IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 11, 2021, accepted May 20, 2021, date of publication June 14, 2021, date of current version July 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3089099

Multi-Domain Aspect Extraction Using Bidirectional Encoder Representations From Transformers

BRUCCE NEVES DOS SANTOS, RICARDO MARCONDES MARCACINI, AND SOLANGE OLIVEIRA REZENDE

Institute of Mathematics and Computer Sciences, University of São Paulo, São Carlos 13566-590, Brazil

Corresponding author: Brucce Neves dos Santos (brucce.neves@usp.br)

This work was supported in part by the National Council for Scientific and Technological Development (CNPq) under Grant 426663/2018-7, and in part by the São Paulo Research Foundation (FAPESP) under Grant 2019/25010-5 and Grant 2019/07665-4.

ABSTRACT Deep learning and neural language models have obtained state-of-the-art results in aspects extraction tasks, in which the objective is to automatically extract characteristics of products and services that are the target of consumer opinion. However, these methods require a large amount of labeled data to achieve such results. Since data labeling is a costly task, there are no labeled data available for all domains. In this paper, we propose an approach for aspect extraction in a multi-domain transfer learning scenario, thereby leveraging labeled data from different source domains to extract aspects of a new unlabeled target domain. Our approach, called MDAE-BERT (Multi-Domain Aspect Extraction using Bidirectional Encoder Representations from Transformers), explores neural language models to deal with two major challenges in multi-domain learning: (1) inconsistency of aspects from target and source domains and (2) context-based semantic distance between ambiguous aspects. We evaluated our MDAE-BERT considering two perspectives (1) the aspect extraction performance using F1-Macro and Accuracy measures; and (2) by comparing the multi-domain aspect extraction models and single-domain models for aspect extraction. In the first perspective, our method outperforms the LSTM-based approach. In the second perspective, our approach proved to be a competitive alternative compared to the single-domain model trained in a specific domain, even in the absence of labeled data from the target domain.

INDEX TERMS Aspect extraction, multi-domain, BERT, transfer learning.

I. INTRODUCTION

Opinion Mining is the task of extracting opinions or sentiments from unstructured texts using Natural Language Processing (NLP), Text Mining, and Machine Learning. The key idea is to analyze automatically large review datasets to classify them into sentiment polarities (i.e., positive, negative, or neutral) [1], [2].

Opinions are formally defined as a 5-tuple $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$, where e_i is the i-th entity of the opinion, a_{ij} is the j-th aspect on which the opinion is related, s_{ijkl} is the sentiment contained in the opinion that can be positive, negative or neutral, h_k is the k-th holder of the opinion, and t_l is temporal information about the opinion [1]. This definition can be applied to different levels of sentiment analysis. The most general level is called the document level and aims to evaluate the overall sentiment of a textual document (e.g., text review) [3], [4].

Sentiment analysis can also be performed at the sentence level, which consists of dividing a document into several sentences and evaluating the sentiment of each sentence individually [3], [4]. Finally, aspect-based sentiment analysis is the most granular level aiming to evaluate the sentiment of entities and aspects [5]–[7]. Figure 1 presents an example of aspect-based sentiment analysis, where ''laptop'' is the entity and ''screen'' is the aspect extracted from the review. Note that sentiment analysis can fail if performed at the document or sentence level.

Aspect-based sentiment analysis is the most promising scenario for Opinion Mining, providing a detailed analysis of consumers' opinions about products and services [8]. However, it is a more complex scenario due to the requirement to extract aspects from the reviews before sentiment classification.

The associate editor coordinating the review of this manuscript and approving it for publication was Thanh Long Vu.

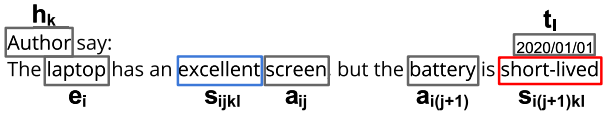


FIGURE 1. Example of aspect-based sentiment analysis, where "laptop" is the entity (e_i), "screen" is the aspect (a_{ij}) with positive sentiment, and "battery" is another aspect ($a_{i(j+1)}$) with negative sentiment.

Aspect extraction from opinion texts is a crucial step in opinion mining. The first proposed initiatives are methods based on linguistic rules, generally used in conjunction with Part-of-Speech tools [9]. More recently, machine learning-based methods have been recognized as state-of-the-art due to the automatic learning of complex patterns from textual features for aspect extraction. In particular, deep learning and neural language models have obtained the best results in aspect extraction tasks [10]–[15]. However, these methods require a large amount of labeled data to achieve such results. Since data labeling is a costly task, there are no labeled datasets available for all domains. A domain in this work is defined as a product category [16], for instance, restaurants, hotels, and smartphones.

A promising alternative that has been used successfully in other areas to deal with the lack of labeled data is the multi-domain knowledge transfer [17]–[21], where labeled data from already known domains is used to learn a model to classify data from a new domain [7]. Thus, multi-domain transfer learning methods should only use labeled data from a source domain and do not require labeled data from the target domain.

In this paper, we propose an approach for aspect extraction in a multi-domain transfer learning scenario. We identified two significant challenges in this scenario: (1) inconsistency of aspects between the target and source domains and (2) semantic distance between aspects. The first challenge occurs mainly when the target and source domains have very different characteristics. For example, aspects extracted from opinions about smartphones (source domain) will be inconsistent with aspects extracted from opinions about hotels (target domain). The second challenge is related to the terms that can have different meanings in different domains. For example, ''light'' can mean ''the lighting of an environment'' in a domain or ''the weight of an object'' in another domain.

To address the challenges of aspect extraction with multi-domain transfer learning, our approach explores the BERT (Bidirectional Encoder Representations from Transformers) neural language model [22], which has been obtaining promising results in several NLP tasks. The structure of the BERT neural network is organized in different layers, where each layer allows to capture information at the level of sentence, syntactic, and semantic characteristics [23]–[25]. We note that the BERT model is useful in dealing with both inconsistent aspects and semantic differences. Our approach, called MDAE-BERT (Multi-Domain Aspect Extraction using Bidirectional Encoder Representations from Transformers), focuses more on the linguistic patterns existing in the BERT layers regarding the inconsistency of aspects between source and target domains. Regarding semantic differences, our approach explores BERT contextual representations, in which the semantic proximity among aspects is determined according to the context of the opinion.

The results are presented and discussed considering two perspectives (1) the aspect extraction performance using F1-Macro and accuracy measures; and (2) comparison between the multi-domain aspect extraction model and the single-domain model, thereby aiming to evaluate whether the use of multi-domain data for training is a competitive alternative if compared to the cost of labeling data for a new domain. Our proposed approach achieved a competitive performance compared to the baseline (LSTM-based aspect extraction), obtaining a minimum increase in aspect extraction performance of 7.99% for F1-Macro and 10.62% for accuracy. Moreover, our MDAE-BERT approach is also a competitive alternative compared to single-domain models, even in the absence of labeled data from the target domain.

The rest of the paper is organized as follows. Section 2 presents background and related studies. The proposed MDAE-BERT approach is described in Section 3. In Section 4, an experimental evaluation of the proposed method is carried out, as well as a comparison with competitive baselines as (1) LSTM-based models for multiple domains and (2) BERT-based models for single domains. Section 5 presents the conclusions and directions for future work.

II. BACKGROUND AND RELATED STUDIES

Opinion Mining aims to extract opinions from unstructured texts, combining Natural Language Processing and Machine Learning techniques. Several levels for sentiment analysis have been proposed since the advent of Opinion Mining. First, document-level sentiment analysis aims to classify the sentiment polarity (e.g., positive, negative, or neutral) of a whole document [1]. Second, sentence-level sentiment analysis divides the document into sentences, and the polarity of each sentence is classified [3], [4]. Sentence-level sentiment analysis is analogous to the previous one because a sentence can be seen as a short document. Sentiment analysis at document or sentence levels failed to deal with phrases with more than one aspect with different sentiments. Thus, sentiment analysis at the aspect level emerged, which is the most granular level. Aspect-based sentiment analysis can be divided into two tasks: (1) extract the entities and aspects from text opinions; and (2) classify the sentiment polarity in the context of each aspect [5]–[7].

In aspect extraction tasks, we have to deal with explicit and implicit aspects [26]. Explicit aspects are words or expressions that directly refer to a technical characteristic of a product or service. For example, in ''the battery has a good life'', the aspect ''battery'' is a technical feature explicitly used by the reviewer. On the other hand, implicit aspects are indirect references to characteristics of products or services, usually through expressions about the behavior of the aspect.

For example, in the text ''My photos are too dark'', the opinion is about the smartphone's implicit aspect ''camera''.

Traditional approaches for aspect extraction are based on linguistic rules, which usually depend on lexicons, part-of-speech tagging, and token dependency relationships. These linguistic resources are often unavailable for multiple domains, as well as being hampered in the presence of reviews with typos, slang, and abbreviation obtained from social platforms [27]–[36]. Moreover, methods based on lexicons, ontologies, or other external resources are useful for explicit aspect extraction, but fail to extract implicit aspects. However, recent methods based on neural language models allow extracting both explicit and implicit aspects by taking advantage of pre-trained models and more complex textual representations, such as word embeddings.

We focus on recent studies that involve two types of deep neural networks for learning neural language models. First, we explore Long Short-Term Memory (LSTM) networks and their variants that allow dealing with long sequences of tokens and used in methods such as ELMO (Embeddings from Language Models) [37]. Second, we explore Transformers networks, which in addition to dealing with token sequences, also use attention mechanisms and provide parallelization for scalable training from a large textual corpus. Both LSTM and Transformers have been successfully explored in the context of aspect-based sentiment analysis, as discussed in the following subsections.

A. LSTM-BASED ASPECT EXTRACTION

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) with loops that allows the persistence of information from long sequences, such as token sequences extracted from reviews. An LSTM unit is called a memory cell or LSTM cell, as illustrated in Figure 2, which has a vector called the cell state to store information, and its value is changed based on the gates. The input gate protects the stored information against irrelevant disturbances. The forget gate selects which pieces of information in the cell state are important and which information should be forgotten. The output gate protects other units from disorders caused by irrelevant content stored in the cell and decides which parts of the cell state are important to generate the output. These gates regulate the flow of information into and out of the cell [38].

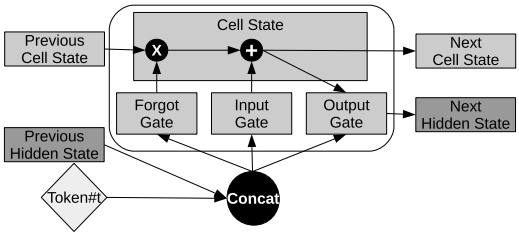


FIGURE 2. Illustration of the LSTM cell.

The flow starts with the forget gate using a sigmoid function to decide which information should be removed from the cell. In the next step, the input gate determines what information should be stored in the cell. Finally, the output gate decides whether the cell information will be visible to other cells.

In [11], the authors evaluated different LSTM network architectures for aspect extraction and presented an experimental comparison involving traditional machine learning-based methods, such as CRF (Conditional Random Field) models. The results showed that methods based on neural language models are more efficient for aspect extraction, even without hand-crafted features.

Several methods for aspect extraction using LSTM networks have been proposed recently. In [13], the authors included a local context strategy based on a window of words surrounding the aspect during the LSTM training. An experimental analysis on bidirectional LSTM for aspect extraction tasks was presented by [39], in which LSTM-based aspect extraction obtained results superior to the top-ranked methods in the 2014 SemEval ABSA aspect extraction contest.[1] Li *et al.* [40] discussed the impact of attention mechanisms in LSTM networks to refine the process of aspect extraction. Ma *et al.* [41] proposed an approach based on LSTM networks for aspect extraction and evaluated the impact of external knowledge resources to improve neural language modeling in sentiment analysis. A recent survey by Nazir *et al.* [8] presents an overview of aspects-based sentiment analysis techniques, where they discussed that LSTM models are very promising for learning implicit knowledge from reviews because they work similarly to the human brain to understand the importance of each word in the review — a useful resource for identifying aspects in reviews.

Although LSTM-based aspect extraction methods have received significant attention in the literature, their use is underexplored in multi-domain scenarios. Another recognized limitation is that LSTM has drawbacks related to scalability since sequential dependence impairs the parallelization of neural network training. Transformers networks [42] is an alternative considered state-of-the-art for natural language processing, which surpasses the drawbacks of LSTM networks.

B. TRANSFORMERS-BASED ASPECT EXTRACTION

Transformers are an evolution of the encoder-decoder architecture to handle sequential data, such as text reviews [42]. Unlike RNN's where the words are presented one at a time in a sequence and the current state depends on the result of the previous state, in the Transformers encoder, words of the sentence are processed in parallel to learn text embeddings, as illustrated in Figure 3.

Word order is crucial in natural language processing (NLP) tasks since words in different positions may have different meanings. In order to parallelize the processing of textual data, Transformers networks introduce a positional encoder to map the distance between the tokens of the sentence. Figure 4 shows the complete architecture of the Transformers network.

[1] https://alt.qcri.org/semeval2014/task4/

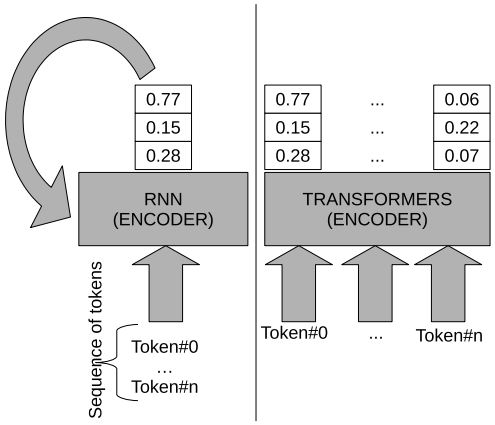


FIGURE 3. Illustration of the difference between RNN and transformers.

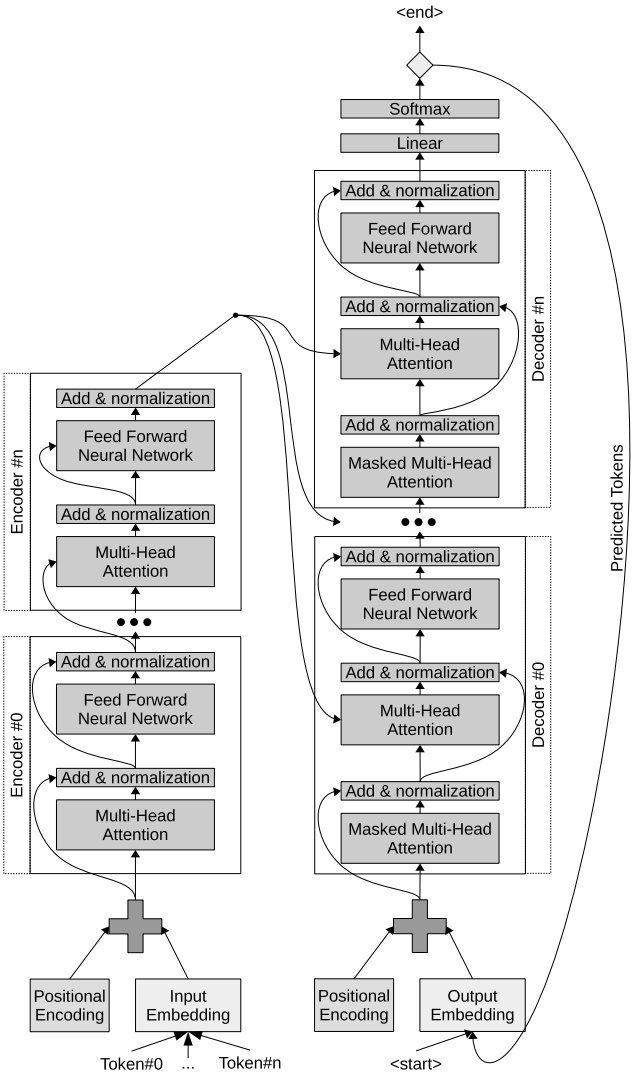


FIGURE 4. Architecture of the transformers network.

Regarding the Transformers encoder step, the **Input Embedding** receives the sentence words and maps them to their respective word embedding. A **Positional Encoding** adds the respective position vector, thereby producing a contextual word embedding considering the word positions of the sentence.

The encoder consists of n identical layers. The original paper uses $n = 6$. The input of the first layer is the contextual word embeddings, and the input of subsequent layers is the output of the previous layer. Each layer is described below.

- **Multi-Head Attention:** This layer consists of x Self-Attention layers that are processed in parallel. The Self-Attention layers complement each other. The original paper uses $x = 8$. Each word will have x attention vectors and these vectors will be concatenated and multiplied by another vector to obtain the final result. In short, the input and output of the layer are attention vectors for each word, thus indicating the most relevant words in the sentence.
- **Feed Forward:** It consists of several Feed Forward Neural Networks (FFNN), which process each word of the sentence in parallel. Each FFNN is a simple neural network with two layers fully connected with a ReLU activation.
- **Add & Normalization:** This layer receives the input vectors and the output vectors from the previous layer to generate residual connections in order to optimize deep network training. Normalization is carried out on these vectors to facilitate optimization and ensure that the Positional Encoder remains stable throughout the process.

After the encoder step, the decoder step is executed for the next word prediction tasks. The decoder aims to predict one word at a time until the special token "<end>" is predicted. Unlike the encoder, which is only executed once, the decoder has to be performed k times, where the predicted word will feed the decoder in the next step. In **Output Embedding**, the decoder receives the words of the sentence that have already been predicted. The first word is the special token "<start>". The **Positional Encoding** of the decoder is a similar operation to the encoder. The decoder contains sub-layers, as described below.

- **Masked Multi-Head Attention:** It works similarly to the encoder sub-layer "Multi-Head Attention". The main difference is to prevent future words from being part of the attention. In other words, it uses only the words that have already been predicted in the attention mechanism.
- **Multi-Head Attention:** It works in the same way as the encoder. However, one of the inputs is the output of the last encoder.
- **Feed Forward:** Similar operation to the encoder step.
- **Add & Normalization:** Similar operation to the encoder step.

The decoder's output at this point is a numeric vector which is processed by two layers:

- **Linear:** It is a fully connected neural network that receives the output vector from the last decoder and projects it into a large vector called "logits vector".

The dimension of this vector is according to the vocabulary size.

- **Softmax:** This layer receives ''vector logits'' and turns each word's score into probability, thereby selecting the word associated with the highest chance. The output from the softmax layer will be added to the ''Output Embedding'' until the word '' <end> '' is produced.

Devlin *et al.* [22] argue that one limitation of Transformers is the fact that it is unidirectional, thereby restricting the choice of architectures that can be used during pre-training.

BERT (Bidirectional Encoder Representations from Transformers) is a pre-training language representation method with the general purpose of generating a ''language understanding'' model. First, BERT is trained through large unlabeled text corpus. Second, we can refine the model for a specific task, in a task called fine-tuning [22]. BERT is available in two architectures as described in Table 1.

TABLE 1. BERT's model architecture.

	Transformers Encoder	Multi-Headed Attention	Dimension Input/Output	Feed Forward layers
BERT_{base}	12	12	768	4
BERT_{large}	24	16	1024	4

BERT consists of a vocabulary of 30000 tokens from WordPiece embeddings [43]. If a word is not in the vocabulary, then that word will be divided into sub-words that exist in the vocabulary. These sub-words can even be a single character. The first 1000 tokens are reserved, and except for five tokens, the other tokens are in the form ''[unused2]''. These five tokens are:

PAD: Token used for padding, for example, when there are sequences of different sizes.

UNK: Token ''unknown'' used to represent a token that is not in the vocabulary.

CLS: Token ''classifier'' used for sequence classification instead of token classification. In general, it is the first token in the sequence.

SEP: Token ''separator'' is used to construct a sequence from sub-sequences. For example, two sequences for sentence classification or question answering tasks. It is also used as the last token in a sequence.

MASK: Token used to mask values. This token is used to replace the word that the model will attempt to predict.

BERT training can be divided into two stages:

- **Pre-Training:** BERT is pre-trained using two unsupervised tasks: (1) ''Masked Language Model (MLM)'' where a percentage of the input tokens are selected at random and replaced with the [MASK] token. The purpose of this task is to predict the masked tokens; and (2) ''Next Sentence Prediction (NSP)'' that receives the sentences A and B to predict if the sentence B is the next sentence of A. This training is important since understanding the relationship between the two sentences is vital in tasks like ''Question Answering (QA)'' and ''Natural Language Inference (NLI)''.
- **Fine-Tuning:** This stage uses a model that has been trained in a large text corpus and carries out more training using the domain-application texts and labels. Compared to the Pre-Training stage, Fine-Tuning has a low cost.

In [14], BERT is evaluated for sentiment analysis tasks to compare the impact of BERT fine-tuning. The results demonstrate that BERT fine-tuning improves the performance of sentiment analysis tasks, including aspect extraction. Zhang and Shi [44] recently evaluated BERT fine-tuning for aspect extraction task. In particular, the authors concluded that aspect extraction using contextual word embeddings from BERT model obtained state-of-the-art results for single-domain aspect extraction, compared to LSTM and CRF-based models. In [15], the authors evaluated the BERT fine-tuning using labeled reviews from different domains. The promising results provide evidence that BERT-based models are efficient for classifying whether a given aspect belongs to a specific domain. Although it is not a specific proposal for multi-domain aspect extraction, these results motivated us to investigate BERT in scenarios in which a target domain does not have labeled aspects.

III. MDAE-BERT: MULTI-DOMAIN ASPECT EXTRACTION USING BERT MODEL

The MDAE-BERT innovates by using labeled data from multiple domains for fine-tuning a pre-trained language model. We unify labeled aspects from different domains into a token classification task. We use the IOB representation for token labels. The IOB is short for inside, outside, and beginning. Each token of a text in the training set is labeled with **class B**, which indicates that the token represents the beginning of an aspect; **class I**, which indicates that the token is inside an aspect; and **class O**, which indicates that the token is outside an aspect. The IOB representation is illustrated in Tables 2 and 3 for aspects formed by a single and multiple tokens, respectively.

TABLE 2. Example of IOB labels for a review with an aspect (screen) formed by a single token.

The	screen	is	amazing	.
O	B	O	O	O

TABLE 3. Example of IOB labels for a review with an aspect (battery life) formed by multiple tokens.

I	would	like	at	least	a	4	hr	battery	life	.
O	O	O	O	O	O	O	O	B	I	O

Let D_s and D_t be two domains, where D_s indicates the source domain and D_t the target domain. In the source domain, $D_s^k = \{(\mathbf{x}_i, \mathbf{x}_i^a, y_i)_s^k\}_{i=1}^{N_s^k}$, where $\mathbf{x} = (x_1, x_2, \ldots, x_S)$ represents the token sequence of size S of a review belonging

to the k-th source domain, $\mathbf{x}^a = (x_1^a, x_2^a, \ldots, x_M^a)$ represents the aspect, where $\mathbf{x}_a$ is a subsequence of size S (with $1 \leq M \leq S$) from $\mathbf{x}$, and y indicates the IOB tag of the token. N_s^k indicates the number of reviews with labeled aspects in the k-th source domain. The target domain is composed by reviews with unlabeled aspects $D_t = \{(\mathbf{x}_i^t)\}_{i=1}^{N_t}$, where N_t indicates the number of reviews in the target domain. The aspect extraction task can be formulated as a token level sequence labeling problem, where given a review of the target domain $\mathbf{x}^t \in D_t$, for each token $x \in \mathbf{x}^t$, the task aims to find a label $y \in \{I, O, B\}$.

Our approach uses the BERT model to encode the review token sequence into a vector representing semantic and linguistic information. As we discussed in Section II, BERT is a pre-trained language model based on deep bidirectional Transformers using two prediction tasks: the masked language model and the next sentence prediction. Below, we present details of how each task is explored in the MDAE-BERT approach.

A. CONTEXTUAL WORD EMBEDDINGS FOR MULTI-DOMAIN ASPECT EXTRACTION

Learning word embeddings for text reviews is a crucial step for aspect extraction. We focus on contextual word embeddings, in which the vector representation of a word is a function of the entire text review in which it occurs. Extracting aspects through their semantic representation is a strategy to deal with the limitations presented in the introduction, particularly the inconsistency between aspects from target and source domains.

In the proposed MDAE-BERT approach, we extend the BERT masked-language model to the multi-domain scenario. Given a token sequence of the review $\mathbf{x} \in D_s^k$, the BERT model first generates a corrupted version $\hat{\mathbf{x}}$ of the review, where approximately 15% of the tokens are randomly selected and replaced by a special token called [MASK]. Thus, the objective function is to reconstruct the masked tokens $\bar{\mathbf{x}}$ from $\hat{\mathbf{x}}$, according to Equation 1,

$$\max_{\theta} \log p(\bar{\mathbf{x}}|\hat{\mathbf{x}}, \theta) \approx \sum_{\mathbf{x} \in D_s^k} \sum_{t=1}^{S} m_t \log \times \left(\frac{\exp(h_\theta(\hat{\mathbf{x}})_t^\top e(x_t))}{\sum_{x'} \exp(h_\theta(\hat{\mathbf{x}})_t^\top e(x'))} \right) \quad (1)$$

where $e(x_t)$ indicates the word embedding vector of the token x_t; the $h_\theta(\hat{\mathbf{x}})_t$ is a sequence of S hidden state vectors according to parameters θ from the Transfomers neural network model; and $m_t = 1$ indicates when x_t is masked.

Note that we aggregate reviews from all source domains into a word embedding learning strategy based on denoising auto-encoding. By masking tokens considering multiple domains, we force the model to generate word embeddings considering the context, particularly the review domain in which the word occurs, since the model tries to predict a masked word that may occur ambiguously in different domains.

B. TOKEN CLASSIFICATION FOR ASPECT EXTRACTION

While the previous step focuses on learning contextual word embeddings, now we define the specific classification task that guides the fine-tuning of the pre-trained BERT model. In general, BERT-based classification tasks use a special token called [CLS] at the beginning of each review. The corresponding word embedding of the [CLS] token is used as a vector representation of the entire sentence, considering both the semantics of the text and the linguistic structure of the sentence.

The same sentence should have different representations in the token classification from reviews in IOB tags according to the considered aspect tokens. Thus, we formulated the token classification as a sentence pair classification task to obtain vector representations of reviews according to the aspect, similar to the next sentence prediction task used in BERT.

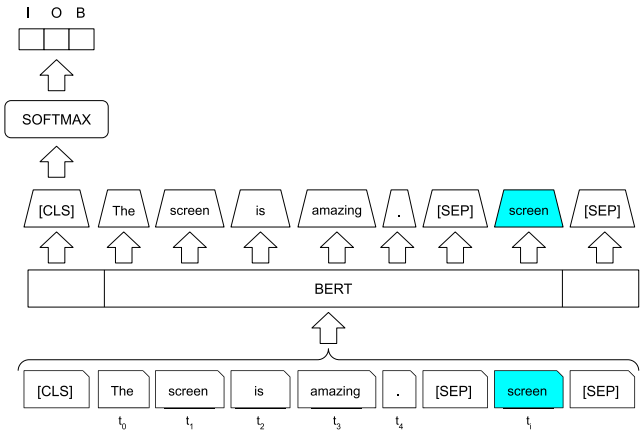


FIGURE 5. Example of MDAE-BERT input for aspect extraction.

Given a review $\mathbf{x} = (x_1, x_2, \ldots, x_S)$ and an aspect $\mathbf{x}^a = (x_1^a, x_2^a, \ldots, x_M^a)$, we use the special [CLS] token at the beginning of the review and include a special [SEP] token after the review to connect the aspect tokens, as shown in Figure 5. Thus, the input $\mathbf{x}^r$ for fine-tuning of the BERT model is as follows:

$$\mathbf{x}^r = \{[CLS], x_1, x_2, \ldots, x_S, [SEP], x_1^a, x_2^a, \ldots, x_M^a, [SEP]\}$$

Now, we can use BERT to encode the input sequence $\mathbf{x}^r$, where the vector corresponding to the [CLS] token considers both the review and aspect sequences. Let $\mathbf{x}_{CLS}^r = BERT(\mathbf{x}^r)$ the corresponding vector of the review-aspect representation, we use $\mathbf{x}_{CLS}^r$ as the input of a classifier composed of a dense layer followed by a softmax layer to classify the aspect in the IOB tags, according to Equations 2 and 3, respectively,

$$\mathbf{L} = tanh(\mathbf{W}_x \mathbf{x}_{CLS}^r + \mathbf{b}_x) \quad (2)$$

$$\hat{y} = softmax(\mathbf{W}_L \mathbf{L} + \mathbf{b}_L) \quad (3)$$

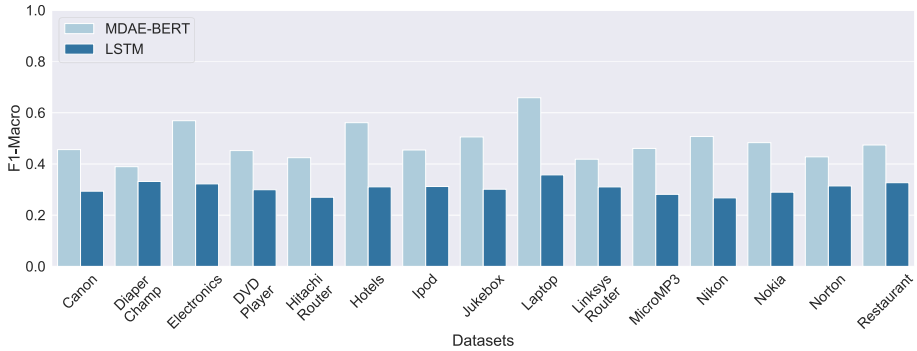


FIGURE 6. Overview of the effectiveness (F1-Macro) of each analyzed method for multi-domain aspect extraction. The X-axis means which dataset was used for testing.

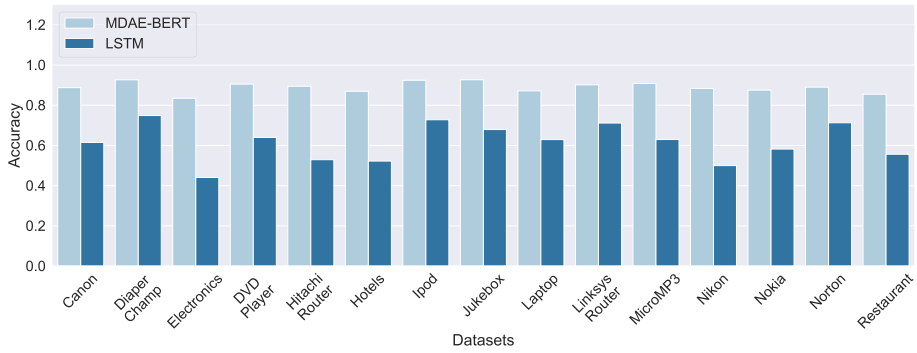


FIGURE 7. Overview of the effectiveness (Accuracy) of each analyzed method for multi-domain aspect extraction. The X-axis means which dataset was used for testing.

where $\hat{y}$ is the estimated probability distribution of the aspect in relation to the IOB tags, and $\mathbf{W}_x$, $\mathbf{W}_L$, $\mathbf{b}_x$, $\mathbf{b}_L$, are weight parameters of the neural network to be optimized.

For model training, while tags B and I are directly estimated from labeled aspect tokens, we sample a subset of review tokens for tag O. In this sampling, we use all tokens in a review that are not aspects.

Our MDAE-BERT model is trained using cross-entropy loss, as defined in Equation 4,

$$\mathcal{L} = -\sum_{D_s^k \in D_s} \sum_{\mathbf{x}^r \in D_s^k} \sum_{c=1}^{C} y_{\mathbf{x}^r}^c \, log(\hat{y}_{\mathbf{x}^r}^c) \tag{4}$$

where C indicates integer codes for each IOB tag, $\hat{y}_{\mathbf{x}^r}^c$ is the predicted probability for input $\mathbf{x}^r$, and $y_{\mathbf{x}^r}^c$ is a value of 1 or 0 that indicates when the IOB tag was correctly predicted by the model.

IV. EXPERIMENTAL ANALYSIS AND DISCUSSION

While most works in the literature evaluate aspect extraction tasks using two datasets (Laptops and Restaurants datasets from 2014 SemEval ABSA aspect extraction context), we collect and provide a repository[2] containing several review datasets of different domains with labeled aspects. In our experimental evaluation, we use 15 datasets (domains) containing reviews in English, as presented in Table 4. In this table, O indicates the number of non-aspect words, B indicates the number of aspects, and I indicates how many words are the continuation of an aspect (i.e., IOB tags). The training process consists of a cross-validation approach at the domain level, where 14 domains were used to train the model, and 1 domain is used to evaluate the aspect extraction task.

In each dataset, all duplicate sentences were removed. A sentence is considered duplicate if another sentence presents the same words in the same order, disregarding any punctuation. For example, the following sentences are considered duplicated: "The Screen is amazing.", "the screen is amazing!!!!" and "ThE, ScReEn, Is, AmAzIng!!!...". It was also guaranteed that there are no duplicate sentences between the datasets. For example, "the battery has a great life" could exist in the Laptop dataset and the MicroMP3 dataset. Stopwords and punctuations were labeled with the "O" tag.

Our MDAE-BERT uses BERT_{base} architecture. LSTM-based aspect extraction uses the GloVe[3] word embedding. Each input review for training step consists of 3 information pieces:

- Sentence with token t_i replaced by the marker "T".
- Token t_i
- Token label t_i in the IOB format.

The results are presented and discussed considering (1) the classification performance using the F1-Macro and Accuracy measures, and (2) a comparison of our MDAE-BERT

[2]https://github.com/BrucceNeves/MDAE-BERT

[3]http://nlp.stanford.edu/data/wordvecs/glove.42B.300d.zip

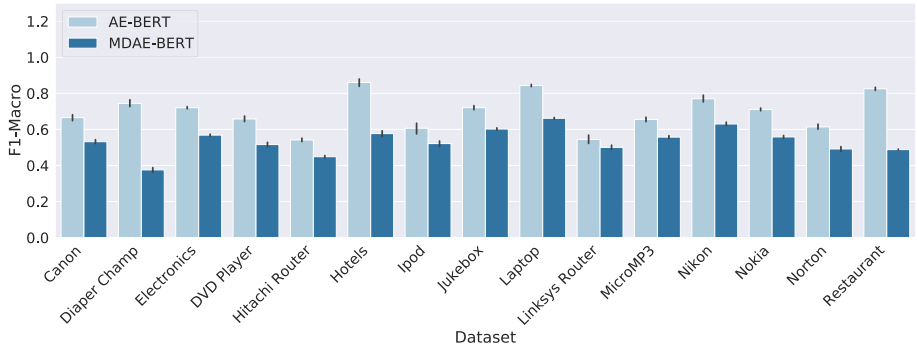


FIGURE 8. Comparison between classification efficiency (F1-Macro) using a model trained in one domain compared to the multi-domain model. The error bars illustrate the standard deviation. The X-axis indicates the domain that was used for testing, in the case of the multi-domain.

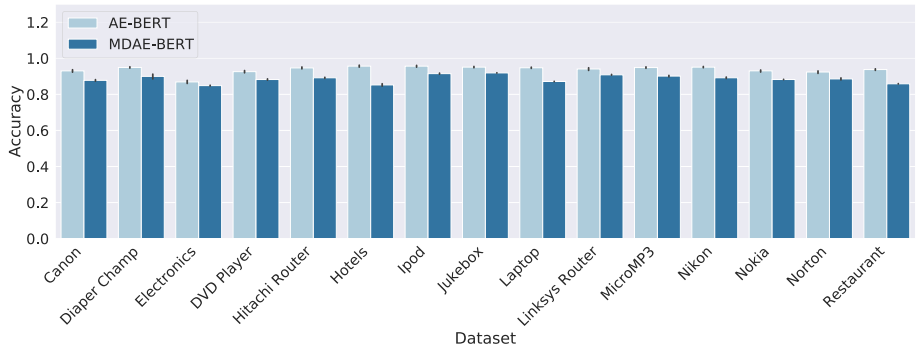


FIGURE 9. Comparison between the efficiency (accuracy) of classification using a model trained in one domain compared to the multi-domain model. The error bars illustrate the standard deviation. The X-axis indicates the domain that was used for testing, in the case of the multi-domain.

(which does not use labeled data from the target domain) and a BERT-based aspect extraction model (AE-BERT) that incorporates labeled data from the target domain.

Figures 6 and 7 show the experimental results (F1-Macro and Accuracy measures) for multi-domain aspect extraction. The proposed MDAE-BERT approach obtained a superior performance compared with the LSTM, obtaining a minimum increase in the aspect extraction performance of 7.99% for F1-Macro and 10.62% for Accuracy. The best result of MDAE-BERT was for the Laptop target domain with 65.86% F1-Macro, thereby achieving almost 90% improvement over LSTM.

The second experiment aims to assess whether using our MDAE-BERT is an alternative if compared to the cost of labeling data for a new domain. In this scenario, we use 90% of labeled aspects of the target domain to train a BERT-based aspect extractor (AE-BERT). For a fair comparison, the MDAE-BERT was evaluated using the same cross-validation test folds, i.e., using the 10% remaining reviews from the target domain. The results are illustrated in Figures 8 and 9, which respectively compare F1-Macro and Accuracy.

As expected, the AE-BERT obtained higher values of F1-Macro and Accuracy since it is trained with data from the target domain. However, it is worth noting that in this scenario, the AE-BERT is a hypothetical model, thereby indicating the result obtained if it were possible to label aspects of the target domain. Even without labeled aspects of the target domain, MDAE-BERT obtained competitive results for most domains.

Table 5 presents the inconsistency between the domains used in the training and the target domain. The inconsistency

TABLE 4. Overview of the reviews datasets used in the experimental evaluation.

Dataset	#Reviews	O	B	I
Canon	886	16627	465	91
Diaper Champ	373	6609	218	48
Electronics	3183	44400	6862	3083
DVD Player	726	11872	338	83
Hitachi Router	311	5499	238	4
Hotels	226	2605	215	59
Ipod	527	11467	192	34
Jukebox	1673	31263	721	135
Laptop	1898	31620	3008	1445
Linksys Router	567	10449	216	29
MicroMP3	988	19815	360	78
Nikon	346	6409	175	54
Nokia	1090	19083	887	141
Norton	380	6877	224	52
Restaurant	3644	52489	5534	2087

TABLE 5. Overview of the inconsistency levels between source and target domain in the experimental evaluation. High values mean high inconsistency.

Target Domain	Target Aspects	Training Aspects	Common Aspects	Level of Inconsistency
Canon	167	4346	139	16,77%
Diaper Champ	71	4339	36	49,30%
Electronics	2061	3096	783	62,01%
DVD Player	130	4344	100	23,08%
Hitachi Router	95	4344	65	31,58%
Hotels	88	4337	51	42,05%
Ipod	114	4351	91	20,18%
Jukebox	195	4334	155	20,51%
Laptop	959	3973	558	41,81%
Linksys Router	100	4350	76	24,00%
MicroMP3	157	4357	140	10,83%
Nikon	80	4364	70	12,5%
Nokia	337	4258	221	34,42%
Norton	122	4333	81	33,61%
Restaurant	1575	3081	282	82,10%

was calculated according to the measure proposed in [7] by computing the number of aspects that the source and target domains have in common. In general, when there is less inconsistency between the target and source domains, there is also less difference in performance between AE-BERT and MDAE-BERT, indicating that MDAE-BERT can learn related concepts from other domains. When the inconsistency is high, as in the target domains ''restaurant'', ''diaper champ'' and ''hotels '', the MDAE-BERT presents some limitations in transferring knowledge among domains. Analyzing multi-domain learning in inconsistent domains motivates further research. We argue that pre-training BERT models from review datasets instead of general domain texts is a promising way to deal with such knowledge transfer limitations.

V. CONCLUDING REMARKS

Multi-domain aspect extraction is a promising solution to mitigate the cost of labeling data for aspect-based sentiment analysis. The proposed MDAE-BERT proved to be a competitive alternative for this task since it uses existing labeled data from other domains to train an aspect extraction model. MDAE-BERT obtained results superior to the LSTM-based aspect extraction for multiple domains. In addition, MDAE-BERT was competitive with AE-BERT, which considers labeled data from the target domain.

We note that aspect extraction results can be improved when exploring the MDAE-BERT confidence scores. For example, if a new target domain has a higher inconsistency in relation to source domains, it is possible to increase the classification process's confidence threshold by each aspect. In this case, only aspects classified with greater confidence for classes B or I would be extracted, thereby increasing the precision (to the detriment of the recall). This strategy will be explored in future works. Another future work to mitigate inconsistency between domains is to refine the pre-training stage of BERT using a large corpus of reviews instead of a general-purpose text corpus.

Finally, another contribution is a pre-trained multi-domain aspect extraction model available at https://github.com/BrucceNeves/MDAE-BERT. This model can be used in English reviews to extract aspects in an unsupervised way. In addition, it is also prepared for a fine-tuning process to refine the model for aspect extraction if new labeled data are available.

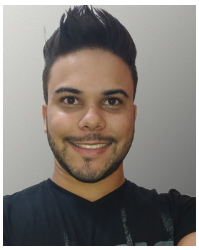
VI. ACKNOWLEDGMENT

The authors would like to thank the NVIDIA for donating computer equipment (GPU Grant Academic Program).

REFERENCES

[1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.
[2] C. C. Aggarwal, "Opinion mining and sentiment analysis," in *Machine Learning for Text*. Boston, MA, USA: Springer, 2018, pp. 413–434.
[3] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 105–112.
[4] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 129–136.
[5] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, Apr. 2013.
[6] I. P. Matsuno, R. G. Rossi, R. M. Marcacini, and S. O. Rezende, "Aspect-based sentiment analysis using semi-supervised learning in bipartite heterogeneous networks," *J. Inf. Data Manage.*, vol. 7, no. 2, p. 141, 2016.
[7] R. M. Marcacini, R. G. Rossi, I. P. Matsuno, and S. O. Rezende, "Cross-domain aspect extraction for sentiment analysis: A transductive learning approach," *Decis. Support Syst.*, vol. 114, pp. 70–80, Oct. 2018.
[8] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *IEEE Trans. Affect. Comput.*, early access, Jan. 30, 2020, doi: 10.1109/TAFFC.2020.2970399.
[9] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, Aug. 2020.
[10] O. Irsoy and C. Cardie, "Opinion mining with deep recurrent neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 720–728.
[11] P. Liu, S. Joty, and H. Meng, "Fine-grained opinion mining with recurrent neural networks and word embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1433–1443.
[12] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl.-Based Syst.*, vol. 108, pp. 42–49, Sep. 2016.
[13] J. Yuan, Y. Zhao, B. Qin, and T. Liu, "Local contexts are effective for neural aspect extraction," in *Proc. Chin. Nat. Conf. Social Media Process.* Singapore: Springer, 2017, pp. 244–255.
[14] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Association for Computational Linguistics, 2019, pp. 2324–2335.
[15] M. Hoang, O. A. Bihorac, and J. Rouces, "Aspect-based sentiment analysis using BERT," in *Proc. 22nd Nordic Conf. Comput. Linguistics*, Turku, Finland, Sep./Oct. 2019, pp. 187–196.
[16] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 440–447.
[17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[18] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.
[19] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big data*, vol. 3, no. 1, p. 9, 2016.

[20] P. Zhang, J. Wang, Y. Wang, and Y. Wang, "A statistical approach to opinion target extraction using domain relevance," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 273–277.
[21] T. A. Rana and Y.-N. Cheah, "A two-fold rule-based model for aspect extraction," *Expert Syst. Appl.*, vol. 89, pp. 273–285, Dec. 2017.
[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Association for Computational Linguistics, 2019, pp. 4171–4186.
[23] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. Bowman, D. Das, and E. Pavlick, "What do you learn from context? Probing for sentence structure in contextualized word representations," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.
[24] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MA, USA: Association for Computational Linguistics, 2019, pp. 4129–4138.
[25] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 3651–3657.
[26] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *Proc. 19th Nat. Conf. Artif. Intell. (AAAI)*, vol. 4, 2004, pp. 755–760.
[27] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2006, pp. 43–50.
[28] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural Language Processing and Text Mining*. London, U.K.: Springer, 2007, pp. 9–28.
[29] S. Blair-Goldensohn, T. Neylon, K. Hannan, G. A. Reis, R. Mcdonald, and J. Reynar, "Building a sentiment summarizer for local service reviews," in *Proc. WWW Workshop NLP Inf. Explosion Era*, 2008, pp. 14–23.
[30] S. S. Htay and K. T. Lynn, "Extracting product features and opinion words using pattern knowledge in customer reviews," *Sci. World J.*, vol. 201, Dec. 2013, Art. no. 394758.
[31] Z. Hai, K. Chang, J.-J. Kim, and C. C. Yang, "Identifying features in opinion mining via intrinsic and extrinsic domain relevance," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 623–634, Mar. 2014.
[32] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proc. 2nd Workshop Natural Lang. Process. Social Media (SocialNLP)*, 2014, pp. 28–37.
[33] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, and Z. Gao, "Improving opinion aspect extraction using semantic similarity and aspect associations," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.
[34] Q. Liu, Z. Gao, B. Liu, and Y. Zhang, "Automated rule selection for opinion target extraction," *Knowl.-Based Syst.*, vol. 104, pp. 74–88, Jul. 2016.
[35] O. F. Gunes, T. Furche, and G. Orsi, "Structured aspect extraction," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, vol. 2016, 2016, pp. 2321–2332.
[36] S. M. Jiménez-Zafra, M. T. Martín-Valdivia, E. Martínez-Cámara, and L. A. Ureña-López, "Combining resources to improve unsupervised sentiment analysis at aspect-level," *J. Inf. Sci.*, vol. 42, no. 2, pp. 213–229, Apr. 2016.
[37] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1, 2018, pp. 2227–2237.
[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[39] T. U. Tran, H. T.-T. Hoang, and H. X. Huynh, "Bidirectional independently long short-term memory and conditional random field integrated model for aspect extraction in sentiment analysis," in *Frontiers in intelligent computing: Theory and Applications*. Singapore: Springer, 2020, pp. 131–140.
[40] X. Li, L. Bing, P. Li, W. Lam, and Z. Yang, "Aspect term extraction with history attention and selective transformation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4194–4200.
[41] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis," *Cognit. Comput.*, vol. 10, no. 4, pp. 639–650, Aug. 2018.
[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, K. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[43] Y. Wu, M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, U. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*. [Online]. Available: https://arxiv.org/abs/1609.08144
[44] Q. Zhang and C. Shi, "Exploiting BERT with global-local context and label dependency for aspect term extraction," in *Proc. IEEE 7th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2020, pp. 354–362.



BRUCCE NEVES DOS SANTOS received the master's degree in computer science from the Federal University of Mato Grosso do Sul, Brazil. He is currently pursuing the Ph.D. degree in the Graduate Program in Computer Science and Computational Mathematics (PPG-CCMC) with the University of São Paulo, Brazil. He has published articles in *Pattern Recognition Letters* and computer science conferences. His research interests include data and text mining and machine learning.

RICARDO MARCONDES MARCACINI received the Ph.D. degree in computer science from the Institute of Mathematics and Computer Science, University of São Paulo, Brazil. He is currently a Professor of computer science at the University of São Paulo. He has published articles in a number of international journals and conferences, such as *Decision Support Systems*, *Pattern Recognition Letters*, *Journal of Information and Data Management*, International Conference on World Wide Web, Web Intelligence Conference, and ACM Symposium on Document Engineering. His research interests include machine learning, data clustering, and data analytics systems.

SOLANGE OLIVEIRA REZENDE received the Ph.D. degree in mechanical engineering from the University of São Paulo, Brazil, and the Ph.D. degree in computer science from the University of Minnesota, USA. She is a Full Professor of computer science at the University of São Paulo. She has published articles in a number of international journals, such as *Pattern Recognition Letters*, *Journal of Information and Data Management*, *Knowledge-Based Systems*, *Intelligent Data Analysis*, *Information Retrieval Journal*, and *Information Processing and Management*. Her research interests include data and text mining, machine learning, and recommendation systems.