards of Decision Support Systems, International Journal of Web Engineering and Technology, and Computational Linguistics in the Netherlands Journal, and co-editor-in-chief of the Journal of Web Engineering. Dr. Frasincar is a member of the Association for Computing Machinery.

Gianni Brauwers received the B.S. degree in econometrics and operations research in 2019 and the M.S. degree in econometrics and management science in 2021 from Erasmus University Rotterdam, Rotterdam, the Netherlands. From 2019 till 2020, he was a Research Assistant at Erasmus University Rotterdam, focusing his research on neural attention models and sentiment analysis. He currently works as a researcher at ABF Research.