Weakly-Supervised Sentence-Based Aspect Category and Sentiment Classification

Olaf Wallaart[a], Flavius Frasincar[a,*], Finn van der Knaap[a]

[a]*Erasmus University Rotterdam, PO Box 1738, 3000 DR, Rotterdam, the Netherlands*

Abstract

Sentiment analysis extracts the sentiment of content creators, enabling users to easily gain valuable insights from such data. Most existing methods rely on supervised learning approaches using labeled data. However, the retrieval of such labeled training data is difficult and expensive, especially for new domains and/or languages. This work focuses on simultaneously detecting aspect categories and sentiment polarities for a given sentence in a weakly-supervised setting. Two methods are proposed that combine an unsupervised labeling algorithm with a neural network architecture. The first proposed two-step model (SB-ASC) takes seed sentences as input for the labeling algorithm. By leveraging the power of pre-trained Sentence-BERT embeddings, the method is able to understand the contextual meaning of sentences to create a high-quality labeled dataset. This dataset is used by a class imbalance-robust BERT-based neural network that jointly learns latent features of aspect categories and the corresponding sentiment. The second proposed method (WB-ASC) uses the same neural network structure but takes seed words instead of seed sentences as input for the labeling algorithm. We conclude that SB-ASC outperforms WB-ASC as well as baselines and state-of-the-art weakly-supervised methods for aspect sentiment detection, achieving F1 scores for aspect category detection of 71.35%, 86.99%, and 73.86%, and F1 scores for sentiment classification of 89.24%, 89.98%, and 75.58% for the SemEval 2016 restaurant-5, restaurant-3, and laptop datasets, respectively. Furthermore, using domain-specific contextual language models boosts performance.

Keywords: Aspect-based sentiment analysis, Weakly-supervised learning, Neural network, Sentence-BERT, Focal loss

1. Introduction

The amount of unstructured data generated by users on the Web is growing, with the global amount of digital data predicted to exceed 175 zettabytes by 2025 [1]. With this trend, the importance of techniques to structure, analyze, and interpret unstructured data continues to grow. A relevant source of unstructured data for organizations, but specifically for retailers, is digitally retrieved customer review data [2, 3, 4]. Reviews are highly relevant for retailers as they entail information about customer satisfaction with respect to existing

*Corresponding author; tel: +31 (0)10 408 1340; fax: +31 (0)10 408 9162

Email addresses: olafwallaart@gmail.com (Olaf Wallaart), frasincar@ese.eur.nl (Flavius Frasincar), 573834fk@student.eur.nl (Finn van der Knaap)

services and products [5]. It also allows for a more targeted market segmentation and service development [6]. For small companies, it might be possible to obtain their customers' opinions manually. However, analyzing huge amounts of review data becomes labor-intensive and time-expensive for large companies. Sentiment analysis is a Natural Language Processing (NLP) sub-task that aims to extract the sentiment and opinions in a given piece of text and combines this information into useful results for companies, researchers, and users [7].

In practice, however, one might not only be interested in the overall sentiment but also in someone's sentiment with respect to various aspects. Aspect-Based Sentiment Analysis (ABSA) [8] takes into account this more fine-grained approach by identifying people's sentiments towards specific aspects. Despite being more difficult, ABSA yields more in-depth results. ABSA consists of three sub-tasks: (1) Opinion Target Extraction (OTE) aims to identify aspect terms within a given piece of text, (2) Aspect Category Detection (ACD) helps in identifying the topics or categories debated in opinionated texts and can be used to connect found aspect terms to predefined aspect categories, and (3) Sentiment Classification (SC) detects the sentiment towards the found aspects. This research focuses on simultaneously performing two of the three tasks, namely ACD and SC, which is also referred to as Aspect-Sentiment Detection (ASD). ACD and SC are usually done sequentially. However, both classification tasks can benefit from each other if they are trained simultaneously [9].

Supervised ABSA algorithms have achieved high performance on evaluation tasks [10, 11, 12]. However, obtaining such labeled training sentences is difficult and expensive, especially for new domains and/or languages [13]. To tackle this issue, weakly-supervised neural models only require minimal input and have shown great potential, especially methods that only require a domain-specific set of predefined keywords for each aspect/sentiment [9, 14]. Yet, these methods also have their shortcomings, for example, Bidirectional Encoder Representations from Transformers (BERT) [15] is used by Kumar et al. [14] to create semantically coherent class vocabularies which are later used to label sentences. However, BERT does not leverage the full semantic information of a sentence. To address the above limitation, we propose to leverage Sentence-BERT (SBERT) [16] to directly create semantically-relevant sentence embeddings. Given a corpus of documents containing sentiment expressions, we develop a classifier for the aspect-sentiment pair that is present in a sentence (we assume that only one aspect is present in a sentence), such that we can output its aspect category and corresponding sentiment label. To achieve this, we perform ASD without using labeled datasets. The only input required by the user is a small set of domain-specific seed sentences (or seed words) for each aspect and/or sentiment. We formulate the following research question:

> **Can we improve weakly-supervised ASD using domain-adapted context-aware sentence embeddings?**

Thus, we leverage the power of domain-adapted context-aware SBERT embeddings [16] and combine this with a semi-supervised joint neural network structure based on BERT to predict aspect and sentiment

simultaneously. Creating an algorithm that classifies aspect and sentiment simultaneously can lead to better results and a more accurate classification [9]. Our proposed solutions are made publicly available at `https://github.com/ofwallaart/SBASC`.

The contributions of this work can be summarized as follows:

- We propose an algorithm to perform weakly-supervised ASD. First, we create a labeled training dataset from a large unlabeled dataset by assigning aspect and sentiment labels using domain-adapted context-aware SBERT embeddings [16]. To this end, we compute the cosine similarity between the seed sentences and unlabeled sentences to assign the most probable aspect and sentiment labels to each sentence. In the second step, a BERT-based neural network is trained to further improve aspect and sentiment classification by learning latent features from the generated labeled dataset.
- We experiment with a loss function called focal loss [17], which is able to handle class imbalance, an often occurring feature in ABSA datasets, without considering the class distribution. A comparison is made with a model that uses a noise-robust loss function called Generalized Cross Entropy (GCE) [18] to see if the performance of existing methods can be further improved by altering the loss function.
- We investigate if using a domain-specific post-training procedure for transformer-based language models where existing models are enriched with domain knowledge further improves performance.
- We analyze the performance of our model on restaurant and laptop reviews. We show that using seed sentences instead of words gives a boost in performance, providing state-of-the-art results for the defined ASD task. Using our findings, organizations are able to extract relevant aspect categories and the corresponding sentiment from unstructured data to enhance customer satisfaction.

The rest of the paper is structured as follows. Section 2 discusses previous work related to ASD. Next, Section 3 gives an overview of the used data. Then, in Section 4, we present the sentence labeler and neural network algorithm followed by, in Section 5, an evaluation of the results. Last, Section 6 presents concluding remarks and topics for future research.

2. Related Work

The goal of ABSA is to find the sentiment of a group of people towards a certain topic. Joint ASD proposes to extract the aspect categories and determine the corresponding sentiment simultaneously. The main advantage is that combining these two tasks allows one to use sentiment information to find aspect categories and vice versa [8]. In this review, we mainly focus on previous work concerning the joint task of unsupervised or weakly-supervised ASD.

Most early methods are syntax-based approaches [19, 20]. They revolve around creating and utilizing aspect and sentiment lexicons. Existing lexicons can be leveraged, or algorithms can be developed to build lexicons automatically. Hu and Liu [19] use a frequency-based method to identify frequent nouns to build

an aspect lexicon, whereas Qiu et al. [21] build aspect and sentiment lexicons from some seed words and syntactic rules. Zhao et al. [22] aim to generalize some of these syntactic structures. The biggest shortcoming is that this method requires users to specify syntactic rules. Furthermore, such methods rely heavily on the accuracy of text parsing methods (e.g., Part-of-Speech (POS) taggers) which are not error-free [8].

Early unsupervised or weakly-supervised models are mainly based on Latent Dirichlet Allocation (LDA) [23]. Although LDA was originally developed for topic modeling, the application can be adjusted to also suit the needs of sentiment analysis by biasing the model to utilize some a priori knowledge, often in the form of sentiment lexicons. Lin et al. [24] propose the Joint Sentiment-Topic (JST) model. This weakly-supervised model modifies the original LDA model by including an additional sentiment layer and by using a domain-independent sentiment lexicon for supervision. Huang et al. [25] further extend this framework by including sentence-level structural knowledge to detect topics and sentiment simultaneously. The proposed method considers each sentence within a review as having its distinct topic mixture, meaning that the topic probability distribution for each sentence in a review is unique. In contrast, traditional models treat an entire review as a single topic mixture rather than individual sentences. As a result, the proposed method is able to outperform traditional LDA-based approaches for topic detection and sentiment detection for online reviews. Zhao et al. [26] employ a different approach called MaxEnt-LDA where the authors integrate a discriminative maximum entropy approach into LDA. In addition, Wang et al. [27] aim to solve the problem of inaccurate approximations for the posterior distribution over topics by using a two-layer structure model inspired by Restricted Boltzmann Machines (RBMs). A downside of the RBM-based method is that it relies on substantial amounts of prior knowledge such as POS tagging and sentiment lexicons.

Like LDA approaches, Zhou et al. [28] propose a topic modeling approach to model the joint distribution of topics and sentiments. However, many existing methods disregard the possibility that words and topics are conditionally dependent, which could result in certain topics having equivocal representations. To address this challenge, Zhou et al. [28] introduce a Weakly-supervised Graph-based Joint Sentiment Topic (W-GJST) model, adopting a joint sentiment topic model and an edge-gated graph convolutional network to construct a graph representation of topics and words, allowing the model to investigate the hidden dependency relationships between them. Furthermore, the authors utilize a multi-label topic classifier alongside an unsupervised self-training approach, eliminating the need for labeled data and pre-defined topic terms. This self-training method takes advantage of topic and word embeddings, as well as conditional topic distributions obtained from the W-GJST model, to train a neural network using unlabeled data. In contrast to many of the above-mentioned topic modeling approaches, we do not model the distribution of topics and sentiments in this work. Instead, we focus on creating a labeled dataset by leveraging semantically coherent class vocabularies, which are constructed using a limited amount of seed words or sentences.

Similarly, Zhuang et al. [9] propose a model where users only need to provide a small set of seed words for each aspect class and each sentiment class as well as an unlabeled corpus of reviews. It extends the autoencoder-based method by He et al. [29] to a joint model by predicting aspect and sentiment labels and

also includes a regularization method to integrate user guidance into the modeling process. Two models are proposed. The first model, Aspect Sentiment Autoencoder (ASA), has two parallel autoencoder structures for aspect and sentiment, respectively. Based on the observation that some sentiment words are specifically used for a certain aspect, the second model, Joint Aspect Sentiment Autoencoder (JASA), exploits the correlation between aspect and sentiment words in sentences by using a joint autoencoder structure. The second model outperforms the first model, confirming the hypothesis that exploiting the correlation between aspect and sentiment words in sentences is beneficial.

Huang et al. [30] propose Joint Aspect Sentiment Topic Embedding (JASen), which learns a joint topic representation for each sentiment-aspect pair. This representation is created in the same embedding space as words so that the surrounding words of topic embeddings properly describe the semantics of topics. JASen outperforms all the baselines on both restaurant and laptop SemEval datasets.

Bhattacharjee and Gangadharaiah [31] propose a topic modeling approach based on a Variational Auto-Encoder (VAE) that performs ABSA without requiring fine-grained labels for either aspects or sentiments. By feeding transformer sequence embeddings into a VAE model, the model learns a document-topic distribution and a token-topic distribution. The proposed approach is able to outperform JASen for most evaluation measures. Yet, while the method allows for the detection of multiple aspects in a document, document-level supervision (i.e., the overall document-level sentiment is used) is employed in contrast to JASen and our approach, which is not always readily available.

Kumar et al. [14] introduce the Context-aware Aspect category and Sentiment Classification (CASC) model, a BERT-based semi-supervised hybrid approach that consists of the following steps. The first step takes a small set of seed words for each aspect category and sentiment class to construct respective semantically coherent class vocabularies with the help of a BERT [15] contextual model. The second step makes use of these constructed vocabularies along with POS tags to label a subset of sentences from the training corpus. Due to the semi-automated labeling process, noise may be induced in the labels during the process. Hence, the last step builds a noise-robust deep neural network for aspect and sentiment classification. Results show that the method outperforms current models such as CAt [32], ABAE [29], and JASen (where CAt and ABAE only extract aspects). However, CASC has its limitations. A disadvantage of BERT is that no independent sentence embeddings are computed. To derive sentence embeddings from BERT, each sentence pair must be processed separately, which is computationally infeasible for large datasets. Reimers and Gurevych [16] address this problem by introducing SBERT, adding a pooling operation to BERT's output to create fixed-sized sentence embeddings. These embeddings are then fine-tuned on semantic textual similarity data using siamese network structures. The weights are updated to produce semantically meaningful embeddings that can be compared using cosine similarity, enabling quick identification of semantically similar sentences. Therefore, we build upon CASC but leverage SBERT embeddings to derive semantically meaningful sentence embeddings to create a high-quality labeled dataset for the joint task of ASD.

More recently, Large Language Models (LLMs) have emerged in the context of ABSA, with zero- and

few-shot settings indirectly representing an unsupervised or weakly-supervised setting. For example, Zhang et al. [33] aim to provide a comprehensive overview of the capabilities of LLMs in the framework of sentiment analysis. Although LLMs outperform smaller language models in few-shot learning settings, they generally struggle to generalize to more difficult tasks. Similarly, for more complex tasks like ABSA and ASD in this work, it has been shown that LLMs perform modestly [34, 35]. For those reasons, this work concentrates on a comparison between the proposed approaches and state-of-the-art approaches for weakly-supervised ASD, and leaves a direct comparison with LLMs as future work.

3. Data

We use unlabeled review data from multiple domains (restaurant and laptop) for training. For the restaurant data, we use unlabeled restaurant reviews from a public Yelp dataset[1]. For the laptop data, we leverage unlabeled Amazon reviews under the laptop category collected by McAuley et al. [36][2].

For evaluation, we use the SemEval 2016 Task 5 Subtask 1 dataset [37]. This XML-structured database contains a training and test set for multiple domain reviews (including restaurant and laptop) for which sentiment-labeled aspects are provided. An example from the restaurant domain of the SemEval 2016 test set is given in Figure 1. Each sentence is labeled with aspect targets and aspect categories, as well as the corresponding sentiment polarities.

```
<sentence id="en_BlueRibbonSushi_478218345:3">
    <text>The entire staff was extremely accomodating and tended to my every need .</text>

    <Opinions>
        <Opinion target="staff" category="SERVICE#GENERAL" polarity="positive" from="11"
            to="16"/>
    </Opinions>
</sentence>
```

Figure 1: A sentence from the SemEval 2016 test set.

We follow a similar data processing procedure as in the works of Huang et al. [30], Kumar et al. [14], and Zhuang et al. [9], and hence, entity types are regarded as aspect classes. We ignore the attributes of entities as the information is too fine-grained. For the restaurant dataset, we neglect the entity type `RESTAURANT` since such sentences do not express aspect-specific opinions. Despite not being directly obvious, we notice that Kumar et al. [14] use a different number of aspect categories (i.e., `FOOD`, `AMBIENCE`, and `SERVICE`), whereas other works use a total of five aspect categories (i.e., `AMBIENCE`, `DRINKS`, `FOOD`, `LOCATION`, and `SERVICE`). Kumar et al. [14] merge the `FOOD` and `DRINK` categories together to a single `FOOD` category. Furthermore, the authors merge `AMBIENCE` and `LOCATION` to form the `AMBIENCE` category. In this way, they follow the approach of He et al. [29] where evaluation is done only on the three major aspects found in the data. In order to make a fair comparison between all models, we compute and report the results of all models for

[1] https://www.yelp.com/dataset
[2] http://jmcauley.ucsd.edu/data/amazon/index_2014.html

both the 3-class restaurant (restaurant-3) dataset as well as the 5-class restaurant (restaurant-5) dataset. For the laptop dataset some rare entity types are removed and only the following eight entity types are kept as aspect classes: `BATTERY`, `COMPANY`, `DISPLAY`, `KEYBOARD`, `MOUSE`, `OS`, `SOFTWARE`, and `SUPPORT`. Last, in line with other research [9, 14, 30], we remove sentences with multiple labels or with neutral sentiment polarities for training and testing to simplify the problem and prevent ambiguity in results. The data distribution of the three considered datasets is as follows. In the restaurant-5 and restaurant-3 datasets, we have 17,027 training instances and 643 test instances. In the laptop dataset, we have 14,683 training instances and 306 test instances.

4. Methodology

The goal of this paper is to develop a method to perform unsupervised or weakly-supervised ASD where we aim to simultaneously extract two closely related elements from a review sentence to form a pair (aspect, sentiment). Formally, the problem can be defined in the following way. Given a sentence X_i from the unlabeled input corpus $\mathcal{X} = [X_1, X_2, \ldots X_N]$, a predefined set A of aspect categories, and a predefined set S of sentiment polarities, the ASD task aims to detect the pair (a, s) that X_i entails in natural language meaning, where a is an aspect category in A and s is a sentiment polarity in S. Note that from the above definition, we assume that only a single pair is present in each sentence X_i.

We first create a labeled training dataset from a large unlabeled dataset by assigning aspect and sentiment labels using SBERT embeddings [16], which is described in Section 4.1. In the second and final step, a neural network is trained with the aim of further improving aspect and sentiment classification by learning latent features from the previously labeled dataset. Section 4.2 presents the structure and inner workings of the neural network.

4.1. Labeled Dataset Creation

We propose two methods for creating a labeled dataset from a small set of user-provided data. Both methods leverage the power of SBERT embeddings to derive semantically meaningful sentence embeddings (i.e., semantically similar sentences are close in vector space). Original BERT [15] models show strong performance on semantic textual similarity tasks. A disadvantage of BERT is that no independent sentence embeddings are computed. To derive sentence embeddings from BERT, every sentence pair needs to be fed into the neural network. However, especially for large datasets, doing so requires an infeasible amount of computation. SBERT adds a pooling operation to the output of pre-trained BERT to derive a fixed-sized sentence embedding. SBERT then fine-tunes these pooled embeddings on semantic textual similarity data by using siamese network structures. The weights are updated in such a way that the produced sentence embeddings are semantically meaningful and can be compared with cosine similarity. This makes it possible to quickly find semantically similar sentences. Using this semantic textual similarity search, we propose a method called Sentence-Based Aspect and Sentiment Classification (SB-ASC). The method uses seed sentences

for each aspect category and sentiment polarity to create a labeled dataset. The second approach, called Word-Based Aspect and Sentiment Classification (WB-ASC), follows a similar starting point as previous literature by using seed words to create a labeled dataset but with SBERT. Both approaches are explained in the following sections.

4.1.1. Labeler Using Seed Sentences

As discussed previously, the construction of BERT makes it unsuitable to quickly perform a semantic similarity search between a large set of sentences since sentence embeddings are not directly available. To leverage the full semantic meaning of a sentence with the goal of constructing a labeled dataset, we opt to use pre-trained SBERT embeddings to derive fixed-size sentence embeddings. In our proposed methods we use the `all-mpnet-base-v2` SBERT model for English datasets. The model has a dimension size d of 768, is fine-tuned on a dataset of 1 billion sentence pairs, and provides the best performance for semantic textual similarity of all pre-trained models in the work of Reimers and Gurevych [16].

Since SBERT is able to efficiently embed entire sentences, we opt to differ from previous literature that often takes words as seeds [38, 39] for aspect and sentiment categories. Instead, we opt to provide a small set of seed sentences per aspect and sentiment category. Obtaining these sentences can be done by using a small sample of existing unlabeled datasets, but can also be fictional as long as the seed sentences entail some value for the relevant aspect or sentiment. We argue that obtaining or creating these sentences takes similar or very little extra effort compared with the seed word set creation. For brevity we only show the labeling process for the aspect categories A, however, we follow the same procedure for the sentiment set S.

We start with a set of seed sentences X^a associated with $a \in A$. For each seed sentence set X^a we compute its SBERT sentence representation as:

$$\underset{|X^a| \times d}{H^a} = SBERT(X^a). \tag{1}$$

Here $|X^a|$ is the number of seed sentences in X^a and d is the hidden dimension size of the chosen SBERT model. Similarly, we compute SBERT embeddings for each unlabeled sentence $X_i \in \mathcal{X}$ as:

$$\underset{|X_i| \times d}{H^{SBERT}} = SBERT(X_i). \tag{2}$$

We are now able to directly compare the embedded seed sentences with unlabeled sentences from the training dataset. However, since the set of seed sentences is limited to a small amount (in this work the seed set consists of five sentences for each category), they might not contain all the semantically relevant information to fully describe an aspect. For example, the seed set could contain specific semantic structures or might not be able to capture the aspect in a general sense. Therefore, we add (concatenate) the average of embeddings to the seed embeddings set:

$$\underset{(|X^a|+1)\times d}{\tilde{H}^a} = \begin{bmatrix} H^a \\ \frac{1}{|X^a|}\sum_{i=1}^{|X_a|} H_i^a \end{bmatrix}. \tag{3}$$

Preliminary research shows that adding the average embedding of seed sentences improves the performance of the final models. Next, we compare an unlabeled training sentence to any of the provided aspects. To compute the semantic similarity between sentences, we use the cosine similarity between (average) SBERT embeddings [16]. The cosine similarity between two vectors is defined as the dot product of two vectors divided by the product of their lengths:

$$\text{cosine}(x, y) = \frac{x \cdot y}{|x||y|}, \tag{4}$$

which can easily be extended to a pairwise computation for matrices. Hence, for each seed set, we compute:

$$\underset{1\times 1}{c_{ij}^a} = \text{cosine}(\tilde{h_i^a}, h_j^{SBERT^\intercal}), \tag{5}$$

so that $C^a \in \mathbb{R}^{(|X^a|+1)\times|\mathcal{X}|}$. These cosine similarity scores for each training sentence with respect to each seed sentence in X^a and their average are transformed into a single similarity score by taking the maximum value over the seed set:

$$\underset{1\times 1}{m_j^a} = \max_{1\leq i\leq |X^a|+1} c_{ij}^a, \tag{6}$$

where m^a is a vector of size $|\mathcal{X}|$ containing the highest similarity score between a training sentence and any sentence from seed set X^a and their average. The operations described in Equations 1, 3, 5, and 6 are performed for each aspect category $a \in A$. Lastly, we compute the arg max of all m^a by concatenating the maximum aspect scores vectors:

$$x_j^a = \arg\max_{a\in A} m_j^a, \tag{7}$$

where $x_j^a \in A$. For a given sentence X_j, we assign aspect category x_j as a label if its respective cosine similarity score $m_j^{x_j}$ is above a certain threshold λ_a. A similar approach is followed for each sentiment polarity $s \in S$ so that we end up with a max polarity score vector. We also require the cosine similarity score for sentiment to be above a threshold λ_s before X_j is assigned a sentiment label. The hyperparameters λ_a and λ_s play an important role in improving the quality of assigned labels since they influence the amount of noise (i.e., labels that do not have a high similarity score to any aspect and sentiment) that is included in the labeled training dataset. An example of the used seed sentences for the `FOOD` category in the restaurant-5 dataset is shown in Table 1. Each seed sentence is fictitious but contains at least one different word from the seed words in order to make the sentences representative of the aspect and sentiment categories (the seed words have been reused from CASC and JASen). In addition, we limit the number of seed sentences to five.

By implementing such rules, we address the potential bias introduced by the generation of seed sentences, as we provide the model with an equally diverse set of sentences while still representing each aspect/sentiment category similarly to using seed words.

Table 1: Example seed sentences for the aspect category `FOOD` for the restaurant-5 dataset.

Food
the food is good,
this is the best sushi buffet we ever had ,
all the dishes tasted the same .
the pizza at old chicago is actually pretty good,
hamburgers bland and buns dry and cold .

The final result is a labeled training corpus $\mathcal{X}_{\mathcal{L}} \subset \mathcal{X}$ that is used in the neural network step to further train and improve our model. Note that the original seed sentences X^a and X^s are not added to the labeled dataset $\mathcal{X}_{\mathcal{L}}$. When one uses fictional seed sentences, these sentences might be good to generally describe a certain aspect but could fail to truthfully capture semantic structures in terms of the actual source data. Since we want to create a labeled corpus that is as close to the real data as possible, we only use sentences in the labeled training corpus that are from the unlabeled corpus. However, when one uses seed sentences that are picked from the unlabeled dataset, the sentences will indirectly also be in the labeled training corpus since they have a cosine similarity score of 1 with themselves. The labeler using seed sentences as input is visualized in Figure 2.

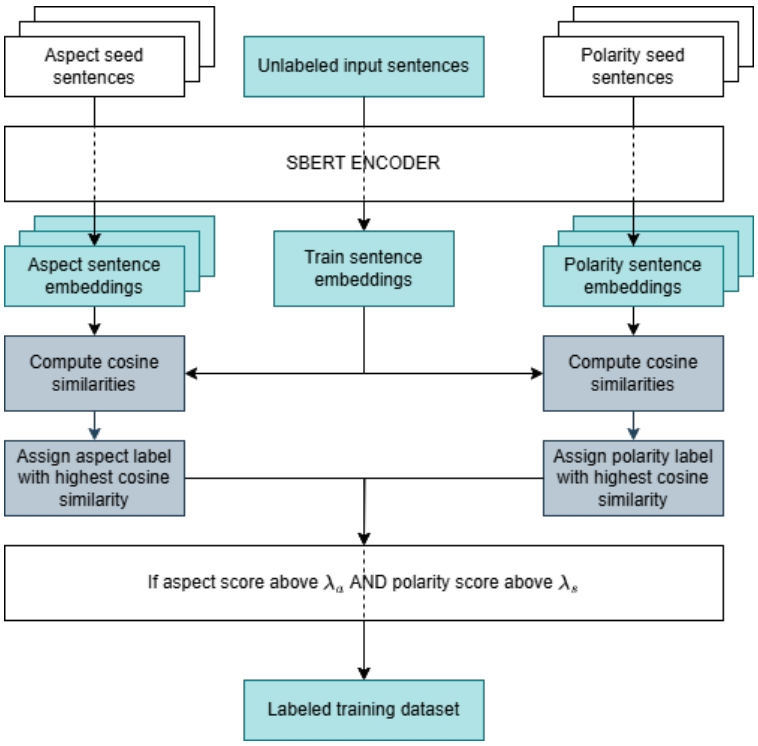

Figure 2: Visual representation of the labeler that uses seed sentences as input.

4.1.2. Labeler Using Seed Words

Next to our sentence labeler, we propose a model that only needs a set of seed words instead of seed sentences. An example of the used seed words for the restaurant-5 dataset is shown in Table 2. We have the same labeling information as in previous work of Huang et al. [30] and Kumar et al. [14]. However, for this method, we still use SBERT for its ability to match semantic similar sentences. The proposed method uses seed words to find relevant seed sentences that, in turn, can be used to match with unlabeled documents. In other words, this word-based approach is used to bootstrap the previously described sentence-based approach. Formally, we start with a set of seed words L^a associated with $a \in A$. Each set L^a contains $|L^a|$ seed words. The goal is to construct a set of sentences X^a so that we can follow the same procedure as in Section 4.1.1.

Table 2: Example seed words for the restaurant-5 dataset.

Aspect/Sent.	Keywords
Location	location, street, block, river, avenue
Drinks	drinks, beverage, wines, cocktail, sake
Food	food, spicy, sushi, pizza, taste
Ambience	ambience, atmosphere, room, seating, environment
Service	service, tips, manager, waitress, servers
Positive	good, great, nice, excellent, perfect
Negative	bad, terrible, horrible, disappointed, awful

Comparing a single word with an entire sentence is possible with SBERT since the word will be regarded as a single-word sentence. However, since a single word is lexically and semantically very different from a sentence, the performance of textual similarity tasks is likely to drop. To overcome this, we propose two approaches to finding related seed sentences.

The first approach encodes every single seed word from the seed set, computes cosine similarities with all training sentences, and appends the most similar training sentence for each seed word to the set of seed sentences. The second approach takes all seed words and uses them to create a single sentence. This sentence is then encoded using SBERT and its cosine similarity is computed with all the training sentences. We then take the top $|L^a|$ sentences with the highest cosine similarity score. However, for the seed sentences to represent all seed words, we make sure that every seed word occurs at least once in any of the selected sentences. In other words, we select the sentence from all training sentences containing seed word l^a that has the highest cosine similarity. Lastly, we concatenate the two sets of training sentences together to obtain a single set of seed sentences $X^a = X^a_l \cup X^a_s$.

From this point, we follow the same procedure as before to construct a predicted aspect and sentiment for each training sentence. We again only include those sentences in the labeled training set $\mathcal{X}_\mathcal{L}$ that have cosine scores above thresholds λ_a and λ_s for aspect and sentiment, respectively. Figure 3 gives a visualization of the labeler that uses seed words as input.

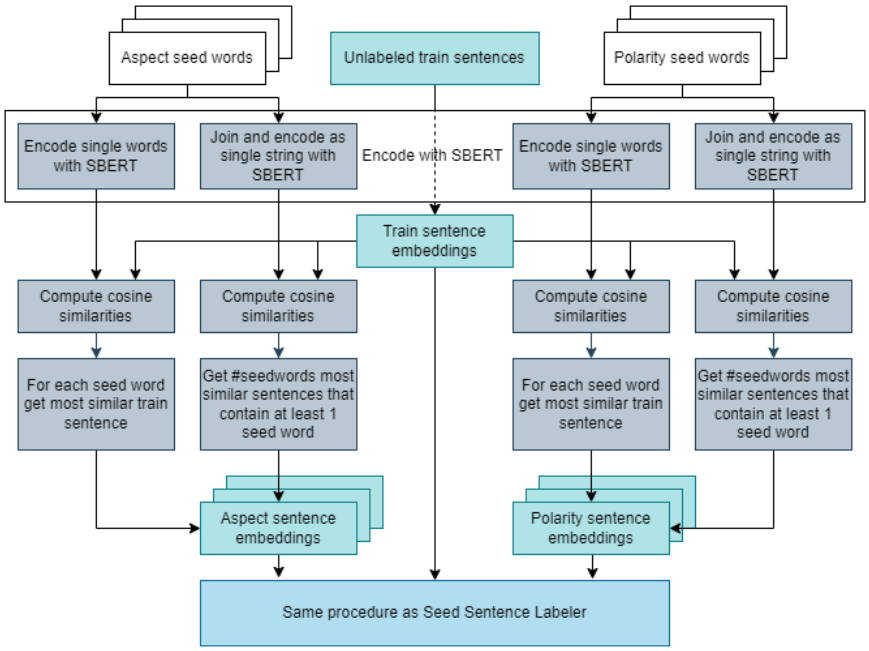

Figure 3: Visual representation of the labeler that uses seed words.

4.2. Joint BERT-Based Neural Network

In this section, we introduce a joint deep neural network that uses a BERT-based architecture for simultaneous aspect and sentiment classification.

4.2.1. Model

From the previous step, we obtain a labeled dataset $\mathcal{X}_{\mathcal{L}}$. However, we are not able to successfully annotate all sentences in $\mathcal{X}$. This is partially caused by the filtering procedure where we do not label sentences that have cosine scores below the pre-defined threshold values λ_a and λ_s. Another cause is the fact that in some sentences the aspect and sentiment are expressed implicitly. Although SBERT is able to understand semantic features, training a neural network on domain-specific data might improve performance since it is able to learn more complex features.

BERT is built as a multi-layer bidirectional transformer encoder that is largely based on an attention-based transformer model proposed by Vaswani et al. [40]. The model consists of several stacked self-attention and point-wise, fully connected layers. The architecture of a single encoder layer consists of two sub-layers. The first is a multi-head self-attention mechanism. The second layer is a position-wise fully connected feed-forward network. Furthermore, every sub-layer is surrounded with a residual connection [41] and subsequently a normalization layer [42]. In this work, for all datasets, the BERT model is architecturally equivalent to the base implementation (`bert-base-uncased`) in the work of Devlin et al. [15] where the model consists of 12 encoder blocks/layers, a hidden dimension size d of 768, and 12 self-attention heads.

Given a sentence from our labeled dataset $X_{\mathcal{L}} \in \mathcal{X}_{\mathcal{L}}$ we compute the input representation as:

$$X_{\mathcal{L}} = ([\text{CLS}], w_1, \ldots, w_{|X_{\mathcal{L}}|}, [\text{SEP}]), \tag{8}$$

where w_i is the ith word present in the sentence and $|X_{\mathcal{L}}|$ is the total number of words present in the sentence. The appended special tokens `[CLS]` and `[SEP]` are used by BERT to indicate the beginning and end of a text sequence (e.g., a sentence). This sentence is then passed through the BERT model that has been initialized with post-trained Domain Knowledge BERT (DK-BERT) [43] to obtain the hidden representation H, where we use the post-trained models of Xu et al. [43] for the restaurant[3] and laptop[4] domains:

$$\underset{(|X_{\mathcal{L}}|+2)\times d}{H^{BERT}} = BERT(X_{\mathcal{L}}). \tag{9}$$

H in Equation 9 is the second-to-last hidden layer. We choose this 11th encoder layer since Kumar et al. [14] find this layer to have the best latent representation of all tokens in a sentence. The explanation follows from the use of post-trained DK-BERT. When encoding the sentences, it is assumed that the output of the last hidden layer is close to its target function (i.e., Masked Language Model and Next Sentence Prediction) and hence the second-to-last layer encodes embeddings of all the tokens in the sentence.

Next, following Kumar et al. [14], we transform the embeddings of individual tokens into a single vector to obtain a global contextual sentence representation by applying mean pooling. We ignore the `[CLS]` and `[SEP]` tokens when computing the sentence representation:

$$\underset{1\times d}{\bar{h}} = \frac{1}{|X_{\mathcal{L}}|}\sum_{i=2}^{|X_{\mathcal{L}}|} h_i^{BERT}, \tag{10}$$

where h_i^{BERT} is the word embedding of the ith word. The final step is to pass the hidden sentence representation $\bar{h}$ to two separate linear layers followed by a softmax layer to compute the aspect and sentiment prediction vectors p^a and p^s, respectively:

$$\underset{|A|\times 1}{p^a} = \text{softmax}(\underset{|A|\times d}{W_a} \times \underset{d\times 1}{\bar{h}^\intercal} + \underset{|A|\times 1}{b_a}), \tag{11}$$

$$\underset{|S|\times 1}{p^s} = \text{softmax}(\underset{|S|\times d}{W_s} \times \underset{d\times 1}{\bar{h}^\intercal} + \underset{|S|\times 1}{b_s}), \tag{12}$$

where p^a and p^s are conditional probability distributions, W_a and W_s are weight matrices, and b_a and b_s are bias terms. A visual representation of the neural network structure is shown in Figure 4.

[3] `https://huggingface.co/activebus/BERT-DK_rest`
[4] `https://huggingface.co/activebus/BERT-DK_laptop`

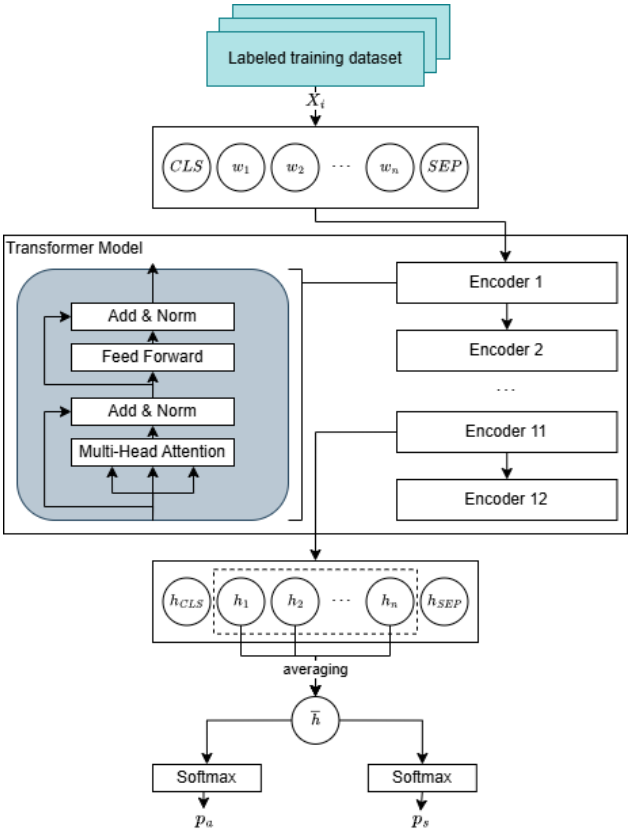

Figure 4: Visual representation of the neural network structure.

4.2.2. Model Training

Due to the semi-supervised labeling step, we train our neural network on a labeled dataset and hence consider it as a supervised machine learning problem. The overall loss of our model $\mathcal{L}$ is calculated as the sum of the aspect classification loss $\mathcal{L}_a$ and SC loss $\mathcal{L}_s$:

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_s. \tag{13}$$

In the remainder of this section, we only formulate the loss function for aspect classification. The loss function for SC is similar.

A commonly used loss function in neural network training is the Categorical Cross-Entropy (CCE) loss function. It is defined as:

$$\underset{1\times 1}{\mathcal{L}_a} = -\sum_j \underset{|A|\times 1}{y_j^a} \times \log(\underset{|A|\times 1}{p_j^a}), \tag{14}$$

where y_j^a is a vector containing the true aspect category for the jth training observation and p_j^a is a vector containing the predicted aspect category for the jth training observation.

An advantage of CCE is that it puts more emphasis on difficult samples during training and is, therefore, able to quickly converge. However, this emphasis on difficult samples becomes problematic when noise is

present in the dataset since CCE loss tends to overfit the noise present in the data [14]. Some loss functions, such as Mean Absolute Error (MAE), are robust to noisy labels. However, training with MAE is challenging due to slow convergence caused by gradient saturation [18]. Zhang and Sabuncu [18] propose the GCE loss that is more robust to noise, called $\mathcal{L}_q$ loss. GCE is a generalized mixture of MAE and CCE. The $\mathcal{L}_q$ loss function puts less emphasis on difficult samples compared to CCE. Compared to MAE, the weighting in $\mathcal{L}_q$ loss can facilitate learning by giving more attention to challenging data points. GCE is defined as:

$$\mathcal{L}_a = \sum_j \frac{1 - \left(\hat{a}_{y_j}\right)^{q_a}}{q_a}, \tag{15}$$

where $\hat{a}_{y_j} = y_j^{a\intercal} p_j^a$ is the predicted probability against the true aspect label and $q_a \in (0, 1)$ is a hyperparameter.

However, the GCE loss does not address class imbalance. In order to apply our method to multiple datasets and real-world scenarios, it is important to address class imbalance. Furthermore, since the datasets evaluated in this work are unbalanced, dealing with this feature might be more beneficial than dealing with noise. Lin et al. [17] define another alteration on the CCE function called focal loss that addresses class imbalance during training. Focal loss applies a modulating term to the cross-entropy loss, naturally handling class imbalance without having to consider the class distribution. Focal loss is defined as:

$$\mathcal{L}_a = -\sum_j (1 - \hat{a}_{y_j})^{\gamma_a} \log(\hat{a}_{y_j}). \tag{16}$$

Setting the focusing parameter $\gamma_a > 0$ reduces the relative loss for well-classified examples, putting more focus on wrongly classified examples, i.e., the scaling factor $(1 - \hat{a}_{y_j})^{\gamma_a}$ decays to zero as confidence in the correct class increases and hence, this scaling factor can automatically down-weight the contribution of easy examples during training and put more focus on difficult examples.

Figure 5 shows a comparison between the GCE loss and focal loss. From Figure 5 it is clear that GCE is indeed a generalization of CCE and MAE, taking only values in between these two functions (for $q \in (0, 1)$). Focal loss shows a more extreme curve (especially when γ is large), giving higher loss values to hard, misclassified examples compared to GCE and lower loss values to well-classified examples. Both CCE and GCE losses are characterized by the fact that instances classified with high confidence still obtain a considerable loss. These easily classified samples will (especially for imbalanced data) dominate the loss and control the gradient, overpowering the smaller and more difficult classes. Depending on the focusing parameter, focal loss considerably down-weights the loss for these well-classified instances. This makes it more suitable for addressing imbalances between categories.

For loss minimization, we use backward propagation. We initialize the weight matrices using a uniform distribution $U(-0.1, 0.1)$ and initialize all bias terms to zero. The AdamW optimizer [44] is used to update the weights and biases. Due to the two-step procedure, one cannot perform a traditional hyperparameter tuning procedure when evaluating tuning performance on the loss value. This is caused by the fact that the loss value of the neural network during training is dependent on the choice of λ_a and λ_s. Furthermore, one

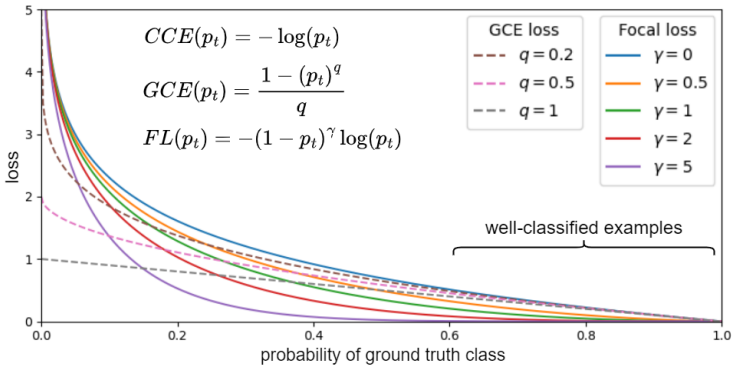

Figure 5: Comparison of Categorial Cross Entropy (CCE) loss, Generalized Cross Entropy (GCE) loss, and Focal Loss (FL). Note that for $\gamma = 0$ and $q \to 0$ both GCE and FL become CCE and for $q = 1$ GCE becomes Mean Absolute Error.

should only evaluate on the created labeled dataset and not on a manually annotated evaluation set since this would devalue the unsupervised aspect of the algorithm's performance. We therefore choose to perform a two-step procedure for hyperparameter tuning. We first determine the optimal value of λ_a and λ_s by fixing the parameters in the neural network to their default values and choosing the highest F1 score (similar to the procedure of Kumar et al. [14]). Next, we tune the hyperparameters in the neural network by using the optimal λ values obtained in the first step. For the neural network, the parameters that are tuned include the learning rate, the parameters of the AdamW algorithm (β_1 and β_2), the batch size, and the γ value in the focal loss functions for aspect and sentiment, γ_a and γ_b, respectively. 80% of the labeled training data is used for tuning the neural network hyperparameters and the other 20% is used for validation, as empirical studies have shown that such a split often works optimally [45]. After hyperparameter optimization, the model is trained using the best settings on the entire training dataset. We use a Tree-structured Parzen estimators (TPE) algorithm [46] for both steps of hyperparameter selection. For the first hyperparameter tuning step (i.e., tuning λ_a and λ_s) the number of evaluation trials is set to 30. For the second step, more hyperparameters need to be tuned so the number of evaluation trials is set to 75.

5. Evaluation

In this section, we compare and evaluate the proposed models. Before discussing the performances of our proposed models, we first present the performance measures that are used to evaluate the results. Then, the baseline and ablation models for comparison are discussed. Next, we discuss the hyperparameter tuning process. Thereafter, we show the performance results for all considered models.

5.1. Experimental Framework

5.1.1. Performance Measures

Evaluation is performed on a similar test dataset to that used by Zhuang et al. [9] to gauge the performance of the proposed models against baseline methods. We evaluate the performance of the models for aspect

classification and sentiment classification separately. The evaluation measures used are accuracy, macro-precision (macro-P), macro-recall (macro-R), and macro-F1 scores. The macro scores are defined as the mean of the individual label-wise scores. Since the $F1$ score combines the precision and recall measures and is designed to work well on imbalanced data, we use it as our main measure for comparing model performance.

5.1.2. Baseline Models

We compare our models to the following baseline models:

- **CosSim**: For each aspect and sentiment we compute the average embedding of seed words using a word2vec model trained on the training corpus. By computing maximum cosine similarities between the seed embeddings and each test sentence (by averaging all word embeddings in a test sentence) we classify aspect and sentiment.
- **CosSim-Sentence**: Similar to CosSim, however, we compute the average embedding of a set of seed sentences instead of seed words by averaging all individual word embeddings present in the seed sentence. These average seed sentence embeddings are then compared to each test sentence for classification (by computing maximum cosine similarities).
- **BERT** [15]: We use the pre-trained BERT language model (12-layer, 768-hidden, 12-attention, uncased) with a classification layer on top. The model is fine-tuned by weakly labeling the unsupervised corpus to generate a training set in the following way: If a sentence contains a seed word from an aspect or sentiment seed set, we label it correspondingly. Two separate models are trained for aspect and sentiment.
- **JASen** [30]: A weakly-supervised model that jointly learns aspect and sentiment representations in a single embedding space using an adaptation of the Skip-Gram model [47]. Next, a convolutional neural network is used as the classifier by training it on pseudo-labels given by the cosine similarity between document embeddings and topic embeddings from the created embedding space.
- **CASC** [14]: A weakly-supervised model using post-trained DK-BERT and a small set of seed words for labeled data preparation. Next, a neural network using labeled data is used for ASD.

The method proposed by Zhuang et al. [9] called JASA shows similar performance to CASC and is also interesting to include as a comparison. However, the code from JASA is not readily available. Hence, we cannot guarantee to exactly reproduce results. Moreover, a one-on-one comparison with our results is also difficult since Zhuang et al. [9] use different training data (the test set is similar, but might also be slightly different due to preprocessing). Results are only available for the restaurant-5 and laptop datasets, and we, therefore, only perform a weak comparison between our results and the results reported by Zhuang et al. [9].

5.1.3. Ablation Models

In order to analyze the performance of individual parts of the models and to understand the effectiveness of different key components, we perform an ablation study. By removing important model components one after another, one can observe and reason how certain elements of the model perform and interact. The following ablation models for both SB-ASC and WB-ASC are proposed:

- **(SB/WB)-ASC:** Our full Sentence/Word-Based Aspect and Sentiment Classification model as proposed in Section 4 using the seed sentence labeler from Section 4.1.1 or seed word labeler from Section 4.1.2.
- **(SB/WB)-ASC w/o DK:** We omit the use of a post-trained BERT language model on domain knowledge as proposed by Xu et al. [43] in the neural network step. Instead, we use the pre-trained base model of BERT (12-layer, 768-hidden, 12-attention, uncased).
- **(SB/WB)-ASC w/o DL:** We omit the neural network step altogether and only use the SBERT sentence labeler as proposed in Section 4.1. The aspect classification of a test sentence is done by finding the largest cosine similarity between the test sentence vector and the aspect vocabulary vectors. The same process is followed for SC. In other words, we omit the threshold values and label every sentence regardless of their cosine similarity score.
- **(SB/WB)-ASC w/o SBERT:** Instead of using pre-trained SBERT to create sentence embeddings in the sentence labeler step, we use the unweighted average of post-trained BERT word embeddings.
- **(SB/WB)-ASC w/o FL:** We replace the focal loss function with a GCE loss function so that the neural network part of our model is similar to the model proposed by Kumar et al. [14].

5.2. Hyperparameter Tuning

In this section, we briefly discuss the results of the hyperparameter tuning process of the SB-ASC and WB-ASC models. Table 3 shows the optimal hyperparameters. From this, we observe that a pattern occurs for the thresholds λ_a and λ_s, which determine if an aspect category or sentiment category is assigned to a sentence, respectively. The hyperparameter tuning process assigns a higher value to the threshold for the aspect categories compared to the threshold for the sentiment categories for all domains. As a result, the model needs stronger evidence to assign an aspect category. This might be because aspects are more specific and their incorrect assignment could confuse downstream tasks (i.e., SC).

5.3. Performance Results

The performance of all considered models for all datasets is shown in Tables 4 and 5. They report results for ACD and SC, respectively. SB-ASC outperforms the baseline methods on most measures. For many tasks, SB-ASC significantly outperforms multiple baseline methods when evaluating F1 scores. When SB-ASC is not significantly different compared to a baseline, the compared baseline is never able to significantly outperform

Table 3: Optimal hyperparameters for the proposed models.

	SB-ASC			WB-ASC		
	Restaurant-5	Restaurant-3	Laptop	Restaurant-5	Restaurant-3	Laptop
λ_a	0.65	0.7	0.7	0.55	0.6	0.5
λ_s	0.55	0.5	0.45	0.5	0.4	0.3
Learning rate	1e-6	1e-6	1e-6	1e-6	1e-5	1e-6
β_1	0.99	0.9	0.99	0.95	0.97	0.99
β_2	0.999	0.95	0.92	0.97	0.999	0.97
Batch size	36	12	12	24	12	18
γ_a	2	4	1	4	4	4
γ_s	4	4	4	4	4	4

SB-ASC. For the ACD, SB-ASC outperforms by a large margin, whereas for SC performance is generally not significantly different from the top-performing baselines.

Table 4: Comparison of methods using accuracy (A), macro-precision (P), macro-recall (R), and macro-F1 score (F1) on ACD. Results are the average over 5 individual training runs. The largest values are in bold. For CosSim, CosSim-Sentence, BERT, JASen, CASC, SB-ASC w/o FL and WB-ASC significance levels are given (*, **, *** for 10%, 5%, and 1%, respectively) for the paired, one-sided t-test assessing the null hypothesis of the mean macro-F1 score of the method being smaller than or equal to the mean macro-F1 scores of SB-ASC.

	Restaurant-5				Restaurant-3				Laptop			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
CosSim	64.29	50.61	51.15	46.20***	80.13	74.38	73.62	73.93***	63.13	65.33	62.99	61.73***
CosSim-Sentence	41.80	36.79	36.67	30.62***	59.38	57.10	61.96	55.22***	48.14	51.53	49.43	47.60***
BERT	71.48	56.72	69.83	58.25***	68.84	66.01	71.62	66.78***	63.39	62.43	60.73	59.78***
JASen	84.14	65.80	71.66	66.90	87.18	84.30	85.36	84.75***	69.38	69.77	69.83	67.51***
CASC	40.06	54.46	59.40	38.02***	81.73	77.76	84.34	77.44***	68.08	70.12	70.47	67.87***
SB-ASC	**88.34**	**73.71**	72.83	**71.35**	**88.65**	**89.26**	85.17	**86.99**	**76.29**	**74.73**	**74.54**	**73.86**
SB-ASC w/o DL	81.80	60.10	**77.40**	63.28	88.58	84.86	**85.33**	84.91	72.64	71.28	72.68	71.48
SB-ASC w/o DK	87.71	63.30	66.29	64.21	84.01	83.31	79.41	80.95	71.07	72.98	67.60	66.92
SB-ASC w/o SBERT	50.70	43.92	52.90	36.74	67.57	67.04	76.09	64.99	41.30	51.96	41.57	37.54
SB-ASC w/o FL	88.57	63.50	67.43	65.16	86.87	87.00	84.97	85.68*	73.94	65.73	68.71	65.03***
WB-ASC	83.27	60.81	70.26	63.89*	83.20	78.91	87.82	81.34***	57.65	60.94	57.32	57.33***
WB-ASC w/o DL	76.83	55.22	69.42	57.44	78.96	74.60	83.69	76.70	61.89	63.44	62.89	62.27
WB-ASC w/o DK	82.58	58.17	63.28	60.19	82.87	77.46	85.65	80.13	51.14	57.38	48.29	47.83
WB-ASC w/o SBERT	47.62	53.67	56.85	44.74	85.61	80.26	86.61	82.45	48.99	50.61	50.92	47.52

Table 5: Comparison of methods using accuracy (A), macro-precision (P), macro-recall (R), and macro-F1 score (F1) on SC. Results are the average over 5 individual training runs. The largest values are in bold. For CosSim, CosSim-Sentence, BERT, JASen, CASC, SB-ASC w/o FL and WB-ASC significance levels are given (*, **, *** for 10%, 5%, and 1%, respectively) for the paired, one-sided t-test assessing the null hypothesis of the mean macro-F1 score of the method being smaller than or equal to the mean macro-F1 scores of SB-ASC.

	Restaurant-5				Restaurant-3				Laptop			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
CosSim	76.39	76.41	78.18	76.00***	76.02	76.28	77.95	75.70***	70.36	70.47	70.42	70.35***
CosSim-Sentence	64.88	66.74	67.60	64.74***	65.19	66.82	67.67	65.03***	62.28	62.39	62.11	61.99***
BERT	68.19	69.55	59.85	58.35***	66.07	73.86	55.46	49.64***	52.05	73.11	53.11	48.32***
JASen	80.47	81.30	76.31	78.40***	80.29	81.77	75.94	77.23***	74.27	74.39	74.26	74.22***
CASC	89.86	88.98	**90.41**	**89.40**	88.74	87.79	88.62	88.15	**76.24**	**76.86**	**77.08**	**76.94**
SB-ASC	**89.98**	89.50	89.04	89.24	**90.80**	**91.13**	89.21	**89.98**	75.90	76.79	75.68	75.58
SB-ASC w/o DL	86.63	85.74	85.62	85.68	86.95	86.19	85.94	86.06	75.24	75.27	75.19	75.20
SB-ASC w/o DK	80.78	80.62	77.52	78.34	81.83	83.14	77.92	79.26	70.36	70.52	70.23	70.20
SB-ASC w/o SBERT	86.03	85.21	87.34	85.62	88.48	87.90	**90.19**	88.21	63.32	76.85	64.11	58.76
SB-ASC w/o FL	89.93	**90.84**	87.71	88.83	89.97	90.68	88.06	88.98	71.99	75.35	71.56	70.76***
WB-ASC	66.25	74.78	55.23	49.58***	72.76	74.42	66.16	66.55***	62.21	69.26	61.50	57.63***
WB-ASC w/o DL	76.98	75.68	74.14	74.70	76.18	74.86	73.45	73.45	65.47	71.51	64.85	62.30
WB-ASC w/o DK	61.12	45.76	49.35	40.42	66.72	65.48	58.92	57.71	59.61	64.52	58.90	54.92
WB-ASC w/o SBERT	63.58	67.03	51.51	42.36	63.95	61.75	53.94	49.12	56.09	56.04	55.96	55.88

The cause for the above results might be twofold. First, the results indicate that SB-ASC obtains substantial benefits from using the semantically meaningful sentence embeddings from SBERT and is able to use this information to correctly detect the aspect category present in a sentence. Second, using the focal loss function to handle class imbalance is more important for ACD in our datasets than dealing with noise. Comparing F1 scores with the best baseline methods on ACD, SB-ASC outperforms by a margin of 4.45 pp (percentage points), 2.24 pp, and 5.99 pp on the restaurant-5, restaurant-3, and laptop datasets, respectively. Note that for the restaurant-5 and restaurant-3 datasets, JASen is the best baseline method. For the laptop dataset, CASC is the best baseline method. For all datasets, SB-ASC significantly outperforms CASC and for the laptop and restaurant-3 datasets SB-ASC significantly outperforms JASen. This indicates a more stable performance over different datasets for SB-ASC, indicating that it is more robust to different datasets and out-of-domain applications compared to CASC and JASen.

When comparing SB-ASC and CASC on SC, SB-ASC does not always outperform CASC. However, the difference in performance is small and never significantly different. Hence, we conclude that the performance on SC is similar.

Weakly comparing our results to JASA, we conclude that for the restaurant dataset (only restaurant-5 is reported in JASA) SB-ASC outperforms JASA on both tasks. For ACD, SB-ASC outperforms JASA by 3.65 pp on the F1 score (the reported JASA F1 score is 67.70). For SC, SB-ASC outperforms by 8.12 pp (81.12 F1 score). The results for the laptop dataset are less conclusive. JASA outperforms SB-ASC on ACD by 4.11 pp (77.97 F1 score). However, SB-ASC is able to perform better, albeit marginally, for SC by 1.28 pp (reported F1 score is 74.30).

The results of our proposed WB-ASC model are modest, as SB-ASC always outperforms WB-ASC. We conclude that the proposed labeling algorithm that finds seed sentences based on a list of seed words is not able to provide high-quality labels. Considering the performance of SB-ASC, the quality of the seed sentence set created by WB-ASC is inferior to that of a manually created set, indicating the importance of finding representative seed sentences.

We conclude that SB-ASC has the best overall performance on jointly detecting aspect and sentiment. CASC often has the best baseline SC performance whereas JASen has the best performance on ACD. When comparing JASen to SB-ASC, the gain in ACD is small (4.45 pp) but for SC, it is large (10.84 pp) for the restaurant-5 dataset. With CASC it is the other way around, with a large gain of 33.33 pp for ACD, but -0.16 pp for SC for the restaurant-5 dataset. The ability of SB-ASC to achieve better results for the joint task shows its true improvement in performance. The enhanced performance of SB-ASC might be attributed to the following causes. SB-ASC uses SBERT embeddings which provide high contextual embeddings for an entire sentence to create a labeled dataset. The algorithm is thus able to create higher-quality labels for the unlabeled dataset compared to CASC and JASen, which use BERT word embeddings and thus lose the meaning of full sentences. Next to creating a high-quality, weakly labeled dataset, SB-ASC builds a BERT-based joint neural network using an imbalance-robust loss function to classify aspect and sentiment

simultaneously, further improving performance as class imbalance is present in the datasets and CASC and JASen both do not account for class imbalance. Furthermore, we conclude that our models can be applied to different languages/domains. All that is needed is a sufficiently large dataset of domain reviews, a pre-trained SBERT model for a specific language, a pre-trained (or domain knowledge post-trained) BERT model, and a set of seed words/sentences.

5.3.1. Ablation Models

Comparing the proposed models with the deep learning ablation models, we conclude that even without the deep learning step SB-ASC's performance for ACD regularly beats baseline models, whereas for SC performance is modest. The deep learning step boosts performance by learning latent features in the data for all SB-ASC models, while also being able to handle noise added in the data due to our labeling procedure. WB-ASC ablation models perform worse than the full model except for the WB-ASC w/o DL model. It is interesting to note that when the F1 score for WB-ASC w/o DL is high, the score for the full model usually improves (e.g., aspect classification for the restaurant-3 dataset). This suggests that there is a certain limit to the amount of noise that can be introduced in the labeling step. When the labeling process adds too much noise and the performance of WB-ASC w/o DL is relatively low, the deep learning algorithm is not able to learn latent features.

As suspected, the results show that using DK-BERT enhances the performance of SB-ASC and WB-ASC on both tasks. By not using post-trained DK-BERT models, performance drops on average by 6.14 pp for ACD and 6.86 pp for SC across all datasets and models. This does, however, come at a cost. Training DK-BERT requires a large amount of extra data for each domain, and this data has to be available. When this large domain-specific data is not available, one is bound to use BERT models that are trained on a general corpus.

Comparing SB-ASC with SB-ASC w/o SBERT, the results show the power of using SBERT in our models. Next to being able to properly capture sentiment, it excels in detecting topics discussed in sentences due to its ability to understand an entire sentence. The same holds when comparing WB-ASC with WB-ASC w/o SBERT.

Last, we compare results for SB-ASC to those of SB-ASC without focal loss (using GCE). For ACD, using focal loss improves the F1 measure for all datasets. Results for SC are less conclusive. We only observe a large performance difference of 4.82 pp for the laptop dataset. For restaurant-3 and restaurant-5, the performance increase is small (less than 1 pp). Since class imbalance is more present for aspect categories in our datasets, the results confirm the ability of focal loss to better handle class imbalances compared to GCE loss. For SC, classes are more balanced, and hence, focal loss is less beneficial compared to GCE.

6. Conclusion

In this section, we discuss the contributions of our research followed by the limitations and potential future work.

6.1. Practical and Theoretical Contributions

This work focuses on weakly-supervised ABSA for different domains. The first proposed two-step model (SB-ASC) takes seed sentences as input for a labeling algorithm. By leveraging the power of pre-trained SBERT embeddings, the method is able to understand the contextual meaning of sentences to create a high-quality labeled dataset. This dataset is used by a class imbalance-robust BERT-based neural network that jointly learns latent features of aspect categories and corresponding sentiment. The second proposed method (WB-ASC) uses the same neural network structure but takes seed words instead of seed sentences as input for the labeling algorithm. Despite the possibility of SBERT matching single words to sentences and efforts to find representative sentences, this algorithm is generally not able to improve on existing methods and never outperforms SB-ASC.

Using SB-ASC, we are able to extract relevant aspect categories and their sentiment from unstructured data by providing a small set of seed sentences. Using seed sentences instead of words gives a boost in performance, providing state-of-the-art results for the defined ASD task. Furthermore, using domain-specific contextual language models (post-trained DK-BERT) on the weakly-supervised ASD task boosts performance.

The findings of this work can also be viewed from a supervised learning perspective. Where traditional supervised methods require large amounts of data to learn latent features, our methods need far less annotated data. By providing a very small amount of training data that captures the meaning of a certain aspect category or sentiment, SB-ASC is able to achieve state-of-the-art performance.

Using the proposed models allows retailers to improve customer satisfaction. Companies are able to accurately analyze large amounts of unstructured data in a timely manner, capturing the most important opinions. This helps companies with improving their products or services. In addition, our models can also aid customers, as reviews can help customers make better decisions.

6.2. Limitations

Our work has some limitations. First, our proposed models are only able to predict one aspect label per sentence. That is, reviews with multiple aspect targets are removed from the dataset. Second, our attempts to construct or find seed sentences from a list of seed words have not been effective. We argue that constructing a sentence is not much more time-consuming than constructing seed words. Nevertheless, it is still interesting to further explore the possibility of finding highly resembling sentences for aspect and sentiment categories from a set of seed words since it will further move the field of NLP from word-based approaches towards more contextual, sentence-based approaches.

6.3. Future work

To address the limitations of our approach, future work could adjust the models to predict multiple labels per sentence. For example, one can set a threshold value for assigning an aspect to a sentence based on the probability score for each aspect. Although this approach makes it possible to classify multiple aspects in a single sentence, it is not possible to assign a sentiment with respect to each detected aspect. To solve this, the second suggestion is to extend our ASD method with OTE to identify aspect terms in a sentence and use this information to predict the aspect category and sentiment of this aspect term. To better exploit the interdependent information between the two tasks, the double task variant of the Left-Center-Right separated neural network with Rotatory attention using deep contextual word embeddings and hierarchical attention [48] could be useful. Last, one can also explore performing the ASD task in a sequential manner where first aspect categories are detected, and then sentiment polarities are predicted for each detected aspect.

Second, future research could focus on enhancing our proposed method of finding relevant aspect and sentiment sentences from seed words. A suggestion is to create a joint embedding algorithm where both words and sentences are encoded in the same vector space. Building on this, future research could aim to perform a sensitivity analysis on the thresholds for assigning aspect and sentiment categories. Furthermore, one could create a domain-specific post-trained SBERT model to better capture sentence representations for given domains, however, for this, a large set of domain-specific semantic textual similarity data is needed, which might not be easy to obtain. Last, as we build upon methods using BERT word embeddings, we opt for SBERT embeddings. Instead, future work could analyze the effect of sentence embeddings based on other transformer models due to their promising performance in the field of sentiment analysis, such as Robustly Optimized BERT [49, 50].

References

[1] D. Reinsel, J. Gantz, J. Rydning, The Digitization of the World from Edge to Core, 2018. `https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf`.

[2] M. Salehan, D. J. Kim, Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics, Decision Support Systems 81 (2016) 30–40.

[3] Y. Liu, C. Jiang, H. Zhao, Assessing product competitive advantages from the perspective of customers by mining user-generated content on social media, Decision Support Systems 123 (2019) 113079.

[4] C. K. H. Lee, How guest-host interactions affect consumer experiences in the sharing economy: New evidence from a configurational analysis based on consumer reviews, Decision Support Systems 152 (2022) 113634.

[5] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (2008) 1–135.

[6] E. V. Kleef, H. C. V. Trijp, P. Luning, Consumer research in the early stages of new product development: A critical review of methods and techniques, Food Quality and Preference 16 (2005) 181–201.

[7] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, second ed., Cambridge University Press, 2020.

[8] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Transactions on Knowledge and Data Engineering 28 (2016) 813–830.

[9] H. Zhuang, F. Guo, C. Zhang, L. Liu, J. Han, Joint aspect-sentiment analysis with minimal user guidance, in: 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), ACM, 2020, pp. 1241–1250.

[10] W. Jin, B. Zhao, L. Zhang, C. Liu, H. Yu, Back to common sense: Oxford dictionary descriptive knowledge augmentation for aspect-based sentiment analysis, Information Processing & Management 60 (2023) 103260.

[11] P. Li, P. Li, X. Xiao, Aspect-pair supervised contrastive learning for aspect-based sentiment analysis, Knowledge-Based Systems 274 (2023) 110648.

[12] X. Zhu, Z. Kuang, L. Zhang, A prompt model with combined semantic refinement for aspect sentiment analysis, Information Processing & Management 60 (2023) 103462.

[13] G. Brauwers, F. Frasincar, A Survey on Aspect-Based Sentiment Classification, ACM Computing Surveys 55 (2023) 65:1–65:37.

[14] A. Kumar, P. Gupta, R. Balan, L. B. M. Neti, A. Malapati, BERT based semi-supervised hybrid approach for aspect and sentiment classification, Neural Processing Letters 53 (2021) 4207–4224.

[15] J. Devlin, M.-W. Chang, K. Lee, 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), ACL, 2019, pp. 4171–4186.

[16] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019), ACL, 2019, pp. 3980–3990.

[17] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV 2017), IEEE, 2017, pp. 2999–3007.

[18] Z. Zhang, M. R. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in: 32nd Annual Conference on Neural Information Processing Systems 2018 (NIPS 2018), Curran Associates, 2018, pp. 8792–8802.

[19] M. Hu, B. Liu, Mining and summarizing customer reviews, in: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), ACM, 2004, pp. 168–177.

[20] A.-M. Popescu, O. Etzioni, Extracting product features and opinions from reviews, in: 2005 Empirical Methods in Natural Language Processing (EMNLP 2005), ACL, 2005, pp. 339–346.

[21] G. Qiu, B. Liu, J. Bu, C. Chen, Opinion word expansion and target extraction through double propagation, Computational Linguistics 37 (2011) 9–27.

[22] Y. Zhao, B. Qin, S. Hu, T. Liu, Generalizing syntactic structures for product attribute candidate extraction, in: 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2010), ACL, 2010, pp. 377–380.

[23] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, The Journal of Machine Learning Research 3 (2003) 993–1022.

[24] C. Lin, Y. He, R. Everson, S. Ruger, Weakly supervised joint sentiment-topic detection from text, IEEE Transactions on Knowledge and Data Engineering 24 (2012) 1134–1145.

[25] F. Huang, C. Yuan, Y. Bi, J. Lu, L. Lu, X. Wang, Multi-granular document-level sentiment topic analysis for online reviews, Applied Intelligence 52 (2022) 7723–7733.

[26] X. Zhao, J. Jiang, H. Yan, X. Li, Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid, in: 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), ACL, 2010, pp. 56–65.

[27] L. Wang, K. Liu, Z. Cao, J. Zhao, G. de Melo, Sentiment-aspect extraction based on restricted Boltzmann machines, in: 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015), ACL, 2015, pp. 616–625.

[28] T. Zhou, K. Law, D. Creighton, A weakly-supervised graph-based joint sentiment topic model for multi-topic sentiment analysis, Information Sciences 609 (2022) 1030–1051.

[29] R. He, W. S. Lee, H. T. Ng, D. Dahlmeier, An unsupervised neural attention model for aspect extraction, in: 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), ACL, 2017, pp. 388–397.

[30] J. Huang, Y. Meng, F. Guo, H. Ji, J. Han, Weakly-supervised aspect-based sentiment analysis via joint aspect-sentiment topic embedding, in: 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), ACL, 2020, pp. 6989–6999.

[31] K. Bhattacharjee, R. Gangadharaiah, Document-level supervision for multi-aspect sentiment analysis without fine-grained labels, arXiv preprint arXiv:2310.06940 (2023).

[32] S. Tulkens, A. van Cranenburgh, Embarrassingly simple unsupervised aspect extraction, in: 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), ACL, 2020, pp. 3182–3187.

[33] W. Zhang, Y. Deng, B. Liu, S. J. Pan, L. Bing, Sentiment analysis in the era of large language models: A reality check, in: Findings of the Association for Computational Linguistics: NAACL-HLT 2024 (Findings NAACL-HLT 2024), ACL, 2024, pp. 3881–3906.

[34] J. Kocon, I. Cichecki, O. Kaszyca, M. Kochanek, D. Szydlo, J. Baran, J. Bielaniewicz, M. Gruza, A. Janz, K. Kanclerz, A. Kocon, B. Koptyra, W. Mieleszczenko-Kowszewicz, P. Milkowski, M. Oleksy, M. Piasecki, L. Radlinski, K. Wojtasik, S. Wozniak, P. Kazienko, ChatGPT: Jack of all trades, master of none, Information Fusion 99 (2023) 101861.

[35] H. Zhang, Y. Cheah, O. M. Alyasiri, J. An, Exploring aspect-based sentiment quadruple extraction with implicit aspects, opinions, and ChatGPT: A comprehensive survey, Artificial Intelligence Review 57 (2024) 17.

[36] J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in: 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015), ACM, 2015, pp. 43–52.

[37] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2016 task 5: Aspect based sentiment analysis, in: 10th International Workshop on Semantic Evaluation (SemEval 2016), ACL, 2016, pp. 19–30.

[38] H. Bashiri, H. Naderi, LexiSNTAGMM: An unsupervised framework for sentiment classification in data from distinct domains, synergistically integrating dictionary-based and machine learning approaches, Social Network Analysis and Mining 14 (2024) 102.

[39] H. Bashiri, H. Naderi, Probabilistic temporal semantic graph: A holistic framework for event detection in Twitter, Knowledge and Information Systems 66 (2024) 7581–7607.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: 31st Annual Conference on Neural Information Processing Systems (NIPS 2017), Curran Associates, 2017, pp. 5998–6008.

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), IEEE, 2016, pp. 770–778.

[42] J. Lei Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[43] H. Xu, B. Liu, L. Shu, P. S. 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), ACL, 2019, pp. 2324–2335.

[44] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations (ICLR 2019), OpenReview.net, 2019.

[45] A. Gholamy, V. Kreinovich, O. Kosheleva, Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation, Technical Report, The University of Texas at El Paso, 2018.

[46] J. S. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: 24th Annual Conference on Neural Information Processing Systems (NIPS 2011), Curran Associates, 2011, pp. 2546–2554.

[47] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: 27th Annual Conference on Neural Information Processing Systems (NIPS 2013), Curran Associates, 2013, pp. 3111–3119.

[48] M. M. Trusca, D. Wassenberg, F. Frasincar, R. Dekker, A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention, in: 20th International Conference of Web Engineering (ICWE 2020), volume 12128 of *LNCS*, Springer, 2020, pp. 365–380.

[49] Z. Liu, W. Lin, Y. Shi, J. Zhao, A robustly optimized BERT pre-training approach with post-training, in: 20th China National Conference (CCL 2021), volume 12869 of *LNCS*, Springer, 2021, pp. 471–484.

[50] H. Bashiri, H. Naderi, Comprehensive review and comparative analysis of transformer models in sentiment analysis, Knowledge and Information Systems 66 (2024) 7305–7361.