Word- and Sentence-Level Representations for Implicit Aspect Extraction

Pantelis Agathangelou, Ioannis Katakis, and Panagiotis Kasnesis

***Abstract*—Aspect terms extraction (ATE), a key subtask for aspect-based sentiment analysis, opinion summarization, and topic modeling aims at extracting grammatical elements (nouns, phrases, and adjectives) from user reviews that reveal the discussed features of the entity under review. These aspect terms are usually the targets of the opinions expressed. Identifying them requires tackling substantial linguistic challenges but, due to the multiple commercial and social applications, significant research effort has been invested in efficiently mining aspects. Recent advances in ATE address methods that exploit a sentence or a word-level encoding of a user review as a solution. This article proposes a novel and effective word- and sentence-level encoding framework, which utilizes a neural network architecture that learns to extract aspect terms. The main advantage of our approach is that it can extract explicit and implicit aspects (i.e., aspects that are not directly mentioned in the user-generated text). We evaluate our method on four widely used datasets where we prove its efficiency against state-of-the-art alternative approaches.**

***Index Terms*—Aspect terms extraction (ATE), opinion summarization, topic modeling.**

I. Introduction

THE extraction of aspects is an essential task for the advancement of important natural language analysis processes such as aspect-based sentiment analysis, opinion summarization, and topic modeling. Aspects are grammatical elements, usually nouns or noun phrases that are used in a sentence to represent the features of an entity. They are linked with opinion words and phrases, where the opinion words target their expressive energy to the aspects [1]. For example, in the sentence "This restaurant has a great service," the term "service" is an (explicit) aspect that absorbs the positive energy of the opinion word "great." However, this relation is not always straightforward and opinion phrases can refer to an aspect

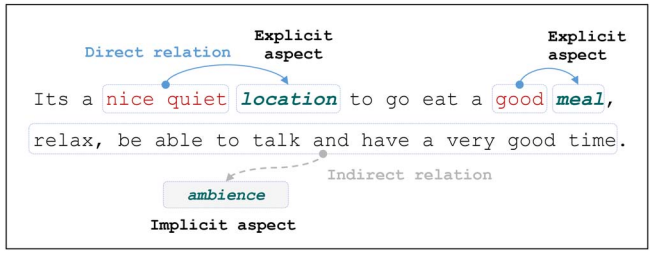

Fig. 1. Implicit and explicit aspects in an example from our datasets. Red: Opinion words; Green: Aspects.

implicitly. For example, in the sentence "Its a nice quiet location to go eat a good meal, relax, be able to talk and have a very good time.," the aspect terms "location" and "meal" refer to implicit aspect here, implying the "ambience" which do not appear in the text (see also Fig. 1). In this case, the aspect terms absorb indirectly the energy of the opinion expression "relax, be able to talk and have a very good time," followed by the opinion terms "nice quite" and "good," respectively. These examples were extracted from a real review in the datasets we used in this article.

The process of identifying explicit or implicit aspects requires a good understanding of the problem to efficiently segregate these terms. Toward this goal, a number of different approaches have been proposed. Some target coextracting aspects terms with opinion terms. These approaches are essentially applying aspect-based sentiment analysis (ABSA) [2], [3], [4]. On the other hand, there are approaches that target extracting only the aspect terms. This is known as aspect terms extraction (ATE) [5], [6], [7], [8] and it is where our method belongs to. Both tasks are commonly addressed by two types of encoding: 1) sentence level; and 2) word level.

In *sentence-level encoding*, approaches are commonly based on recurrent [9], [10], recursive [11], [12], [13], or convolution neural networks [14], [15]. More specifically, in recurrent neural network (NN)-based approaches, a sequence of input tokens is utilized to produce a language model. In recursive-based models, parsing algorithms and sentence compositionality functions [16] are used to model a sentence. Finally, convolutional-based methods depend on deep-layer variants to produce a sentence representation. Independently of the target task (ABSA or ATE), the encoding methods focus on encoding the relation of the sentence terms to the opinion/aspect terms of a sentence.

Manuscript received 12 May 2023; revised 8 March 2024 and 15 April 2024; accepted 16 April 2024. Date of publication 14 May 2024; date of current version 2 October 2024. This work was supported by the European Union's Horizon 2020 research and innovation program under Grant Agreement 101017558 (ALAMEDA). *(Corresponding author: Pantelis Agathangelou.)*

Pantelis Agathangelou and Ioannis Katakis are with the Department of Computer Science, School of Sciences and Engineering, University of Nicosia, Nicosia CY-2417, Cyprus (e-mail: agathangelou.p@live.unic.ac.cy; katakis.i@unic.ac.cy).

Panagiotis Kasnesis is with the Department of Electronics Engineering, University of West Attica, 12243 Athens, Greece (e-mail: pkasnesis@uniwa.gr).

Digital Object Identifier 10.1109/TCSS.2024.3391833

The second group of methods (*word-level encoding*), takes advantage of attention NNs [7], [17], [18].

These methods primarily focus on the relations of the opinion/aspect terms to the rest of the terms in a sentence. This is a quality that if exploited appropriately may also aid in the exploration of additional associations between aspect terms and other sentence or opinion terms in a sentence that indicate implicit features [19].

Our method exploits properties of both encoding approaches: the sentence-level approach to exploit nearby dependencies and efficiently mine direct relations, and the word-level approach for the exploitation of distant, indirect relations. As discussed above, the sentence-level and the word-level approach apply ABSA or ATE utilizing a different type of encoding. As regards the sentence-level, underlying methods tend to encode sentence terms and their relation with nearby opinion/aspect terms. This process consequently leads to explicit opinion/ATE. On the other hand, the word-level approach even though it explores short and distant relations between sentence and opinion/aspect terms, in practice, it is tailored to extract word-to-word associations. However, this property alone cannot extract strong nearby opinion/aspect terms or other sentence terms dependencies. Considering the aspects that each encoding provides, we propose two innovative encoders, one word level and another sentence level, and integrate them into a machine translation model for implicit and explicit ATE.

A. Contributions

This article makes the following contributions.

1) It proposes a novel method that extracts implicit and explicit aspect terms from user reviews.
2) It introduces a sentence-level encoder and a language fusion encoding function that better grasps sentence compositionality.
3) It introduces an innovative word-level encoder that combines the *absolute-by-relative* with the *absolute-by-absolute* position encoding to overlap the sentence-level encoder, and further explore aspect-to-sentence terms relations.
4) Our method is compared against *ten* state-of-the-art methods, where it is ranked second with respect to the first which is a far more computationally demanding approach in terms of memory and number of parameters. It, also, outperforms the best overall approach in one out of four datasets in absolute values.
5) It outperforms all competitive methods in the datasets that are rich in expressions with implicit aspects and overall presents a very competitive performance in all datasets.

B. Definitions

To aid the reader in following our discussion, we provide the definitions as follows.

1) *Aspect*: A feature of an entity or an entity on which people have expressed their opinion about. It can be a word or phrase, i.e., "restaurant, Mp3 player."
2) *Aspect category*: A unique predefined category that aspects belong to in a particular domain, i.e., *price, service,* and *food quality*. Explicit or implicit aspects may be assigned to an aspect category.
3) *Explicit aspect*: Aspects that appear in the text and are expressively commented on by opinion holders. For example, in the statement "...I would recommend reservations on weekends though.," the term "reservations" is an explicit aspect because it is expressively commented by the opinion word "recommend."
4) *Implicit aspect*: Aspects that do not appear in the text, but are implied by the contexts in the text (e.g., in ". . . great place to go for a drink too because they have 100 kinds of beer," the aspect terms "drink" and "kinds of beer" refer to the implicit aspect "food," which does not appear in the text but is implied).

II. Related Work

A. Aspects Extraction

Early research in aspect-level opinion mining has identified the importance of opinion words or expressions that carry sentiment as good identifiers for aspects not mentioned in the context but implied. Following this observation, Su et al. [20] utilize pointwise mutual information and a score function to map opinion words to predefined manually annotated implicit aspects. In another work [21], multigrain topic models, a combination of a latent Dirichlet allocation and a probabilistic latent semantic analysis model, are used to extract aspects, which are next clustered around coherent topics. For example, *waitress* and *bartender* are part of the same topic (*staff*) for restaurants. In a joint aspect-sentiment topic model [22], the authors extract aspects and aspect-dependent sentiment lexicons. Apart from identifying the association between directly related opinion words and aspects in users' reviews, they also validated the usefulness of their method to infer the implicit aspects from specific cases of opinion words. The above methods [20], [21], [22] although presenting moderate results in extraction performance, they identify the relation of aspects with opinion words and phrases and then the relation of the latter with implicit aspects. One such relation that serves as motivation for our approach is illustrated in the sentence of Fig. 1. In this example, direct relations connect explicit aspects with aspect-dependent opinion words and indirect relations connect these aspects with an opinion modifier expression which implies an implicit aspect.

Initially the methods that studied ATE introduced part-of-speech tagging and syntactic parsing rules [23], [24] to extract nouns and noun phrases that would become candidate aspect terms. Despite their decent accuracy, these manually crafted patterns were unable to generalize successfully on aspects that did not appear in the datasets and were mainly crafted for explicit aspects extraction. Detecting implicit aspects on the other hand is challenging as they are not mentioned but are only implied by using short or long expressions. Pattern recognition and text analysis studies that researched implicit aspect extraction identify the importance of associations between context

words and entities in a clause [25], [26], [27]. Our strategy aligns with this approach as we utilize one encoder to grasp direct relations and another to identify distant indirect relations between aspects, and opinion phrases in a clause that indicate implicit aspects.

B. Sentence-Level Approaches

Upon the advent of neural networks, architectures such as convolutional [14], [15], [28], recurrent [9], [10], [29], and recursive NNs [11] were also used for the task, while also other alternatives such as combined long short-term memory (LSTM) with conditional random fields (CRFs) [30] and recursive models with CRFs [12], [13]. More specifically in [14], Hyun et al. introduce the target-level sentiment analysis task to extract sentiment polarity from targets (aspects). The method leverages distance information between a target and its neighboring words to improve classification performance. In [15], a two-step CNN mainly relies on [31] and [28] distance features to apply the semantic role labeling task. In [9], several RNN variants and CRFs are evaluated on the ATE task. They demonstrate that fine-tuning word vectors combined with linguistic features in RNN models, while also using word vectors as features in linear CRFs models can further improve the extraction performance. Our method partially aligns with this approach as we fine-tune features to feed a CRF for the ATE task. In [10], two target-dependent LSTM models integrate the connections between an entity (or target word) with the context words to produce patterns tailored for the target-dependent sentiment classification task. Similarly, in [11] an adaptive recursive neural network exploits semantic composition functions to propagate the sentiments of words to aspects, while in [13] a bidirectional recursive neural network explores the semantic association of words in two paths, and a CRF exploits the respective linear associations for the ATE task. In [29], several RNN variants and CRF models are investigated for the slot filling task. In [30], a combination of a Bi-LSTM with a CRF model in several sequence labeling tasks reveals that CRFs can extract linear connections among the sequence of elements and further improve extraction performance.

C. Word-Level Approaches

Recent advances in neural networks [7], [17], [18], [32] have successfully tackled many of the limitations of the above neural implementations, such as grasping long-range dependencies, the ability to distinguish parts of a sentence that are relevant to aspect extraction and parsing errors because of the informal language in opinion reviews. Attention-based models can grasp long-range dependencies and can encode time-dependent relationships in natural language. This set of features that the attention mechanism provides, we identify in this work as word-level approach. The remarkable features of the attention mechanism introduced in machine translation [33], comprises a seminal work that paved the way to exploit this architecture for other natural language processing (NLP) tasks. This set of methods motivated the use of encoding or decoding to develop language models such as GPT [34] and BERT [35]. These models however require significant computational and data resources to be effective.

D. Pretrain Language Model Approaches

In comparison with sentence-level and word-level approaches as discussed above, recently, several methods rely on the pretrained language model BERT [35] for encoding. These methods, however, are favored by the advantages of the context-based pretrained word representations over the regular pretrained word vector representations [36], [37], [38]. Some state-of-the-art methods that have employed BERT encoding for ATE include the fine-tuned BERT [39], the post-training model in BERT-PT [39], the adversarial training model in BAT [40], the contrastive learning over self-augmented data method [41], and the PH-SUM [42]. The advantages of BERT encoding have also been successfully applied in various NLP problems, related to ATE such as name entity recognition, semantic role labeling, and topic modeling. One such method that leverages BERT encoding for text summarization via a topic-aware model is introduced in [43].

In this work, we employ some of the above-mentioned methods as discussed in Sections II-B and II-C and innovate in the following ways: first, we employed the encoder–decoder scheme presented in [33]. This structure has presented a state-of-the-art performance in translation-based tasks, which are considered of higher complexity in comparison with ATE. In our setting, we have combined two innovative encoders in the above-mentioned architecture: the word-level encoder and the sentence-level encoder. Both are tailored to improve the aspect extraction performance, focusing specifically on the extraction of implicit aspects which may extend from single to multitoken terms. Then, a decoder combines and disassembles the patterns that have been extracted by the two encoders, producing thus a composite sentence representation which is forwarded next for further processing. On top of all layers, a CRF exploits the sequential relations from the network output.

III. The Proposed Model

A. Problem Formulation

A common problem formulation for the task of aspect extraction is that of sequence labeling. In this setting, every term in the input sequence is aligned with one label in the output sequence. The most common scheme that is used for this case is the $B/I/O$ sequence. In this scheme, B stands for "begin aspect," I for "inside aspect," and O for "outside of aspect." In this work, we also adopt the $B/I/O$ scheme and take it into account for model development and evaluation.

B. Overview

Our model word–sentence, namely WoSe, is organized into four blocks (see Fig. 2). The first and the second blocks are encoders, the third is a decoder, and the last block integrates a prediction layer and a CRF. The encoders are the word-level and the sentence-level encoders. The decoder receives the output of

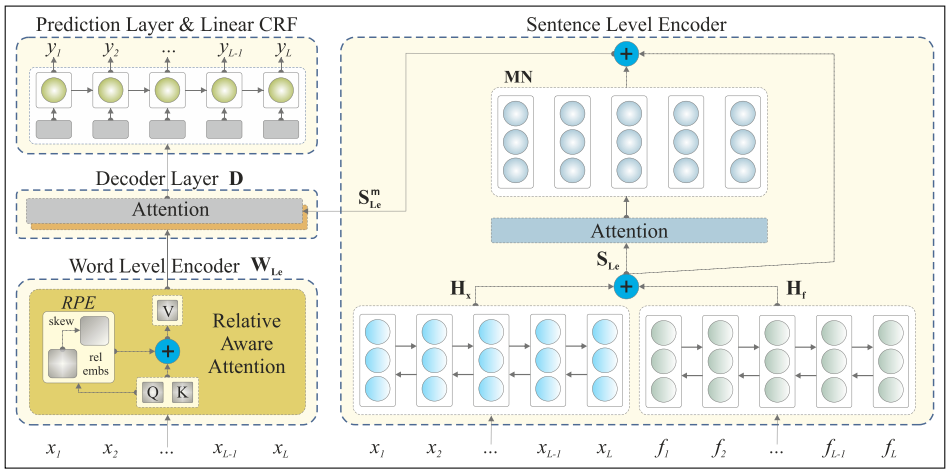

Fig. 2. Our model's high-level architecture.

the first and the second block. Its purpose is to combine and disassemble the explored patterns that exist in the word- and sentence-level encoders with respect to the optimization task (aspect extraction). In the fourth block, we introduce a prediction layer and a CRF. The prediction layer produces the network score. It is a linear projection of the decoder's output, which next is forwarded to a linear CRF. This final layer explores and identifies nearby dependencies with respect to the optimization task. In the following sections, we provide details about these blocks.

C. Input Feed

The input in our method is a sequence of token indices $X = [x_1, x_2, \cdots, x_{L-1}, x_L]$ and POS-tag indices $F = [f_1, f_2, \cdots, f_{L-1}, f_L]$, where L is the sentence length. These indices are converted into a sequence of dense word vectors or embeddings I_{we} with size d^{we} and a sequence of POS-tag embeddings I_{te} with size d^{te} [44]. These sequences of embeddings (words and POS-tags) are concatenated to formulate a sequence of vectors. These sequences are $I_{\text{we}} = [e_1; e_2; \cdots; e_{L-1}; e_L]$, with size $L \times d^{\text{we}}$ and $I_{te} = [t_1; t_2; \cdots; t_{L-1}, t_L]$ with size $L \times d^{te}$. Next, the word- and sentence-level encoders employ either the word, or the word and the POS-tag sequence of embeddings as input features for their processing.

D. Word-Level Encoder

This part receives a sequence of word vectors I_{we} as input ($L \times d^{\text{we}}$) and produces an encoder of similar size, exploiting a relative aware attention.

1) Relative-Aware Attention Layer: An attention layer is a computation mechanism that explores the most significant elements in a sequence of vectors. This results in an output sequence of weighted average vectors that define the inner-semantic vector's association with respect to a specific task.

In this process, the sequence of word vectors I_{we} is first projected onto three linear and learnable sequences of vectors. The queries $Q = I_{\text{we}} \times W^q$, the keys $K = I_{\text{we}} \times W^k$, and the values $V = I_{\text{we}} \times W^v$ with size $L \times d^{\text{we}}$, respectively. The matrices W^q, W^k, W^v with size $d^{\text{we}} \times d^{\text{we}}$ are parameters that are fine-tuned during the optimization process. In a regular attention layer (1), queries and keys interact and produce an inner-semantic association among the elements of the input feed I_{we}

$$\text{Att}(I_{\text{we}}) = \text{softmax}\left(\frac{Q \times K^T}{\sqrt{d^{\text{we}}}}\right) \times V. \tag{1}$$

For example, let us define a matrix $S(i_q, j_k)$ with size $L \times L$ that embeds this semantic association, after the $Q \times K^T$ computation in (1). Assume a matrix value with index $S(i_q = 1, j_k = 2)$. This, defines the semantic association between the word vector $I_{\text{we}}\{e_i = 1\}$ and the word vector $I_{\text{we}}\{e_i = 2\}$. The greater this value, the greater the inner-vectors' semantic association. Then, this matrix interacts with the values V and finally produces the sequence of weighted average vectors, the output of an attention layer. Note that, $\sqrt{d^{\text{we}}}$ stands for the scaling parameter and is important for keeping the inner-vectors semantic scores low.

The above implementation, however, suffers from a significant weakness. It fails to identify the ordering of the elements in a sequence. Consequently, it cannot explore complex sequential patterns that exist in a task such as aspect extraction. The most common remedy for sequence interaction in an attention layer, includes either absolute positional encoding [33] or relative positional encoding [45]. In our setting, we have employed the relative positional encoding presented in [46]. This implementation fits nicely with our method, because of two key attributes. The first relates to the reduction of the computation complexity and the second with the inner elements association that this relative encoding produces. It applies a transformation process named "skewing," which transforms an "absolute-by-relative" elements association into "absolute-by-absolute" elements association [46]. The latter attribute of this relative position encoding aligns nicely with the attributes of

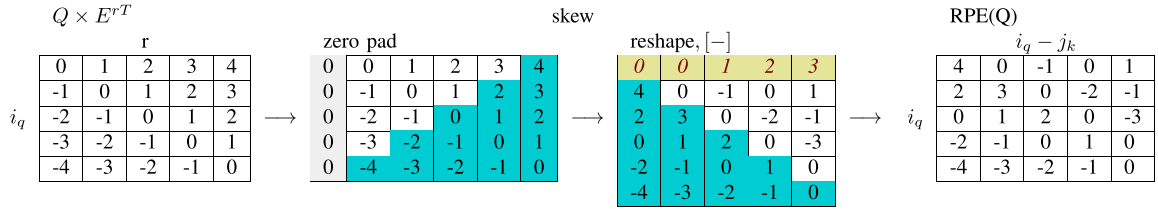

Fig. 3. Example of the skewing process in the relative position encoding function $RPE(Q)$. Values in the matrix $Q \times E^{rT}$ represent the relative distance of input elements. Sample assumes an input of five elements length. Rows indexes i_q represent the elements, column indexes r, are the relative elements distance.

the encoder we introduce next, the sentence-level encoder. The relative positional encoding is calculated as

$$\text{RPE}(Q) = \text{skew}(Q \times E^{rT}) \tag{2}$$

Fig. 3 explains the skewing process. Then, the relative-aware attention mechanism is implemented after combining (1) and (2) into the following form:

$$W_{\text{Le}} = \text{softmax}\left(\frac{Q \times K^T + \text{RPE}(Q)}{\sqrt{d^{\text{we}}}}\right) \times V \tag{3}$$

where E^r is the relative embedding's matrix with parameters $L \times d^{\text{we}}$. W_{Le} is the result of the relative-aware attention layer, which in our method is the word-level encoder W_{Le}.

In this block, the word-level encoder W_{Le} extracts complex pairwise semantic associations among the input feed elements. However, this encoder alone cannot grasp strong nearby relations that are explored in the next block, the sentence-level encoder.

E. Sentence-Level Encoder

The input features in this block are two sequences of vectors. The first, is the sequence of words, the embeddings I_{we} and the second, the sequence of the POS-tag embeddings I_{te}. The first data transformation in this block are implemented via the exploitation of two bidirectional recurrent neural network (Bi-RNN) cells. One exploits the sequence of word embeddings and the other the sequence of the POS-tag embeddings; these produce a Bi-RNN for each of the sequence of embeddings, the word and the POS-tag. We use the notation H_x for the word and H_f for the POS-tag's Bi-RNNs, respectively (see Fig. 2).

In our method, we assume that each element in the Bi-RNNs H_x, H_f encompasses the property $p(x^t|x_1^{t-1})$ of a language model. Since each Bi-RNN consists of two paths, a forward and a backward, for each set of input features, the sequence of elements in these networks embed a forward and a backward language propagation. The sequence of the forward word vector language propagation is implemented as

$$H_x^f = [p(x_1); p(x_2|x_1); \cdots ; p(x_L|x_1^{L-1})] \tag{4}$$

and the backward as

$$H_x^b = [p(x_1|x_2^L); \cdots ; p(x_{L-1}|x_L); p(x_L)] \tag{5}$$

respectively. For presentation purposes, instead of the word embeddings e, we use the notation x of the words or token indices in the elements of the H_x^f, H_x^b language propagation in (4) and (5), respectively. Next, we concatenate these two propagations to produce the representation of the first language model H_x as

$$H_x = [H_x^f; H_x^b]. \tag{6}$$

This network H_x (6) has size $L \times h$, where h is the network depth.

The second forward H_f^f and backward H_f^b language propagations of the POS-tag embeddings are as follows:

$$H_f^f = [p(f_1); p(f_2|f_1); \cdots ; p(f_L|f_1^{L-1})] \tag{7}$$

and the backward as

$$H_f^b = [p(f_1|f_2^L); \cdots ; p(f_{L-1}|f_L); p(f_L)]. \tag{8}$$

Similarly as above, instead of the POS-tag embeddings t, we use the notation f of the POS-tag indices in the elements of the H_f^f, H_f^b language propagations. Next, we concatenate them to provide the representation of the second language model H_f with size $L \times h$ as

$$H_f = [H_f^f; H_f^b]. \tag{9}$$

Now, each element in the language models H_x, H_f [(6) and (9)] encompasses a different set of characteristics. The elements of the H_x language model can predict either the next or the previous word sequences. Similarly, the elements of the H_f language model can predict the next POS-tag or the previous POS-tag sequences. The first H_x language model relies on a word-level and the second H_f on an ensemble of words (i.e., a number of different words may be assigned with the same POS-tag). We add these models (the word-level language model and the ensemble-level POS-tag language model) together element wise, to produce a neural sentence representation, that we name *language model fusion* S_{L_e}. In other words, we have the following:

$$S_{L_e} = H_x + H_f. \tag{10}$$

1) Memory Network (MN): Feature augmented models [28], [47], [48] are usually favored by the parsing process and perform better generalization in comparison with single models that rely solely on word embeddings. However, this improvement is strongly related to the quality of this parsing process. Parsing errors may lead to a loss of generalization or model overfit. Moreover, encoding information about which content words are more important in relation to mentioned aspects, favors the generalization improvement [5], [18]. We alleviate

these issues introducing a MN over the sentence representation S_{Le} (10) introduced earlier. This MN which is a single attention layer (1), filters parsing errors, while also identifies the most important patterns of the fused language model S_{Le}. The memory augmented sentence-level encoder that we obtain after the MN is calculated as

$$S^m_{\text{Le}} = S_{\text{Le}} + \text{Att}(S_{\text{Le}}). \tag{11}$$

This block, with the sentence-level encoder S^m_{Le} (11), extracts complex sequence language patterns among the grammatical elements of a sentence. Its role aligns and overlaps with the previously introduced word-level encoder and both provide the necessary patterns for the next disassembling layer, the decoder.

F. The Decoder Layer

For applying the decoding process in our method, we employ the decoding mechanism presented in [33]. This consists of two attention layers. The first one receives the sentence-level encoder S^m_{Le}, uses the computations of (1) and produces an output we name augmented sentence-level encoder $\text{Att}(S^m_{\text{Le}})$. The second attention layer receives two inputs: 1) the augmented sentence-level encoder $\text{Att}(S^m_{\text{Le}})$; and 2) the word-level encoder W_{Le}, and produces the decoding output. This latter attention layer is the most significant as it applies the disassembling process. It uses the computations of (1) and produces an output matrix D with size $L \times h$. We name this matrix the decoder's output. In other words: $Q = \text{Att}(S^m_{\text{Le}}) \times W^d_q, \quad K = W_{\text{Le}} \times W^d_k, \quad V = W_{\text{Le}} \times W^d_v$, where W^d_q, W^d_k, W^d_v are learnable parameters of size $h \times h$. This process aims at improving sentence compositionality. In the experimental setup, we demonstrate its effectiveness against other benchmark methods in the evaluation of explicit and implicit aspects. Finally, the output of this block, the matrix D is sent to the final block that consists of the prediction layer and the CRF.

G. Prediction Layer & CRF

This is the final block of our model, it consists of two layers. The prediction layer or network score and a linear CRF layer.

1) Network Score: The network score N^s is produced after the linear projection of the decoder's output, the matrix D. The calculation of this linear projection is being done by

$$N^s = D \times W^s + b^s \tag{12}$$

where W^s is a matrix of $h \times \Delta$ learning parameters, b^s is the respective bias with size Δ, and N^s the network score (12), with size $L \times \Delta$. The vector Δ includes the labels of the $B/I/O$ scheme.

2) CRF: The final layer encompasses a CRF [49]. A CRF considers the correlations between tags in neighborhoods and scores the whole sequence of tags. Thus, a linear-chain CRF calculates the conditional probability of the tag sequences as follows:

$$p(y|N^s) = \frac{\exp(s(N^s, y))}{\Sigma_{y' \in y} \exp(s(N^s, y'))} \tag{13}$$

where N^s, y is the set of all input and tag sequences and $s(N^s, y)$ is the score function, which is implemented in our setting as follows:

$$s(N^s, y) = \Sigma^L_i (A_{y_{i-1}, y_i} + N^s_{i, y_i}). \tag{14}$$

Quantity A_{y_{i-1}, y_i}, in (14) measures the transition score and N^s_{i,y_i} measures the network score among the tag sequences. We train our model using the negative log likelihood of $p(y|N^s)$ as the loss in a sentence: The model also employs the Viterbi algorithm for decoding the tag sequence, maximizing the conditional probability $p(y|N^s)$ of (13). Then, the model calculates the prediction cost as follows: $J(\theta) = \sum_s^{|m_b|} L(s)$, where m_b is the mini-batch size. Finally, the error $J(\theta)$ is minimized by an optimization algorithm that implements the training phase of the model.

IV. Experimental Setup

In this section, we provide details about the tuning process of the hyperparameters of our method, word–sentence-level representations for implicit aspect extraction (WoSe), as well as information related to the datasets and the alternative methods we utilize for the experimental section.

A. Hyperparameter Tuning and Training

In WoSe, we initiate the word vectors representations utilizing the fastText,[1] pretrained vectors [50], [51], which we keep static during the training of the model. The POS-tag representations were initialized as one-hot vectors with 46 dimensions and remained static during training. For parsing each sentence and producing the POS-tagging labels, we used the NLTK parser. All other model parameters were tuned during training using the Adam optimization algorithm [52] with a learning rate of 3×10^{-3} and exponential decay during the training process. Additionally, model parameters were regularized with a per-minibatch L_1, L_2 regularization strength of 6×10^{-3} and 10^{-3}, respectively. One attention layer was used for each encoding and two for the decoding block. All attention layers were supported by a dropout, batch normalization layers and a residual network connection [53]. The latter in the decoder was also equipped with the masking property. The depth representation h of the attentions, the Bi-RNNs, and the word vectors d^{we} was set to 300 dimensions. When ELMo [54] contextualized word embeddings were used by our method, this dimension was set to 1024. In this case the embeddings were fine-tuned during training. The learning rate was also set to 1.25×10^{-3}. The mini-batch size was set to 32 and 0.5 for the dropout rate. All datasets were split in train/test parts following the SEMEVAL challenge splits as shown in Table I. For every training dataset 10% was reserved randomly for validation. Each experiment was iterated five times in a k-fold setting and the evaluation metric $F_1 \sim f(p, r)$ score, which is calculated on the exact word spans between predicted and ground truth annotated aspects, was based on the respective average on the test sets. After every iteration, the model's parameters were fine-tuned from

[1] https://fasttext.cc/docs/en/english-vectors.html

TABLE I
BENCHMARK DATASETS AND STATISTICS

Dataset		L^{max}	Sent.	S. w. Asp.[a] (s)	Asp. (a)
lap14	Train	83	3045	1488	2358
	Test		800	422	654
rest14	Train	79	3041	2021	3697
	Test		800	606	1133
rest15	Train	68	1315	832	1192
	Test		685	402	542
rest16	Train	78	2000	1233	1743
	Test		676	429	622

[a]Sentences With Aspects.

the beginning. Finally, we set the number of training epochs to 100, while the model version that achieved the best average score on the validation set was saved and its weights were used to obtain the model's performance on the test set.

B. Preprocessing and Datasets

During the preprocess (for WoSe only) at each dataset, we converted every word to lower case and replaced any numbers with a digit pseudovariable. The input feed length was set to a maximum value, different for each dataset (see Table I). For alleviating sentences of varying length, we padded with zeros the remaining length up to the maximum value. Delimiters and special characters were not removed, as many of those were part of the ground-truth aspects in some cases. To evaluate the effectiveness of our method for the ATE task, we conducted experiments on four benchmark datasets from the SEMEVAL ABSA (2014, 2015, and 2016) challenge. Table I presents these benchmark datasets and statistics.

Dataset lap14 contains reviews of laptops (Semeval 2014), rest14, rest15, rest16 are about restaurants (Semeval 2014, 2015, 2016).

Reproducibility Note: All source codes that is required to run the following experiments are available at: https://github.com/unic-ailab/wose.

C. Competitive Methods

To evaluate our method, we employed the following benchmark methods for comparison.

1) *cLSTM*: The concatenated context vector LSTM implementation introduced in [9]. In this method, a Bi-RNN (LSTM) exploits pretrained word vectors & contextual features to grasp long-range aspect-content terms dependencies. This baseline shares some attributes with the sentence-level encoder we introduced in Section III-C. However, this method alone encodes the content and cannot explore successfully distant direct and indirect relations, i.e., opinion terms and phrases that exist in the same message and relate to aspects.
2) *RNCRF*: A recursive neural network with high-level representations for a CRF and opinion/aspect terms coextraction presented in [12]. This method utilizes a dependency-tree structure to encode opinion-aspect terms relations and produce high-level feature representations for each word in a context. On top of these representations, a CRF grasps features connections and extracts opinion/aspect terms. Our approach also builds features representations for each word in a context and a CRF for the linear association of these features with the aspect terms. However, we demonstrate that by combining a set of different encodings, in our case the word and the sentence-level encoder, more fine-grained features for the ATE task can be produced.
3) *CMLA*: A multilayer coupled attention architecture for opinion/aspect terms coextraction presented in [17]. This method introduces a tensor operation and a couple of attention mechanisms that interactively learn information between opinion and aspect terms relations. Different from this approach, our method does not use opinion terms prototypes for aspect associations however, via its architecture properties, it is able to explore direct and indirect relations of the aspect terms with the rest of the terms in the sentence.
4) *HAST*: A neural network with history attention and selective transformation for opinion/aspect terms coextraction presented in [7].

   This framework exploits two key features in the attention mechanisms that it introduces. The history attention reduces the error space for each aspect prediction and the selective transformation exploits the opinion summary to strengthen the correlations between opinion and aspect terms in a sentence. In comparison, our method is more holistic in the exploration of aspect-content term relations. The sentence-level encoder, is tailored to grasp direct relations, while a second the word-level overlaps the first in the distant indirect relations. In the ablation study, we present a weakness of HAST method, which is the poor performance to extract multitoken aspect dependencies, which, in contrast, our method tackles successfully.
5) *DE-CNN*: A double embeddings (generic, domain) convolutional neural network for ATE presented in [48]. This method utilizes two types of embeddings to encode opinion-aspect terms relations for the ATE task: domain embeddings and general embeddings. The domain embeddings are produced as follows. First, the general fastText [51] embeddings are collected which are next fine-tuned on a related to the benchmark dataset corpus. For the general type of embeddings, the authors utilize the pretrained glove [37] dense word vectors. Despite the state-of-the-art performance in some benchmark datasets, this method is tailored for explicit aspect term extraction. Moreover, in comparison with our method we rely on a single type of general embeddings to grasp explicit and implicit aspect relations.
6) *BERT-PT*: A BERT post-training model for review reading comprehension and ABSA, as presented in [39]. This method relies on a regular BERT base pretrained encoder that is twice fine-tuned. First, on a domain knowledge dataset, and next on a post-training process to augment

TABLE II
SUMMARY OF CHARACTERISTICS OF THE COMPARING METHODS WITH THE PROPOSED WOSE

Method	Ref.	Approach
cLSTM	[9]	RNN
RNCRF	[12]	Recursive + CRF
CMLA	[17]	Attention
HAST	[7]	Attention + RNN
DE-CNN	[8], [48]	CNN
BERT-PT	[39]	BERT
BAT	[40]	BERT
PH-SUM	[42]	BERT
BARTABSA	[55]	BART
WoSE	Ours	RNN+Attention+ CRF

TABLE III
ABLATION STUDY NOTATIONS & MEANING

Notation	Meaning
w/o:	Without a specific part of our method
abs:	Absolute position encoding
rel:	Relative position encoding
soft-max:	Softmax prediction layer
crf:	Conditional Random Field prediction layer
MN:	Memory Network
POS:	Part Of Speech tagging process
WL:	Word-Level encoder
SL:	Sentence-Level encoder
WoSe:	Word & Sentence-Level encoder
ELMo:	The ELMo contextualized word embeddings
skew:	The relative position encoding of (2)

the ATE performance. Our method although uses fewer parameters it demonstrates its effectiveness against this fine-tuned pretrained language model.

7) *BERT*: We include evaluation results for BERT fine-tuned for ATE [39].
8) *BAT*: An adversarial training model for aspect-based sentiment analysis with BERT [40]. This method introduces an adversarial process that acts as a regularization procedure in the embedding space of a general BERT and a BERT-PT [39] pretrained model to produce more robust results for ATE. In the ablation study, we show some of the advantages and disadvantages of this method with respect to WoSe.
9) *PH-SUM* [42] a BERT-based model that employs two simple modules called parallel aggregation and hierarchical aggregation to be utilized on top of BERT for two main ABSA tasks namely ATE.
10) *BARTABSA* [55] a BART-based model that solves all ABSA subtasks in an end-to-end framework.

Also, Table II presents the summary of characteristics of the comparing methods with the proposed WoSe. We present the approach that each method employed for evaluation of their model. In comparison with other methods, WoSe is a synthetic method that employs RNN, attention, and a CRF features.

V. RESULTS & DISCUSSION

In this section, we evaluate our model on the task of aspect extraction. All experimental results refer to the F_1 score metric which was calculated based on the precision and recall values of the exact matching between predicted and annotated aspects and was measured on the test sets after training the models.

A. Ablation Study

First, we study the alternative implementations of WoSe (see Table IV). We use many variations to study the value and contribution of each component independently. We use the notations of Table III for the variations of WoSe which are constructed with the elements we describe in Sections III-B and III-C and a final prediction layer.

In the experimental results of Table IV, we notice that $\text{WL}^{\text{abs}}_{\text{soft-max}}$ produced the worst results. This alternative did not operate successfully in this setting as it could not generate coherent predictions in all cases. Next, we have $\text{WL}^{\text{rel}}_{\text{soft-max}}$, this model produced significantly higher scores compared to $\text{WL}^{\text{abs}}_{\text{soft-max}}$ and it proves that it can operate independently for ATE. The $\text{SL}_{\text{soft-max}}$, performed better compared to $\text{WL}^{\text{abs}}_{\text{soft-max}}$ and $\text{WL}^{\text{rel}}_{\text{soft-max}}$. This indicates that the sentence-level encoder provides the greatest contribution in the overall performance of our method. On the other hand, $\text{WoSe}^{\text{abs}}_{\text{soft-max}}$ and $\text{WoSe}^{\text{rel}}_{\text{soft-max}}$ which are constructed by combining the WL and the SL encoders performed better from the basic approaches that use either WL or SL in most cases except from the rest15 or rest16 datasets, where the $\text{SL}_{\text{soft-max}}$ presented slightly higher scores. However, the standard deviations ($s.d$) in $\text{WoSe}^{\text{abs}}_{\text{soft-max}}$ and $\text{WoSe}^{\text{rel}}_{\text{soft-max}}$ models was significantly higher. This shows that these WoSe models produced higher to $\text{SL}_{\text{soft-max}}$ scores. Consequently, we may safely infer that the WL and the SL encoders target different aspects and these different aspects utilized via the decoder layer which results in the improvement of the overall generalization. This observation is also confirmed by the rest of our experiments.

$\text{WoSe}^{\text{abs}}_{\text{soft-max}}$ produced inferior generalization compared to $\text{WoSe}^{\text{abs}}_{\text{crf}}$. This implementation, however, outperformed $\text{WoSe}^{\text{abs w/o } MN}_{\text{soft-max}}$ in all cases with performance benefits 0.77%, 0.27%, 0.04%, and 0.18% in the corresponding datasets. This implies that the MN favored the aspect extraction task.

TABLE IV
RESULTS FOR VARIOUS WOSE IMPLEMENTATIONS

Dataset/Benchmark	lap14 $\pm$ $s.d$	rest14 $\pm$ $s.d$	rest15 $\pm$ $s.d$	rest16 $\pm$ $s.d$
$\text{WL}^{\text{abs}}_{\text{soft-max}}$	4.28% $\pm$0.70	5.62% $\pm$0.14	6.89% $\pm$0.82	5.10% $\pm$0.26
$\text{WL}^{\text{rel}}_{\text{soft-max}}$	65.33% $\pm$0.83	76.01% $\pm$1.10	58.72% $\pm$1.71	63.39% $\pm$1.66
$\text{WoSe}^{\text{abs w/o } MN}_{\text{soft-max}}$	76.15% $\pm$0.85	84.10% $\pm$0.44	67.53% $\pm$1.07	71.16% $\pm$1.21
$\text{SL}_{\text{soft-max}}$	76.16% $\pm$0.99	84.21% $\pm$0.22	67.72% $\pm$0.76	72.54% $\pm$0.83
$\text{WoSe}^{\text{abs}}_{\text{soft-max}}$	76.92% $\pm$0.69	84.37% $\pm$0.79	67.84% $\pm$1.20	71.34% $\pm$2.15
$\text{WoSe}^{\text{rel}}_{\text{soft-max}}$	76.79% $\pm$0.79	84.04% $\pm$0.44	67.57% $\pm$1.52	71.60% $\pm$1.62
$\text{WoSe}^{\text{abs}}_{\text{crf}}$	79.36% $\pm$1.26	84.15% $\pm$0.54	69.44% $\pm$1.72	73.26% $\pm$1.05
$\text{WoSe}^{\text{rel w/o skew}}_{\text{crf}}$	73.36 % $\pm$1.05	79.97 % $\pm$0.86	63.22 % $\pm$1.85	65.77 % $\pm$1.81
$\text{WoSe}^{\text{rel w/o POS}}_{\text{crf}}$	80.55% $\pm$1.10	86.33% $\pm$0.62	**70.65%** $\pm$0.87	73.55% $\pm$1.85
$\text{WoSe}^{\text{rel}}_{\text{crf}}$	81.06% $\pm$1.16	85.64% $\pm$0.64	69.99% $\pm$0.92	73.94% $\pm$0.84
ELMo + $\text{WoSe}^{\text{rel},*}_{\text{crf}}$	**81.64%** $\pm$1.60	**88.38%** $\pm$1.19	70.60.% $\pm$1.95	**76.73%** $\pm$1.93

Note: Underline values indicate the best performance between $\text{WoSe}^{\text{rel}}_{\text{crf}}$ and $\text{WoSe}^{\text{rel w/o POS}}_{\text{crf}}$ and boldface the respective best in a dataset.

TABLE V
ALGORITHM RANKING BASED ON THE RESULTS OF TABLE IV

	lap14	rest14	rest15	rest16	avg.r
$WL^{abs}_{soft\text{-}max}$	11	11	11	11	11.0
$WL^{rel}_{soft\text{-}max}$	10	10	10	10	10.0
$WoSe^{abs\ w/o\ MN}_{soft\text{-}max}$	8	7	8	8	7.75
$SL_{soft\text{-}max}$	7	5	6	5	5.75
$WoSe^{abs}_{soft\text{-}max}$	5	4	5	7	5.25
$WoSe^{rel}_{soft\text{-}max}$	6	8	7	6	7.00
$WoSe^{abs}_{crf}$	4	6	4	4	4.50
$WoSe^{rel\ w/o\ skew}_{crf}$	9	9	9	9	9.00
$WoSe^{rel\ w/o\ POS}_{crf}$	3	3	1	3	2.50
$WoSe^{rel}_{crf}$	2	2	3	2	2.25
ELMo + $WoSe^{rel,*}_{crf}$	1	1	2	1	1.25

The $WoSe^{rel\ w/o\ skew}_{crf}$ performed the worst generalization from all other alternatives but for $WL^{abs}_{soft\text{-}max}$ and the $WL^{rel}_{soft\text{-}max}$. This indicates the contribution and adaptation of the skewing relative position encoding in the overall performance of our method. The $WoSe^{rel}_{crf}$ implementation scored the best results in lap14 and rest16 datasets, whereas the $WoSe^{rel\ w/o\ POS}_{crf}$ the best results in the rest14 and rest15 datasets. The CRF layer that was used in these models played a significant role in the generalization improvement. It revealed that linear correlation among aspect elements is critical for the task. Moreover, since our method could produce state-of-the-art results, without the POS-tagging process, it implies that WoSe can also be used in an end-to-end process without relying on external resources. Finally, ELMo + $WoSe^{rel,*}_{crf}$ is the implementation that receives the contextualized word embeddings of [54] as input feed in the model architecture. The asterisk$^{(*)}$ notation implies w/o POS for rest14, rest15 evaluations (see Table IV), where best performance obtained without the POS features. What we observe here is that WoSe performance is improved even more and outputs F_1 scores largely compared to pretrained language encoding methods. For the easy readability of the above-mentioned alternative evaluations, we also provide Table V with the ranking of each alternative evaluation.

B. Comparison Against the State-of-the-Art and Pretrained Transformers

In this section, we discuss the results of the WoSe model in comparison with other approaches that use pretrained word embeddings for their implementation and approaches that use pretrained language BERT models [35].

We observe that the proposed model $WoSe_o$ presents better F_1 scores on all datasets against all the pretrained word embeddings implementations. Note that in this case $WoSe_o$ corresponds to the best value achieved by WoSe that can be either $WoSe^{rel}_{crf}$ or $WoSe^{rel\ w/o\ POS}_{crf}$ depending on the dataset as discussed.

Table VI presents the training parameters of the two BERT models: $BERT_{BASE}$ and $BERT_{LARGE}$. These BERT implementations were employed by the methods in [39] and [40] that we discuss next. Table VI also presents the parameters of the WoSe model. Comparing WoSe's parameters with the respective of

TABLE VI
TRAINING PARAMETERS

Model	Layers	Hidden	Heads	Parameters
$BERT_{LARGE}$	24	1024	16	340M
$BERT_{BASE}$	12	768	12	110M
ELMo + WoSe	5	1024	1	40M
WoSe	5	300	1	3.5M

TABLE VII
WOSE AGAINST THE STATE OF THE ART

Dataset/Benchmark	lap14	rest14	rest15	rest16
cLSTM [9]	75.71%	82.01%	68.26%	70.35%
RNCRF [12]	78.42%	84.93%	67.74%	69.72%
CMLA [17]	77.80%	85.29%	70.73%	72.77%
HAST [7]	79.52%	85.61%	71.46%	73.61%
HAST (ours)	79.14%	85.04%	71.13%	73.92%
DE-CNN [8], [48]	81.59%	85.20%	68.28%	74.37%
BERT [39]	79.28%	-	-	74.10%
BERT (ours)	81.48%	85.32%	68.67%	75.13%
BERT-PT [39]	84.26%	-	-	77.97%
BERT-PT (ours)	84.40%	88.17%	71.02%	80.41%
BAT [40]	85.57%	-	-	81.50%
BAT (ours)	85.97%	88.15%	72.25%	81.13%
PH-SUM [42]	**86.09%**	-	-	**82.34%**
BARTABSA [55]	83.52%	87.07%	**75.48%**	-
$WoSe_o$	82.52%	87.16%	72.10%	76.20%
ELMo + $WoSe_o$	83.82%	**90.05%**	72.86%	78.83%

Note: **Bold values** indicate the highest score in each column.

$BERT_{BASE}$ and $BERT_{LARGE}$ it is safe to say that the number of parameters in BERT implementations is greater by several orders of magnitude compared to WoSe. Table VII presents the comparison of WoSe against the employed competitive and the transformer's based, the BERT, the BERT-PT, the BAT, the PH-SUM, and the BARTABSA. Similarly Table VIII presents the respective rankings of WoSe against the state of the art.[2] Where there is the notation (ours) in the results, we refer to optimal score values we obtained after applying the respective cited methods source code on the benchmark datasets we used for WoSe by our team.

Observing the results, we note the following. $WoSe_o$ outperformed all dense word vectors benchmark methods in all datasets. The ELMo + $WoSe_o$ implementation boosted the optimum performance of WoSe and provided better F_1 scores in all datasets. It worth's to note that despite our model presented inferior performance to BERT-PT, BAT, and PH-SUM at lap14, and rest16 datasets, it outperformed BERT base in all cases and BERT-PT, BAT in rest14, and rest15 datasets. This is an important outcome especially if we consider that WoSe uses a singe layer for each of the encoders W_{Le} and S^m_{Le}. This is an indication that our method managed to grasp better the associations between the aspect terms with the rest of the content terms, by using fewer sets of neural representations (see also

[2]We have included only the benchmark methods from Table VII where evaluation scores are available from all datasets.

TABLE VIII
RANKING OF WOSE AGAINST THE STATE OF THE ART

Dataset/Benchmark	lap14	rest14	rest15	rest16	*avg.r*
cLSTM [9]	11	11	10	10	10.50
RNCRF [12]	9	10	11	11	10.25
CMLA [17]	10	7	7	9	8.25
HAST [7]	7	5	4	8	6.00
HAST (ours)	8	9	5	7	7.25
DE-CNN [8], [48]	5	8	9	6	7.00
BERT (ours)	6	6	8	5	6.25
BERT-PT (ours)	2	2	6	2	3.00
BAT (ours)	1	3	2	1	1.75
WoSe_o	4	4	3	4	3.75
ELMO + WoSe_o	3	1	1	3	2.00

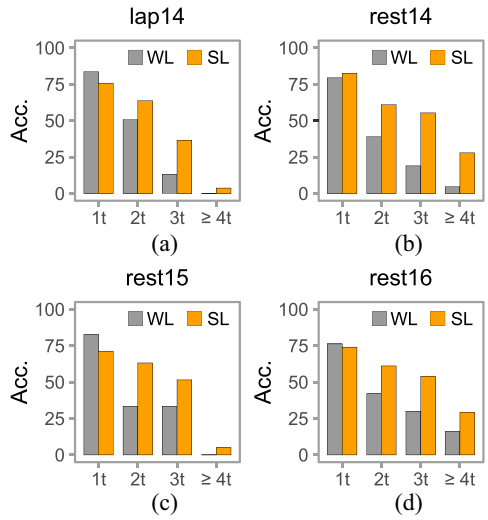

Fig. 4. Single $1t$ and multitoken $(2t, 3t, \geq 4t)$ aspect terms length prediction accuracy for the word-level (WL) and the sentence-level (SL) encoders. (a) for lap14, (b) for rest14, (c) for rest15, and (d) for rest16 benchmark dataset.

Table VI). Thus, our framework yielded a more effective and efficient model.

C. Performance Contribution of Each WoSe Element

This section aspires to shed light on the contribution of each encoding method, the word-level and the sentence-level (Sections III-B and III-C) in the performance of variable length aspect terms. Fig. 4 depicts this evaluation result on single $(1t)$ and multitoken $(2t, 3t, \geq 4t)$ aspect terms length prediction accuracy for the employed benchmark datasets.

What we observe here is that in all cases the word-level encoder's accuracy is high on single-token $1t$ aspect terms length, but drops abruptly on multitoken (i.e., two $2t$, three $3t$, and greater than four $\geq 4t$) aspect terms length. This, imprint the prediction performance of the word-level encoder, a fact which also aligns we the remark we highlighted in the affiliated section, that the word-level encoder extracts strong pairwise semantic associations among the input feed elements.

TABLE IX
IMPLICIT EXAMPLES AND ANNOTATION

1	lap14	This thing is awesome, everything always *works*, everything is always easy to *set up*, everything is compatible, its literally everything I could ask for.
2	lap14	Everything is so easy and intuitive to *setup* or *configure*.
3	lap14	The investment of a new MacBook Pro came at a *price*, but totally worth it for a good piece of mind.
4	rest14	If you want good tasting, well seasoned *Latin food* eat at Cabana and you can't go wrong.
5	rest14	How pretentious and inappropriate for MJ Grill to claim that it provides *power lunch* and *dinners*!
6	rest14	One caveat: Some of the *curried casseroles* can be a trifle harsh.
7	rest15	Excellent food, although the *interior* could use some help.
8	rest15	There is something about their *atmosphere* that makes me come back nearly every week.
9	rest15	Over all the looks of the place exceeds the actual *meals*.
10	rest16	This *place* rocks!!
11	rest16	To the owners of *Open Sesame*... Bravo... I can't wait to come back to dine at your restaurant!.
12	rest16	Another plus is the open feel of the restaurant with *glass walls* on all sides.

Note: Italic font indicate the implicit aspect terms.

As regards the sentence-level encoder, we observe that its prediction performance is more balanced along the single and the multitoken aspect term lengths and in all cases is greater than the word-level encoder on multitoken aspect terms length. It is also significant to note here that the two encoders target different aspect terms, an observation that is also verified by the experimental results we present next.

VI. EXPLICIT & IMPLICIT ASPECTS

In this section, we study the ability of our algorithm to identify explicit and implicit aspects. If an aspect is mentioned in the text (e.g., "battery life") or is directly related to an opinion term or phrase (e.g., "good design," "i like the *brightness* and *adjustments*"), we consider it and label it as explicit. Similarly, if an aspect is implied or is related to an opinion phrase indirectly (e.g., "i know real Indian food and this was not it" and "i have dropped mine a couple times with only a *slim plastic case* covering it."), is labeled as implicit. If a predicted aspect (explicit or implicit) exists in the list of annotated aspects (ground truth) it is considered as accurate prediction (regardless of its position in the sentence).

In Table IX, we provide some additional examples and a short description about the annotation process. More specifically, at the first example at Table IX, the aspects *works* and *set up* indicate service quality, at the second example *setup* and *configure* indicate user experience, at the third example aspect *price* indicate the expensive laptop, at the fourth example aspect *Latin food* indicates food quality, at the fifth example *power lunch* and *dinners* indicate poor service quality, at the sixth example the aspect *curried casseroles*, indicate poor service quality, at the seventh example aspect *interior* indicates the ambience, at the eighth example *atmosphere* indicates the ambience, at the ninth example aspect *meals* indicate service quality, at the tenth example aspect *place* indicates the restaurant, at the eleventh example *Open Sesame* indicates service or food quality, and at the twelfth example aspect *glass walls* indicate ambience. In most cases, the implicit aspects relate with an aspect category or a noun phrase that is not mentioned but is implied in the text. Overall, we followed this general annotation rule and we labeled these examples as implicit aspects.

TABLE X
SINGLE AND MULTITOKEN TERMS PREDICTION ACCURACY FOR EXPLICIT AND IMPLICIT ASPECTS ON LAP14 AND REST14 BENCHMARK DATASETS

	lap14				rest14			
Acc.	1t	2t	3t	$\geq$ 4t	1t	2t	3t	$\geq$ 4t
(%) in eval.	(55.00%)	(34.00%)	(7.00%)	(4.00%)	(72.00%)	(19.00%)	(5.00%)	(4.00%)
(%) Expl./Impl.	(77/23)	(77/23)	(76/24)	(79/21)	(85/15)	(71/29)	(55/45)	(45/55)
HAST (ours)	82.93%	74.63%	46.51%	29.63%	91.69%	68.90%	61.40%	19.51%
BERT (ours)	87.71%	85.10%	62.79%	22.22%	92.72%	75.47%	71.93%	29.27%
BERT-PT (ours)	92.26%	87.68%	62.22%	29.63%	93.09%	84.83%	77.59%	35.71%
BAT (ours)	**92.82%**	**91.04%**	64.44%	**37.04%**	93.56%	83.41%	74.14%	37.51%
WoSe	88.76%	85.05%	64.44%	33.33%	**94.49%**	80.38%	78.95%	46.51%
ELMo + WoSe	85.92%	86.26%	**70.45%**	33.33%	93.22%	**86.12%**	**78.95%**	**48.78%**

Note: Bold values indicate the highest score in each column.

TABLE XI
RANKINGS FOR SINGLE AND MULTITOKEN TERMS FOR EXPLICIT AND IMPLICIT ASPECTS ON LAP14 AND REST14 BENCHMARK DATASETS

	lap14					rest14				
Ranking	1t	2t	3t	$\geq$ 4t	avg.r	1t	2t	3t	$\geq$ 4t	avg.r
HAST (ours)	6	6	5	3	5.00	6	6	5	6	5.75
BERT (ours)	4	4	3	4	3.75	5	5	4	5	4.75
BERT-PT (ours)	2	2	4	3	2.75	4	2	2	4	3.00
BAT (ours)	1	1	2	1	1.25	2	3	3	3	2.75
WoSe	3	5	2	2	3.00	1	4	1	2	2.00
Elmo + WoSe	5	3	1	2	2.75	3	1	1	1	1.50

TABLE XII
SINGLE AND MULTITOKEN TERMS PREDICTION ACCURACY FOR EXPLICIT AND IMPLICIT ASPECTS ON REST15 AND REST16 BENCHMARK DATASETS

	rest15				rest16			
Acc.	1t	2t	3t	$\geq$ 4t	1t	2t	3t	$\geq$ 4t
(%) in eval.	(72.00%)	(16.50%)	(7.00%)	(4.00%)	(74.50%)	(17.00%)	(4.50%)	(4.00%)
(%) Expl./Impl.	(85/15)	(86/14)	(82/18)	(83/17)	(93/7)	(93/7)	(93/7)	(100/0)
HAST (ours)	77.03%	70.73%	**61.76%**	14.29%	75.14%	65.56%	58.33%	30.43%
BERT (ours)	80.17%	65.82%	41.67%	**31.82%**	81.12%	68.37%	66.67%	53.85%
BERT-PT (ours)	**85.04%**	70.93%	50.00%	27.27%	85.98%	76.00%	74.07%	**69.23%**
BAT (ours)	84.89%	71.76%	56.76%	26.09%	**86.81%**	**80.61%**	**77.78%**	61.54%
WoSe	78.55%	74.39%	60.00%	7.07%	82.68%	76.04%	73.08%	45.83%
ELMo + WoSe	74.36%	**75.00%**	50.00%	19.05%	85.81%	76.60%	67.86%	52.17%

Note: Bold values indicate the highest score in each column.

TABLE XIII
RANKINGS FOR SINGLE AND MULTITOKEN TERMS FOR EXPLICIT AND IMPLICIT ASPECTS ON rest15 AND rest16 BENCHMARK DATASETS

	rest15					rest16				
Ranking	1t	2t	3t	$\geq$ 4t	avg.r	1t	2t	3t	$\geq$ 4t	avg.r
HAST (ours)	5	5	1	5	4.00	6	6	6	6	6.00
BERT (ours)	3	6	5	1	3.75	5	5	5	3	4.50
BERT-PT (ours)	1	4	4	2	2.75	2	4	2	1	2.25
BAT (ours)	2	3	3	3	2.75	1	1	1	2	1.25
WoSe	4	2	2	6	3.50	4	3	3	5	3.75
Elmo + WoSe	6	1	4	4	3.75	3	2	4	4	3.25

Tables X and XII present statistics of this annotation process in the evaluation datasets, while Tables XI and XIII present the respective ranking statistics. We introduce the proportion of explicit/implicit aspects per token length, the respective proportion in a dataset and the prediction performance of some selected benchmark methods (e.g., in lap14 and one token length $1t$, 55% of the evaluation dataset are one token length aspects from which 77% are explicit and 23% implicit).

TABLE XIV
SENTENCE EXAMPLE WITH THE GROUND-TRUTH ASPECTS, HIGHLIGHTED AS ***Explicit***$^{[e]}$, ***Implicit***$^{[i]}$ AND THE RESPECTIVE PREDICTIONS FROM SOME COMPETITIVE METHODS

Sentence	[1] ***Dishes***$^{[e]}$ denoted as [2] ***"Roy's Classics"***$^{[i]}$ (marked on the [3] ***menu***$^{[e]}$ with asterisks) are tried-and-true [4] ***recipes***$^{[e]}$, such as [5] ***macadamia-crusted mahi mahi***$^{[i]}$, or subtly [6] ***sweet honey-mustard beef short ribs***$^{[i]}$.
HAST	***Dishes*** denoted as ***"Roy's Classics"*** (marked on the ***menu*** with asterisks) are tried-and-true ***recipes***, such as ***macadamia-crusted mahi mahi***, or subtly ***sweet honey-mustard beef short ribs***.
BAT	***Dishes*** denoted as ***"Roy's Classics"*** (marked on the ***menu*** with asterisks) are tried-and-true recipes, such as ***macadamia-crusted mahi mahi***, or subtly ***sweet honey-mustard beef short ribs***.
ELMo + Wose	***Dishes*** denoted as ***"Roy's Classics"*** (marked on the ***menu*** with asterisks) are tried-and-true recipes, such as ***macadamia-crusted mahi mahi***, or ***subtly sweet honey-mustard beef short ribs.***
WoSe	***Dishes*** denoted as ***"Roy's Classics"*** (marked on the ***menu*** with asterisks) are tried-and-true ***recipes***, such as ***macadamia-crusted mahi mahi***, or subtly ***sweet honey-mustard beef short ribs***.

Note: Underline denotes the predicted aspect.

Now, we discuss the performance of the methods that is depicted in Tables X and XII. We can make the following observations. The BAT benchmark method out-performed WoSe at single and multitoken terms accuracy in most cases, $(6/8)$ at lap14, and rest16 benchmark datasets. However, this performance relation is reversed at rest14 and rest15 benchmark datasets, where we out-performed BAT in $(6/8)$ cases. As regards the rest benchmark methods, WoSe performed best in most cases. The BERT-PT in $(9/16)$ cases, BERT in $(13/16)$ cases, and HAST in $(15/16)$ cases. As regards the comparison of WoSe versus ELMO + WoSe implementation, we observe that the latter which uses contextual word embeddings favored the performance of WoSe mainly on multitoken terms prediction in most cases.

It is noteworthy to refer that in cases where the content was rich in implicit annotated aspects, such as the gray-shaded cases in lap14 and rest14, WoSe performance was superior to all other competitive benchmark methods. This is remarkable and indicates that our method developed a strong sentence compositionality for ATE and especially for the difficult aspect cases such as the implicit which are not referred directly in the content. As an indication of how our method performs on such cases where explicit and implicit annotated aspects exist, we employ an example from the rest14 dataset and present it in Table XIV.

In Table XIV, the first row demonstrates the sentence example and the ground-truth aspects. We annotated three explicit and three implicit, as these are highlighted in color by the labels $[e]$ and $[i]$, respectively. Now, we comment on how the benchmark methods that are depicted in Table XIV process and predict aspects in comparison with WoSe. We start with the HAST method, where we observe that it managed to predict successfully all three explicit aspects and only a portion of the implicit, numbered [5] and [6]. This prediction performance of HAST is also reflected at the results of Tables X and XII, where HAST performs competitive on single-token aspect terms, but is suboptimal on multitoken terms against the BERT-based methods and WoSe in most cases. The second method BAT managed to predict successfully two out of three explicit aspects and one

TABLE XV
BENCHMARK DATASETS AND STATISTICS ON IMPLICIT ASPECT EXTRACTION

	lap14		rest14		rest15		rest16		
#implicit	acc.	impl. 138	acc.	impl. 219	acc.	impl. 80	acc.	impl. 40	total 477
HAST (ours)	97.10%	134	94.06%	206	100.00%	80	100.00%	40	460
BERT (ours)	97.83%	135	92.69%	203	97.50%	78	100.00%	40	456
BERT-PT (ours)	94.93%	131	94.98%	208	98.75%	79	100.00%	40	458
BAT (ours)	97.83%	135	94.98%	208	95.00%	76	100.00%	40	459
WoSe	94.20%	130	96.80%	212	97.50%	78	97.50%	39	459
Elmo + WoSe	94.93%	131	98.17%	215	98.75%	79	97.50%	39	**464**

out of three implicit. Despite the state-of-the-art performance of BAT in the ATE task, this example shows some of the weaknesses of the method, where there are aspects, here the implicit that are not predicted successfully. Now, as regards our method, WoSe and ELMo + WoSe implementations, predicted successfully all explicit, but for ELMo + WoSe which failed in explicit numbered [4], and WoSe in only two out of three implicit. The ELMo + WoSe implementation predicted successfully the implicit numbered [5] and failed in [2] where no other method managed to predict any token from this aspect and in the implicit numbered [6] because it extracted more token-terms than those were labeled. Comparing this example with the results in Tables X and XII, we observe that the contextualized word embeddings favored the prediction performance of WoSe on multitoken terms, however, because of their attributes, the contextualized, in some cases WoSe fail to predict accurately the $B/I/O$ target boundaries. This is a studied issue which has been addressed in methods [8], [56], and [57].

Overall, in this section after the comparison with selected state-of-the-art benchmarks methods on single and multitoken aspect terms accuracy, and empirical evaluation on how competitive methods apply prediction on a real example, we may safely infer that the proposed approach successfully identifies aspect terms and more specifically the implicit.

VII. IMPLICIT ASPECT EXTRACTION

This section studies the overall performance of WoSe algorithm on implicit aspect extraction. Table XV presents these results. What we observe here is that despite WoSe does not perform the best score at each dataset, overall it extracted the greatest number of implicit aspects 464 from the total 477 implicit aspects from all datasets. Additionally, the rest of benchmark methods performance score results indicate that the implicit aspects relate to word-level encoding, as this is confirmed by the total performance score results of HAST method that performed the second best score with 460 total implicit aspects, respectively.

VIII. CASE STUDIES

In this section, we have included Table XVI which presents some case studies of the outputs of each method with respect to ground-truth aspects. What is observed is that in most cases the results of the proposed WoSe method aligns with the ground-truth aspects, while also the respective results of the compared benchmark methods with the ground truth.

TABLE XVI
CASE STUDIES FOR COMPARING THE OUTPUTS OF THE PROPOSED AND COMPARED METHODS

Golden Aspects	HAST	BERT	BERT-PT	BAT	WoSe
"Sandwiches," "burgers," "salads," "lemon-dressed cobb"	Sandwiches, burgers, salads	"Sandwiches," "burgers," "salads," "lemon-dressed cobb"	"Sandwiches," "burgers," "salads," "lemon-dressed cobb"	"Sandwiches," "burgers," "salads," "lemon-dressed cobb"	"Sandwiches," "burgers," "salads," "lemon-dressed cobb"
"Log on," "wifi connection," "battery life"	Battery life, wifi connection	"Log on," "wifi connection," "battery life"	"Log on," "wifi connection," "battery life"	"Log on," "wifi connection," "battery life"	"Log on," "wifi connection," "battery life"
"Half/half pizza"	Half/half pizza, buffalo	"Pizza," "buffalo"	"Pizza," "buffalo"	"Pizza," "buffalo"	"Pizza," "buffalo"
"Meal," "service," "ambiance"	Meal, service, ambiance, restaurant	"Meal," "service," "ambiance"	"Meal," "service," "ambiance," "restaurant"	"Meal," "service," "ambiance," "restaurant"	"Meal," "service," "ambiance," "restaurant"

IX. CONCLUSION AND FUTURE WORK

A. Conclusion

This work studies the task of ATE and implements an innovative neural network framework that successfully identifies implicit and explicit aspects. The method *uses* a machine translation cell and combines other neural modules that are tailored to improve the model's generalization. We experimented with four widely used datasets comparing with state-of-the-art approaches. The experimental results obtained from various setups document the superiority of the proposed framework. Key factors toward this positive outcome are the word and the sentence-level encoders introduced. Moreover, the CRF layer contributed toward identifying the linear relationship in the extraction of the aspect terms.

B. Future Work

In the immediate future, we intend to explore deeper architectures, such as [42] and [55], and modify them to be applicable for aspect categorization and coextraction of aspect/opinion terms. Moreover, as regards the implicit aspects we intend to study the surrounding context that defines these aspects, the relation to aspect categories, and how this content can be employed to improve aspect category classification. Finally, another direction in our research agenda is to utilize the capabilities of the aspect-extraction model to systems that involve language-based human–computer interaction such as conversational agents.

REFERENCES

[1] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA: ACM, 2004, pp. 168–177.
[2] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, "Aspect-based sentiment analysis: A survey of deep learning methods," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 6, pp. 1358–1375, Dec. 2020.

[3] Z. Fan, Z. Wu, X.-Y. Dai, S. Huang, and J. Chen, "Target-oriented opinion words extraction with target-fused neural sequence labeling," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1: Long and Short Papers, Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 2509–2518.

[4] W. An, F. Tian, P. Chen, and Q. Zheng, "Aspect-based sentiment analysis with heterogeneous graph neural network," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 1, pp. 403–412, Feb. 2023.

[5] X. Li and W. Lam, "Deep multi-task learning for aspect term extraction with memory interaction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2886–2892. Accessed: Sep. 11, 2017. [Online]. Available: https://aclanthology.org/D17-1310

[6] K. Liu, L. Xu, and J. Zhao, "Opinion target extraction using word-based translation model," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 1346–1356. Accessed: Jul. 14, 2014. [Online]. Available: https://aclanthology.org/D12-1123

[7] 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. (IJCAI)*, Stockholm, Sweden: AAAI Press, 2018, pp. 4194–4200.

[8] Z. Wei, Y. Hong, B. Zou, M. Cheng, and J. Yao, "Don't eclipse your arts due to small discrepancies: Boundary repositioning with a pointer network for aspect extraction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Seattle, Washington D.C: Association for Computational Linguistics, Jul. 2020, pp. 3678–3684. Accessed: Jul. 10, 2020. [Online]. Available: https://aclanthology.org/2020.acl-main.339

[9] 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.*, Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1433–1443.

[10] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. COLING 26th Int. Conf. Comput. Linguistics: Tech. Papers*, Osaka, Japan: COLING 2016 Organizing Committee, Japan, 2016, pp. 3298–3307.

[11] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics,* vol. 2: Short Papers, Baltimore, MD, USA: Association for Computational Linguistics, Jun. 2014, pp. 49–54.

[12] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Kerrville, TX, USA: Association for Computational Linguistics, 2016, pp. 616–626.

[13] H. Luo, T. Li, B. Liu, B. Wang, and H. Unger, "Improving aspect term extraction with bidirectional dependency tree representation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 7, pp. 1201–1212, Jul. 2019.

[14] D. Hyun, C. Park, M. Yang, I. Song, J. Lee, and H. Yu, "Target-aware convolutional neural network for target-level sentiment analysis," *Inf. Sci.*, vol. 491, pp. 166–178, 2019.

[15] E. R. Fonseca and J. L. G. Rosa, "A two-step convolutional neural network approach for semantic role labeling," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Dallas, TX, USA. Piscataway, NJ, USA: IEEE, Aug. 4–9, 2013, pp. 1–7.

[16] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Seattle, WA, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642.

[17] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Coupled multi-layer attentions for co-extraction of aspect and opinion terms," in *Proc. 31st AAAI Conf. Artif. Intell. (AAAI)*, San Francisco, California, USA: AAAI Press, 2017, pp. 3316–3322.

[18] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proc. Conf. Empirical Methods Natural Lang. Process.,* Austin, TX, USA: Association for Computational Linguistics, Nov. 2016, pp. 214–224.

[19] M. Tubishat, N. Idris, and M. A. Abushariah, "Implicit aspect extraction in sentiment analysis: Review, taxonomy, opportunities, and open challenges," *Inf. Process. Manage.*, vol. 54, no. 4, pp. 545–563, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457317306684

[20] Q. Su, K. Xiang, H. Wang, B. Sun, and S. Yu, "Using pointwise mutual information to identify implicit features in customer reviews," in *Proc. 21st Int. Conf. Comput. Process. Oriental Lang.: Beyond Orient: Res. Challenges Ahead (ICCPOL)*, Berlin, Heidelberg: Springer-Verlag, 2006, pp. 22–30. [Online]. Available: https://doi.org/10.1007/11940098_3

[21] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, New York, NY, USA: ACM, 2008, pp. 111–120. [Online]. Available: https://doi.org/10.1145/1367497.1367513

[22] X. Xu, X. Cheng, S. Tan, Y. Liu, and H. Shen, "Aspect-level opinion mining of online customer reviews," *China Commun.*, vol. 10, no. 3, pp. 25–41, 2013.

[23] W. Maharani, D. H. Widyantoro, and M. L. Khodra, "Aspect extraction in customer reviews using syntactic pattern," *Procedia Comput. Sci.*, vol. 59, pp. 244–253, 2015.

[24] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proc. SocialNLP@COLING*, 2014, pp. 28–37.

[25] Y. Zhang and W. Zhu, "Extracting implicit features in online customer reviews for opinion mining," in *Proc. 22nd Int. Conf. World Wide Web (WWW) Companion*, New York, NY, USA: ACM, 2013, pp. 103–104. [Online]. Available: https://doi.org/10.1145/2487788.2487835

[26] L. Sun, S. Li, J. Li, and J. Lv, "A novel context-based implicit feature extracting method," in *Proc. Int. Conf. Data Sci. Adv. Analytics (DSAA)*, 2014, pp. 420–424.

[27] J. Chen, L. Sun, Y. Peng, and Y. Huang, "Context weight considered for implicit feature extracting," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, 2015, pp. 1–5.

[28] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2493–2537, Nov. 2011.

[29] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Proc. INTERSPEECH*, 2013, pp. 3771–3775.

[30] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.

[31] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA: ACM, 2008, pp. 160–167. [Online]. Available: https://doi.org/10.1145/1390156.1390177

[32] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA: Association for Computational Linguistics, Nov. 2016, pp. 606–615.

[33] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. 30*, I. Guyon et al., Eds., Glasgow, Scotland, United Kingdom: Curran Associates, Inc., 2017, pp. 5998–6008.

[34] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI blog, 2019.

[35] 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: Human Lang. Technol.*, vol. 1: Long and Short Papers, Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

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

[37] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[38] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *Proc. Int. Conf. Lang. Resour. Eval. (LREC)*, 2018, pp. 52–55.

[39] 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: Human Lang. Technol.*, vol. 1: Long and Short Papers, Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 2324–2335.

[40] A. Karimi, L. Rossi, and A. Prati, "Adversarial training for aspect-based sentiment analysis with BERT," in *25th Int. Conf. Patt. Recog. (ICPR).*, 2020, pp. 8797–8803.

[41] Y. Pan, Y. Hong, Q. Xu, J. Chen, and J. Yao, "Aspect term extraction via contrastive learning over self-augmented data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2022, pp. 1–8.

[42] A. Karimi, L. Rossi, and A. Prati, "Improving BERT performance for aspect-based sentiment analysis," in *Proc. 4th Int. Conf. Natural Lang. Speech Process. (ICNLSP)*, M. Abbas and A. A. Freihat, Eds., Trento, Italy: Association for Computational Linguistics, Nov. 12–13, 2021, pp. 39–46. Accessed: Nov. 13, 2021. [Online]. Available: https://aclanthology.org/2021.icnlsp-1.5

[43] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian, and N. Al-Nabhan, "T-BERTSum: Topic-aware text summarization based on BERT," *IEEE Trans. Comput. Social Syst.*, vol. 9, no. 3, pp. 879–890, Jun. 2022.

[44] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[45] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 2: Short Papers, New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 464–468.

[46] C.-Z. A. Huang et al., "Music transformer," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[47] 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, 2016.

[48] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Double embeddings and CNN-based sequence labeling for aspect extraction," in *Proc. 56th Annu. Meet. Assoc. Comput. Linguistics*, vol. 2: Short Papers, Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 592–598.

[49] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn. (ICML)*, San Francisco, CA, USA: Morgan Kaufmann, 2001, pp. 282–289.

[50] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, Miyazaki, Japan: European Language Resources Association (ELRA), May 2018, pp. 52–55.

[51] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 2017.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR), Conf. Track Proc.*, San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds., May 7–9, 2015.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[54] M. E. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1: Long Papers, New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. Accessed: Jun. 6, 2018. [Online]. Available: https://aclanthology.org/N18-1202

[55] H. Yan, J. Dai, T. Ji, X. Qiu, and Z. Zhang, "A unified generative framework for aspect-based sentiment analysis," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.,* vol. 1: Long Papers, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Bangkok, Thailand: Online, Association for Computational Linguistics, Aug. 2021, pp. 2416–2429. Accessed: Aug. 6, 2021. [Online]. Available: https://aclanthology.org/2021.acl-long.188

[56] D. Ma, S. Li, and H. Wang, "Joint learning for targeted sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium: Association for Computational Linguistics, Oct.–Nov. 2018, pp. 4737–4742. Accessed: Nov. 4, 2018. [Online]. Available: https://aclanthology.org/D18-1504

[57] X. Li, L. Bing, P. Li, and W. Lam, "A unified model for opinion target extraction and target sentiment prediction," in *Proc. 33rd AAAI Conf. Artif. Intell. 31st Innov. Appl. Artif. Intell. Conf. 9th AAAI Symp. Educational Adv. Artif. Intell. (AAAI/IAAI/EAAI)*, Honolulu, Hawaii, USA: AAAI Press, 2019. Accessed: Jul. 17, 2019. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33016714

Pantelis Agathangelou received the M.Sc. degree in information systems from the Faculty of Pure and Applied Sciences, Open University of Cyprus, Nicosia, Cyprus, in 2013. He is currently working toward the Ph.D. degree with the Department of Computer Science, University of Nicosia, Nicosia, Cyprus.

His research interests include data mining, pattern classification, sentiment analysis, mining social networks, and artificial intelligence. He has codeveloped DidaxTo (http://deixto.com/didaxto/), a tool that implements an unsupervised approach for discovering patterns that will extract a domain-specific dictionary from product reviews.

Ioannis Katakis received the B.Sc., M.Sc. and Ph.D. degrees from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2004, 2007, 2009, respectively. He is a Professor, a Co-Founder, and a Co-Director of the Artificial Intelligence Laboratory (https://ailab.unic.ac.cy/), Computer Science Department, University of Nicosia, Nicosia, Cyprus, and an Academic Coordinator of the M.Sc. degree in data science. He served in multiple universities as a Lecturer and a Senior Researcher. He was included in the J. Ioannides' (Stanford University) list of top young Greek scientists based on the impact of their work. His research has been cited more than 9000 times by the scientific community. He published more than 70 papers in international conferences and scientific journals, organized workshops, and special issues. His research interests include data science, machine learning, deep learning, sentiment analysis, smart cities, conversational agents, social networks, and computational social science.

Dr. Katakis is an Editor of journals *Information Systems* and *Online Social Networks and Media*.

He has an extensive experience in European research projects where he participated as a Principal Investigator and Task Leader.

Panagiotis Kasnesis received the M.Sc. and Ph.D. degrees in computer science from the Department of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, in 2013, 2018, respectively.

He is currently a Senior Researcher with the University of West Attica, Athens, Greece, and is a CTO and a Co-Founder of ThinGenious PC, Athens, Greece. His research interests include machine/deep learning, multiagent systems, and the Internet of Things.