ALDONAr: A Hybrid Solution for Sentence-Level Aspect-Based Sentiment Analysis Using a Lexicalized Domain Ontology and a Regularized Neural Attention Model

Donatas Meškelė, Flavius Frasincar

Erasmus University Rotterdam, P.O. Box 1738, 3000DR, Rotterdam, the Netherlands

Abstract

Aspect-based sentiment analysis allows one to compute the sentiment for an aspect in a certain context. One problem in this analysis is that words possibly carry different sentiments for different aspects. Moreover, an aspect's sentiment might be highly influenced by the domain-specific knowledge. In order to tackle these issues, in this paper, we propose a hybrid solution for sentence-level aspect-based sentiment analysis using A Lexicalized Domain Ontology and a Regularized Neural Attention model (ALDONAr). The bidirectional context attention mechanism is introduced to measure the influence of each word in a given sentence on an aspect's sentiment value. The classification module is designed to handle the complex structure of a sentence. The manually created lexicalized domain ontology is integrated to utilize the field-specific knowledge. Compared to the existing ALDONA model, ALDONAr uses BERT word embeddings, regularization, the Adam optimizer, and different model initialization. Moreover, its classification module is enhanced with two 1D CNN layers providing superior results on standard datasets.

Keywords: Aspect-based sentiment analysis, Hybrid model, Lexicalized domain ontology, Bidirectional gated neural network, Regularization

Email addresses: donatas.meskele@gmail.com (Donatas Meškelė), frasincar@ese.eur.nl (Flavius Frasincar)

1. Introduction

Sentiment Analysis (SA) is usually addressed using aspect extraction [1, 2], opinion identification [3], or aspect-based sentiment classification [4, 5, 6]. Due to the scale of each approach, we solely focus on the latter. Knowledge-based, machine learning, and hybrid are the most frequently used techniques for aspect-based sentiment classification [7].

Hybrid sentiment classifiers, which use machine learning and knowledge-driven models, tend to show superior performance compared to separate approaches [8, 9, 10, 11]. In a two-step procedure, [10] uses a lexicalized domain ontology classifier backed up with a support vector machine (SVM). However, recent studies have proposed advanced deep learning methods as auspicious substitutes for SVMs [12, 13, 14, 15]. [13] introduces the Content Attention-Based Aspect-Based Sentiment Classification model (CABASC). By employing two sequential neural attention mechanisms: one for a global view (disregarding word order) and one for a local view (considering word order by integrating a gated recurrent unit (GRU) [16]), CABASC proves to be a state-of-the-art aspect-based sentiment classification model. Because of this the SVM model from [10] was replaced with a Deep Bidirectional Gated Recurrent Unit (DBGRU) based on CABASC and presented in our previous work [8]. The proposed model dubbed A Lexicalized Domain Ontology and a Neural Attention Model (ALDONA) provided increased accuracy for aspect-based sentiment classification compared to the CABASC model.

Although time consuming, the manually created ontology is chosen to ensure accurate relationships among entities and their properties. Thus, the field-specific knowledge is captured using a lexicalized domain ontology introduced in [10]. The classifier distinguishes generic, category-dependent, and context-dependent sentiment types, and using different rules categorizes them into *Positive* and *Negative* classes. Due to its ambiguous representation, the *Neutral* class has been omitted.

The data flows through the neural attention model as follows. Initially, the sentence and aspect words are converted into corresponding word embedding vectors, then the bidirectional context attention mechanism extracts word ordering, past, and future information and correlations among aspects and their context words and assigns weights for every word. This information is summarized into the weighted embedding sentence vector by the sentence-level content attention mechanism. Finally, the aspect sentiment is derived in the classification module.

In this paper, we extend our previous work on ALDONA [8]. We replace GloVe word embeddings [17] with BERT Large Uncased word embeddings [18] (BERT contextual word embeddings are better proxies for word semantics in text than GloVe non-contextual word embeddings), the minibatch gradient descent with the minibatch Adam optimizer [19] (a state-of-the-art optimizer), adjust initialization of weight matrices (using a normal distribution with a smaller variance due to the relatively high number of neurons), and introduce an L2 norm regularization term (to reduce overfitting). Moreover, the classification module has been enhanced with two 1D CNN layers for additional flexibility in the sentiment computations. The resulting method is called A Lexicalized Domain Ontology and a Regularized Neural Attention Model (ALDONAr). Evaluated on benchmark datasets, ALDONAr produced higher accuracy in comparison with other models, including the state-of-the-art CABASC [13], DBGRU [8], and ALDONA [8].

The structure of the paper is as follows. Some complementary literature is presented in Section 2. The detailed definition of ALDONAr is given in Section 3. Section 4 and Section 5 explain the data and compare ALDONAr performance with other benchmark models, respectively. Conclusion and future research are presented in Section 6. The source code (in Python 3) is provided at `https://github.com/donmesh/ALDONAr`.

2. Related Literature

Due to its complexity, sentiment analysis (SA) [20] is usually performed using several subtasks, such as aspect extraction [1, 2], opinion identification [3], and aspect-based sentiment classification [4, 5, 6]. The latter subtask is often solved using knowledge-based, machine learning, and hybrid models [7]. Knowledge-driven sentiment classification exploits domain specific knowledge derived by an ontology reasoner from a given ontology to determine a sentiment value. The employed ontology can be designed manually [10, 11], semi-automatically [21], or fully automatically [22]. Methods using ontologies crafted using design criteria [23] have shown good performance for aspect-based sentiment classification [10, 21].

Although advanced machine learning algorithms might increase efficacy of the classification performance, this approach usually requires an expensive manual feature engineering in a data preprocessing stage. The Bag-of-Words (BoW) model is often used as an effective simplifying approach [24, 25, 26]. Neglecting word semantics and ordering strongly undermines the performance of this technique [27]. As a result, intricate deep learning methods have gained attention in the recent research [28, 29].

[30] presents a relative aspect-context position model which penalizes distant context words. To account for word semantic relationships, the recurrent neural networks (RNN) have been employed in sentiment analysis tasks [15, 31, 32]. Extracting information from the left and right context words for a given aspect [32], as well as exploiting syntax and semantics of a given sentence by means of a bidirectional gated neural network [15], have proved to be exceptionally useful. The state-of-the-art Recurrent Attention Network (RAN) employs this idea and extracts polarity values with a deep position-weighted bidirectional LSTM [12]. On the other hand, CABASC has shown that the used LSTM could be successfully replaced by a Gated Recurrent Unit (GRU) in order to increase classification efficiency [13]. CABASC has also shown better performance than many other advanced classifiers, including Attention-Based Aspect Embedding-LSTM (ATAE-LSTM) which enhances word embeddings and LSTM hidden states with the aspect embeddings [33], Memory Network (MemNet) which uses multi-hop attention memory [30], Interactive Attention Network (IAN) exploiting one LSTM attention model for a target and another for context [34].

Using domain-specific knowledge and statistical relations, hybrid sentiment classification models have proven to be highly effective [9, 11]. [10] introduces a two-step classifier: a lexicalized domain ontology is backed up with a support vector machine (SVM). In the first stage the ontology reasoner infers the sentiment type (generic, category-dependent, or context-dependent) of a word and predicts its polarity value (positive or negative). Given that both or none of the values are assigned, the SVM bag-of-words phase is activated. The proposed solution was enhanced with neural networks [12, 14] and their gated representations [13, 15] and elaborated in [8]. In this paper we extend this former research and propose a new model dubbed A Lexicalized Domain Ontology and a Regularized Neural Attention model (ALDONAr).

3. Methodology

To determine the polarity value of a specific sentiment ALDONAr aims to extract the domain-based knowledge using a lexicalized domain ontology. If this is unsuccessful, the sentiment is derived using word ordering, past, and future information, as well as correlations among context and aspect words by a neural attention model.

3.1. Lexicalized Domain Ontology

The lexicalized domain ontology reasoner infers indirect relationships among entities and their properties and in conjunction with direct relationships determine the polarity value of an aspect.

We make use of a manually created ontology (proposed by [10]), as it is more reliable in terms of its correctly defined elements compared to its (semi-) automatically created counterparts. The ontology's main components are the *SentimentValue*, a super-class of *Positive* and *Negative* classes, the *Neutral* class is not implemented due to its ambiguous representation in datasets; the *AspectMention* models aspect mentions; the *SentimentMention* specifies the various sentiment expressions and their types.

Three sentiment types are defined in this ontology. Type-1 represents generic sentiments which have unequivocal meaning in every situation and for every aspect (e.g., "disgusting" belongs to *Negative* class). Type-2 sentiments are category-dependent, and thus depend on the category they are used with (e.g., "savory" is *Positive* for the *SustenanceMention* class, but is not defined and ignored for the *NamedLocationMention*). Type-3 represents context-dependent sentiments that depend on an aspect's category (e.g., *warm beer* is *Negative*, but *warm pizza* is *Positive*). Driven by domain text, new subclasses are created given that those sentiment-aspect combinations do not yet exist in the knowledge graph. As the types are ordered and exclusive, dependency to the higher order category is only possible if none of the lower order types is matched (e.g., a given sentiment can be of type-2 only if its type is not type-1). Sentiment mentions are linked to their corresponding aspects by means of the *aspect* annotation.

In order to cope with multiple words having the same meaning, word lexicalizations (as lemmas) are introduced in terms of the *lex* annotation attached to every concept. Word negation is handled by confirming that neither of its three preceding words is in the negation set {*not, no, never, isn't, aren't, won't, wasn't, weren't, haven't, hasn't, don't, doesn't, can't, couldn't*} [35]. When encountered, these words change sentiment polarity value to its opposite.

3.2. Neural Attention Model

The neural attention model is designed to extract an aspect's polarity value based on statistical relationships among an aspect and its context words. The model has four main components which are built on top of another one and are presented in the following subsections.

3.2.1. Word Embeddings

Given an input sentence of length N and an aspect of length L, these can be represented as sequences of tokens $S = [s_1, s_2, ..., s_N]^T$ and $S_a = [s_i, s_{i+1}, ..., s_{i+L}]^T$, respectively. For each word, which is in the WordPiece vocabulary [36], the BERT Uncased assigns a predefined word id. As there might be some domain specific words which are not in the vocabulary, the model contains 1000 empty slots which can be used to overcome this issue. The input sentence is converted into a tensor of shape [V x N x d], where V is number of layers (12 for BERT Base and 24 for BERT Large), N is the length of a sentence, and d is the length of an embedding vector (768 for BERT Base and 1024 for BERT Large). Summing the last four layers is shown to produce favorable results [18]. Hence, the embedded sentence and the embedded aspect are given by:

$$E = [e_1, e_2, ..., e_N]^T \in \mathbb{R}^{N \times d}, \tag{1}$$

$$E_A = [e_i, e_{i+1}, ..., e_{i+L}]^T \in \mathbb{R}^{L \times d}, \tag{2}$$

respectively.

The sentence S is split into two parts: from the beginning of the sentence to the end of the aspect (S_{LS}) and from the beginning of the aspect to the end of the sentence (S_{RS}). Their respective embedded representations are:

$$\begin{aligned} E_{LS} &= [e_1, ..., e_{i-1}, e_i, ..., e_{i+L}]^T, \\ E_{RS} &= [e_i, ..., e_{i+L}, e_{i+L+1}, ..., e_N]^T. \end{aligned} \tag{3}$$

3.2.2. Bidirectional Context Attention Mechanism

A potential flaw of the unidirectional gated recurrent unit (presented as a sufficient method to infer relationships among words in [13]) is that it accounts only for information produced by processing words in forward manner. As a result, importance of each word is not completely determined.

The problem can be handled by the bidirectional recurrent neural networks (BRNN) which incorporate both past (words which are before the current word) and future (words which are after the current word) information to infer the attention weight for every word in a sentence.

We provide an overview of the unidirectional gated recurrent unit (GRU) [16] and the bidirectional gated recurrent unit (BGRU) in order explain the bidirectional context attention module. Influence of a previous hidden state h_{n-1} on a newly generated memory $\tilde{h}_n$ is determined by the reset gate r_n in the gated recurrent unit. The new hidden state h_n is then constructed from the output of the previous hidden state h_{n-1} and the new memory $\tilde{h}_n$ controlled by the update gate u_n:

$$\begin{aligned}
r_n &= \sigma(e_n W_r + h_{n-1} U_r + b_r), \\
u_n &= \sigma(e_n W_u + h_{n-1} U_u + b_u), \\
\tilde{h}_n &= \tanh(e_n W_h + (r_n \odot h_{n-1}) U_h + b_{\tilde{h}}), \\
h_n &= u_n \odot h_{n-1} + (1 - u_n) \odot \tilde{h}_n,
\end{aligned} \tag{4}$$

where $\odot$ represents the element-wise multiplication, σ and tanh are the sigmoid and hyperbolic tangent functions, respectively, $e_n \in \mathbb{R}^{1\times d}$ is a word embedding vector, $r_n \in \mathbb{R}^{1\times d}$ and $u_n \in \mathbb{R}^{1\times d}$ are the reset and update gates, and $\tilde{h}_n \in \mathbb{R}^{1\times d}$ and $h_n \in \mathbb{R}^{1\times d}$ are the new memory and the new hidden state, respectively. $W_r \in \mathbb{R}^{d\times d}$, $U_r \in \mathbb{R}^{d\times d}$, $W_u \in \mathbb{R}^{d\times d}$, $U_u \in \mathbb{R}^{d\times d}$, $W_h \in \mathbb{R}^{d\times d}$, $U_h \in \mathbb{R}^{d\times d}$ are weight matrices and $b_r \in \mathbb{R}^{1\times d}$, $b_u \in \mathbb{R}^{1\times d}$, $b_{\tilde{h}} \in \mathbb{R}^{1\times d}$ are bias vectors. An alternative to GRU is the long short-term memory model (LSTM) [37], which introduces additional flexibility. We have used GRU instead of LSTM due to its reduced complexity that enables a shorter computation time.

The bidirectional gated recurrent unit (BGRU) is created by applying GRU from both ends of a sequence and then combining their final states into one:

$$\begin{aligned}
\overrightarrow{h_n} &= f(\overrightarrow{\Theta} \,|\, e_n, \overrightarrow{h_{n-1}}), \\
\overleftarrow{h_n} &= f(\overleftarrow{\Theta} \,|\, e_n, \overleftarrow{h_{n+1}}), \\
h_n &= g(\overrightarrow{h_n},\ \overleftarrow{h_n}),
\end{aligned} \tag{5}$$

where $\overrightarrow{h_n}$ and $\overleftarrow{h_n}$ are hidden states obtained from the forward and backward directions, e_n is the new input, $\overrightarrow{\Theta}$ and $\overleftarrow{\Theta}$ are parameters to be optimized (here, two sets of weight matrices and bias vectors described in Equation 4), $f(\cdot)$ is the unidirectional recurrent neural network (here, GRU) and $g(\cdot)$ is the activation function combining $\overrightarrow{h_n}$ and $\overleftarrow{h_n}$. Here, the $g(\cdot)$ function is defined as follows:

$$h_n = \tanh(\overrightarrow{h_n} W_{fw} + \overleftarrow{h_n} W_{bw} + b_{bi}), \tag{6}$$

where $W_{fw} \in \mathbb{R}^{d\times d}$ and $W_{bw} \in \mathbb{R}^{d\times d}$ are weight matrices and $b_{bi} \in \mathbb{R}^{1\times d}$ is a bias vector. The hyperbolic tangent activation function is used instead of the classical sigmoid due to its better performance [38].

The bidirectional gated recurrent unit separately transforms the left and right embedded sentence parts into the final hidden states:

$$\begin{aligned} H_{LS} &= [h_1, ..., h_{i_l-1}, h_{i_l}, ..., h_{i_l+L}]^T, \\ H_{RS} &= [h_{i_r}, ..., h_{i_r+L}, h_{i_r+L+1}, ..., h_N]^T. \end{aligned} \tag{7}$$

The multilayer perceptron is then utilized to assign a bidirectional context attention weight for each word:

$$\begin{aligned} \beta_l &= \sigma(h_l W_1 + b_1) + b_l, \\ \beta_r &= \sigma(h_r W_2 + b_2) + b_r, \end{aligned} \tag{8}$$

$$\begin{aligned} \beta_{LS} &= [\beta_1, ..., \beta_{i_l}, ..., \beta_{i_l+L}]^T, \\ \beta_{RS} &= [\beta_{i_r}, ..., \beta_{i_r+L}, ..., \beta_N]^T, \\ \beta_A &= [\tfrac{\beta_{i_l}+\beta_{i_r}}{2}, ..., \tfrac{\beta_{i_l+L}+\beta_{i_r+L}}{2}]^T, \\ \beta_{LC} &= [\beta_1, ..., \beta_{i-1}]^T, \\ \beta_{RC} &= [\beta_{i+L+1}, ..., \beta_N]^T, \\ \beta &= [\beta_{LC}, \beta_A, \beta_{RC}]^T, \end{aligned} \tag{9}$$

where $W_1 \in \mathbb{R}^{d\times 1}$ and $W_2 \in \mathbb{R}^{d\times 1}$ are weight matrices, $b_1 \in \mathbb{R}$ and $b_2 \in \mathbb{R}$ are biases, and $b_l \in \mathbb{R}$ and $b_r \in \mathbb{R}$ are hyperparameters.

As β_{LS} and β_{RS} both contain aspect information (namely, $[\beta_{i_l}, ..., \beta_{i_l+L}]^T$ and $[\beta_{i_r}, ..., \beta_{i_r+L}]^T$), the final aspect weight β_A is constructed as an average of their respective parts. The left and right context weights (β_{LC} and β_{RC}, respectively) are then obtained by excluding aspect information. The bidirectional context attention weights $\beta \in \mathbb{R}^{N\times 1}$ are produced by concatenating the left context, aspect, and right context attention weights.

The weighted memory $M_w = [m_{w_1}, ..., m_{w_N}]^T \in \mathbb{R}^{N\times d}$ is constructed to incorporate relations between a word and its context. Each element $m_{w_n} \in \mathbb{R}^{1\times d}$ is calculated as follows:

$$m_{w_n} = \beta_{tiled} \odot e_n, \tag{10}$$

where $\beta_{tiled} \in \mathbb{R}^{1\times d}$ is formed by replicating an element $\beta \in \mathbb{R}$ d times and $\odot$ represents the element-wise multiplication. The weighted memory M_w is then fed into the sentence-level content attention module.

3.2.3. Sentence-Level Content Attention Module

The sentence-level content attention module captures the global view of a sentence by integrating the weighted memory with the explicit aspect and sentence representations.

Vector representations produced by averaging all word embedding vectors have been shown to be an effective representation technique in the past [39]. Thus, aspect and sentence's summaries are given by:

$$v_a = \frac{1}{L}\sum_{l=1}^{L} e_l, \quad v_s = \frac{1}{N}\sum_{n=1}^{N} e_n. \tag{11}$$

Then a word attention score c_n is computed for every word in a sentence:

$$c_n = \tanh(m_{w_n} W_3 + v_a W_4 + v_s W_5 + b_3) W_6, \tag{12}$$

where $m_{w_n} \in \mathbb{R}^{1\times d}$ is the weighted memory slice of the word s_n, $v_a \in \mathbb{R}^{1\times d}$ and $v_s \in \mathbb{R}^{1\times d}$ are the aspect and sentence representations, respectively, $W_3 \in \mathbb{R}^{d\times m}$, $W_4 \in \mathbb{R}^{d\times m}$, $W_5 \in \mathbb{R}^{d\times m}$, $W_6 \in \mathbb{R}^{m\times 1}$ are weight matrices and $b_3 \in \mathbb{R}^{1\times m}$ is a bias vector.

By means of the *softmax* function every word attention score is converted into a word attention weight:

$$\alpha_n = exp(c_n) / \sum_{j=1}^{N} exp(c_j) \tag{13}$$

producing a sentence attention weight vector $\alpha = [\alpha_1, ..., \alpha_N]^T \in \mathbb{R}^{N\times 1}$. The locally weighted memory is weighted using the newly constructed attention weights that results in a weighted embedding sentence vector $v_{we} \in \mathbb{R}^{1\times d}$:

$$v_{we} = \alpha^T M_w \tag{14}$$

where $\alpha = [\alpha_1, ..., \alpha_N]^T \in \mathbb{R}^{N\times 1}$, $M_w \in \mathbb{R}^{N\times d}$.

3.2.4. Classification Module

A combination of explicit relationships between the weighted embedding sentence vector v_{we} and the aspect representation v_a, and the former vector v_{we} and the sentence representation v_s, generated by the classification module, support model's ability to generalize:

$$\begin{aligned} v_{sw} &= \tanh(v_s W_7 + v_{we} W_8 + b_4), \\ v_{aw} &= \tanh(v_a W_9 + v_{we} W_{10} + b_5), \\ v_o &= \tanh(v_{sw} W_{11} + v_{aw} W_{12} + b_6), \end{aligned} \tag{15}$$

where $W_7 \in \mathbb{R}^{d\times d}$, $W_8 \in \mathbb{R}^{d\times d}$, $W_9 \in \mathbb{R}^{d\times d}$, $W_{10} \in \mathbb{R}^{d\times d}$, $W_{11} \in \mathbb{R}^{d\times k}$, and $W_{12} \in \mathbb{R}^{d\times k}$ are weight matrices, $b_4 \in \mathbb{R}^{1\times d}$, $b_5 \in \mathbb{R}^{1\times d}$, and $b_6 \in \mathbb{R}^{1\times k}$ are bias vectors, $v_a \in \mathbb{R}^{1\times d}$ and $v_s \in \mathbb{R}^{1\times d}$ are the aspect and sentence representations, and $v_{we} \in \mathbb{R}^{1\times d}$ is the weighted embedding sentence vector.

The output vector $v_o \in \mathbb{R}^{1\times k}$ is fed into two 1D convolutional layers ($CONV1D$) with $stride = 1$ and $padding = SAME$ to extract meaning from the weighted embedding sentence vector.

$$\begin{aligned} conv_1 &= \tanh(CONV1D(v_o, kernel_1) + b_{conv_1}), \\ conv_2 &= \tanh(CONV1D(conv1, kernel_2) + b_{conv_2}), \end{aligned} \tag{16}$$

where $kernel_1$ and $kernel_2$ are kernels ($[kernel\_size$, $input\_units = k$, $output\_units = q]$ and $[kernel\_size$, $input\_units = q$, $output\_units = q]$, respectively), $b_{conv_1} \in \mathbb{R}^{1\times q}$ and $b_{conv_2} \in \mathbb{R}^{1\times q}$ are bias vectors.

An illustration of 1D CNN is given in Figure 1. A horizontal sequence of black rectangles represents an input vector, a striped black block - a convolutional kernel, a checked black block - an output vector. How many blocks to the right the kernel is pushed is determined by $stride$, while $padding = SAME$ ensures the same dimensions of input and output vectors.

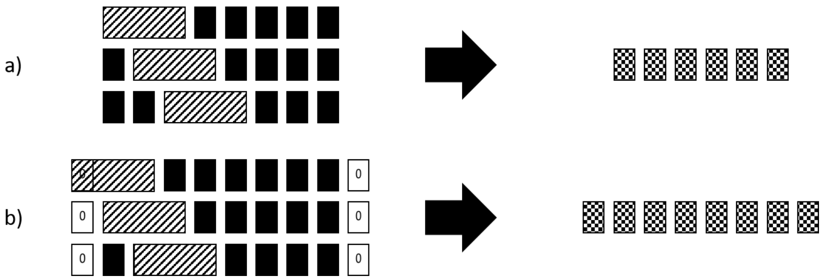

Figure 1: Given an input vector (a sequence of black rectangles) and a kernel (a black striped rectangle), their 1D convolution is represented as a sequence of checked black rectangles. Both parts have the same input vector of length 8, $kernel\_size = 3$, and $stride = 1$. However, part *a)* does not use any padding ($padding = 0$), whereas part *b)* has $padding = SAME$. The same dimensions are achieved by appending enough zeros at the beginning and at the end of the input vector.

The output $conv_2 \in \mathbb{R}^{1\times q}$ is converted into a vector $v_L \in \mathbb{R}^{1\times|C|}$ by the linear layer:

$$v_L = conv_2 W_{13} + b_7, \tag{17}$$

where $|C|$ is the number of possible aspect polarity categories, $W_{13} \in \mathbb{R}^{q\times|C|}$ is a weight matrix and $b_7 \in \mathbb{R}^{1\times|C|}$ is a bias vector.

Finally, aspect's polarity probabilities $p \in \mathbb{R}^{1\times|C|}$ are then obtained by applying the *softmax* function on the output vector of the linear layer v_L:

$$p = softmax(v_L). \tag{18}$$

3.2.5. Regularization and Loss Function

The model's complexity is controlled by employing the dropout technique. An additional method that has been used to reduce overfitting is L2 (Tikhonov) regularization. The cross-entropy loss is chosen as the objective function:

$$loss = -\sum_{C}\sum_{S} y_{c,s}\ln(p_{c,s}) + \lambda||\Theta||_2^2, \tag{19}$$

where C is the set of polarity categories, S are the training examples, $p_{c,s} \in [0,1]$ is the estimated probability that a given aspect in a sentence s belongs to a category c, $y_{c,s} \in \mathbb{B}$ is the true probability that the aspect in the sentence s is in the category c, λ is the weight of the L2-regularization term, and Θ is the parameter set which contains all weight matrices introduced previously.

4. Data

The Special Interest Group on the Lexicon of ACL (SIGLEX) has made the data publicly available in terms of an annually held SemEval (Semantic Evaluation) workshop. We make use of "Subtask 1 Restaurant Domain English Training Data" (train dataset 2016) and "Subtask 1 Restaurant Domain English Gold Annotations Data" (test dataset 2016) from the SemEval 2016 Task 5 [6]. Additionally, we utilize "2015 ABSA Restaurant Reviews - Train Data" (train dataset 2015) and "2015 ABSA Restaurants Reviews - Test Data - Gold Annotations" (test dataset 2015) from the SemEval 2015 Task 12 [5]. Every opinion in a review sentence has an aspect (*target*), aspect category (*category*), character-wise location of an aspect (*from* and *to*), and the aspect's polarity value (*polarity*). An example is shown in Figure 2.

It can be seen in Table 1 that data is skewed and positive reviews form the majority class in training and test datasets. The bar charts in Figure 3 depict aspect category distributions. The most dominant category is "food#quality".

The following normalization procedure is performed. We replace all upper case letters by their lower case counterparts, change "&quot;", "&apos;" and "&amp;" into a double quote symbol ("), an apostrophe ('), and the word "and", respectively. Furthermore, we strip blank spaces, punctuation signs,

```
<sentence id="en_PagodaRestaurant_478006817:4">
    <text>Nice ambience, but highly overrated place.</text>
    <Opinions>
        <Opinion target="ambience" category="AMBIENCE#GENERAL"
            polarity="positive" from="5" to="13"/>
        <Opinion target="place" category="RESTAURANT#GENERAL"
            polarity="negative" from="36" to="41"/>
    </Opinions>
</sentence>
```

Figure 2: Example data snippet.

	Negative		Neutral		Positive		Total	
	Freq.	%	Freq.	%	Freq.	%	Freq.	%
Train data 2016	488	25.97	72	3.83	1319	70.20	1879	100
Test data 2016	135	20.77	32	4.92	483	74.31	650	100
Train data 2015	280	21.89	36	2.81	963	75.29	1279	100
Test data 2015	207	34.67	37	6.20	353	59.13	597	100

Table 1: Polarity distribution in train and test datasets

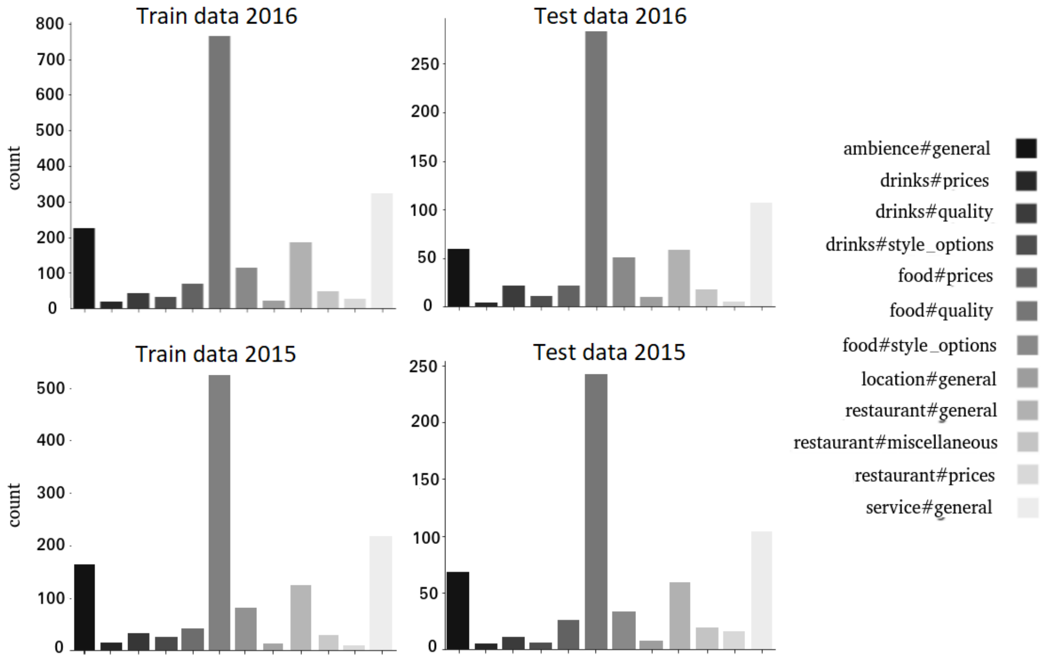

Figure 3: Aspect categories in train and test datasets.

and numbers. Sentences having only implicit aspects (*target="NULL"*) are left out, reducing our train and test datasets by approximately 25%. The implicit aspects are not considered by our model. Moreover, sentences without *Opinions* are also excluded (up to 5% from train and test datasets) without any loss, as they are irrelevant for aspect-based sentiment analysis. Sentences containing several aspects are treated as separate instances. Moreover, the train dataset is split into 75/25 proportions for training and validation (hyperparameter search) purposes. To be used in conjunction with our domain ontology, words are tokenized and lemmatized.

5. Performance Evaluation

We compare the performance of ALDONAr to other benchmark models. The lexicalized domain ontology classifier (Ont) proposed by [10] is limited to *Negative* and *Positive* aspect polarity values. In case it does not provide a definite prediction, the majority class is assigned (here, *Positive*).

Along with simpler neural attention models we evaluate the state-of-the-art CABASC which is based on the context attention model [13]. BaseA exploits the content attention mechanism [13], BaseB uses the sentence-level content attention model [13], while BaseC employs the sentence-level position attention mechanism [13].

Furthermore, we modified the context attention mechanism and introduced other reference models. By replacing the unidirectional gated recurrent unit with the unidirectional long short-term memory model, we created CTX-LSTM [8]. Substitution of the mentioned model with the bidirectional long short-term memory module resulted in CTX-BLSTM [8]. Similarly, we introduced CTX-BGRU which utilizes the bidirectional gated recurrent unit [8]. Additionally, the Deep Bidirectional Gated Recurrent Unit (DBGRU) [8] and A Lexicalized Domain Ontology and a Neural Attention model (ALDONA) [8] have been assessed. ALDONAr-base is obtained substituting BERT Large with BERT Base in ALDONAr.

Performance of ALDONAr and all methods mentioned above have been summarized in Table 2. Accuracy has been chosen as a common evaluation criterion for SemEval assignments. It is not surprising that higher accuracy has been achieved by more sophisticated techniques, as they tend to capture various data peculiarities unavailable to less intricate models. This is mainly caused by additional flexibility provided by the neural attention layers.

	2016		**2015**	
	Train accuracy	**Test accuracy**	**Train accuracy**	**Test accuracy**
Ont	75.5	77.4	79.8	66.2
BaseA	87.7	84.6	87.7	76.4
BaseB	87.8	84.6	85.4	78.2
BaseC	85.3	83.8	90.4	79.2
CABASC	85.3	84.8	89.0	77.7
CTX-LSTM	86.5	84.6	91.6	77.7
CTX-BGRU	86.0	85.4	86.0	75.9
CTX-BLSTM	86.0	85.2	85.7	75.9
DBGRU	91.1	85.5	85.9	77.4
ALDONA	90.7	86.1	88.8	80.6
ALDONAr-base	88.8	86.5	89.8	82.2
ALDONAr	93.1	87.1	92.7	83.8

Table 2: Percentage classification accuracy for train and test datasets achieved by various models

Compared to our previous research from [8], ALDONAr includes the advanced Adam optimizer [19] ($\beta_1 = 0.9$, $\beta_2 = 0.999$), L2 norm regularization term ($\lambda = 0.01$), and normally ($N(0, 0.015^2)$) initialized weight matrices. We have also replaced GloVe with BERT Large word embeddings and introduced two 1D CNN layers ($stride = 1$, $padding = SAME$, $kernel\_size = 2$). Other hyperparameters are $b_r = b_l = 0.5$, $d = 1024$, $m = 300$, $q = 256$, $k = 128$, $dropout\_probability = 0.3$, and $batch\_size = 128$. We have reimplemented all benchmark models and introduced their stable versions. Based on the accuracy reported in Table 2 ALDONAr is the best performing model.

Hyperparameters ($dropout\_probability$, $batch\_size$, β_1, β_2, $kernel\_size$, $stride$, $padding$, m, q, k, initialization of Normal distribution) are obtained using the grid search method and evaluated based on the accuracy on the validation set (25% of the train dataset). The hyperparameter d is determined by the length of the used word embedding vectors, whereas b_r and b_l are taken as given in [13].

As it can be observed in Figure 4, ALDONAr can handle complex sentences, where sentiment is rather implicitly expressed. The model assigns similar weights to all words, showing their almost uniform importance deriving the polarity value (negative) of “margaritas” (because the whole sentence conveys the meaning and not a specific word). However, the word “either” is stonger emphasized, which indicates that it plays a major role in the given sentence (as backed also by intuition).

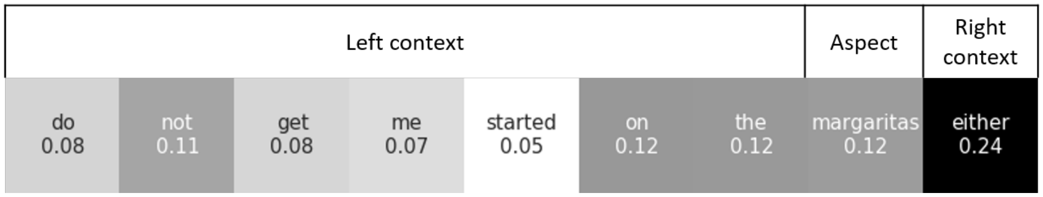

Figure 4: ALDONAr attention visualization.

6. Conclusion

In this paper, we proposed a new model called ALDONAr that extends the 2-stage hybrid model for sentence-level aspect-based sentiment classification ALDONA. As ALDONA, ALDONAr is constructed to determine aspect's polarity by means of a lexicalized domain ontology reasoner and statistical relations extracted by a neural attention model. Differently than ALDONA, ALDONAr uses a more advanced optimizer, a regularizer, a different model initialization, as well as a CNN-boosted classification module and BERT Large word embeddings. Using two standard datasets, it has been shown that ALDONAr provides better results (in terms of accuracy) than ALDONA.

Future research can concentrate on classification of implicit aspects. For this purpose, one could identify the most similar aspect words which could serve as proxies for aspects. Moreover, expansion of the lexicalized domain ontology would allow one to capture new concept relationships that could potentially lead to better classification accuracy. Thus, we plan to investigate semi-automatic methods for domain ontology building in our sentiment analysis context.

References

[1] Z. Chen, A. Mukherjee, B. Liu, Aspect Extraction with Automated Prior Knowledge Learning, in: 52nd Annual Conference of the Association for Computational Linguistics (ACL 2014), volume 1, ACL, 2014, pp. 347–358.

[2] S. Poria, E. Cambria, A. Gelbukh, Aspect Extraction for Opinion Mining with a Deep Convolutional Neural Network, Knowledge-Based Systems 108 (2016) 42–49.

[3] S. R. Das, M. Y. Chen, Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web, Management Science 53 (2007) 1375–1388.

[4] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, SemEval-2014 Task 4: Aspect Based Sentiment Analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), ACL, 2014, pp. 27–35.

[5] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2015 Task 12: Aspect Based Sentiment Analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), ACL, 2015, pp. 486–495.

[6] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, G. Eryiğit, SemEval-2016 Task 5: Aspect Based Sentiment Analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), ACL, 2016, pp. 19–30.

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

[8] D. Meškelė, F. Frasincar, ALDONA: A Hybrid Solution for Sentence-Level Aspect-Based Sentiment Analysis using a Lexicalised Domain Ontology and a Neural Attention Model, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC 2019), ACM, 2019, pp. 2489–2496.

[9] E. Cambria, Affective Computing and Sentiment Analysis, IEEE Intelligent Systems 31 (2016) 102–107.

[10] K. Schouten, F. Frasincar, Ontology-Driven Sentiment Analysis of Product and Service Aspects, in: 15th Extended Semantic Web Conference (ESWC 2018), volume 10360 of *LNCS*, Springer International Publishing, 2018, pp. 608–623.

[11] K. Schouten, F. Frasincar, F. de Jong, Ontology-Enhanced Aspect-Based Sentiment Analysis, in: 17th International Conference on Web Engineering (ICWE 2017), volume 10360 of *LNCS*, Springer International Publishing, 2017, pp. 302–320.

[12] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent Attention Network on Memory for Aspect Sentiment Analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), ACL, 2017, pp. 452–461.

[13] Q. Liu, H. Zhang, Y. Zeng, Z. Huang, Z. Wu, Content Attention Model for Aspect Based Sentiment Analysis, in: Proceedings of the 2018 World Wide Web Conference (WWW 2018), IW3C2, 2018, pp. 1023–1032.

[14] S. Jebbara, P. Cimiano, Aspect-Based Sentiment Analysis Using a Two-Step Neural Network Architecture, in: Semantic Web Challenges. Third SemWebEval Challenge at ESWC 2016. Revised Selected Papers, volume 641, Springer International Publishing, 2016, pp. 153–170.

[15] M. Zhang, Y. Zhang, D.-T. Vo, Gated Neural Networks for Targeted Sentiment Analysis, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016), AAAI Press, 2016, pp. 3087–3093.

[16] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), ACL, 2014, pp. 1724–1734.

[17] J. Pennington, R. Socher, C. D. Manning, GloVe: Global Vectors for Word Representation, in: Proceedings of the 2014 Empirical Methods in Natural Language Processing (EMNLP 2014), ACL, 2014, pp. 1532–1543.

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805, 2018.

[19] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), 2015. arXiv:1412.6980.

[20] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015.

[21] A. Coden, D. Gruhl, N. Lewis, P. N. Mendes, M. Nagarajan, C. Ramakrishnan, S. Welch, Semantic Lexicon Expansion for Concept-Based Aspect-Aware Sentiment Analysis, in: Semantic Web Evaluation Challenge, volume 475 of *CCIS*, Springer International Publishing, 2014, pp. 34–40.

[22] H. Alani, S. Kim, D. E. Millard, M. J. Weal, W. Hall, P. H. Lewis, N. R. Shadbolt, Automatic Ontology-Based Knowledge Extraction from Web Documents, IEEE Intelligent Systems 18 (2003) 14–21.

[23] T. R. Gruber, Toward Principles for the Design of Ontologies Used for

Knowledge Sharing, International Journal of Human-Computer Studies 43 (1995) 907–928.

[24] Z. S. Harris, Distributional Structure, WORD 10 (1954) 146–162.

[25] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment Classification using Machine Learning Techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), volume 10, ACL, 2002, pp. 79–86.

[26] G. Paltoglou, M. Thelwall, More than Bag-of-Words: Sentence-based Document Representation for Sentiment Analysis, in: Proceedings of Recent Advances in Natural Language Processing (RANLP 2013), ACL, 2013, pp. 546–552.

[27] Q. Le, T. Mikolov, Distributed Representations of Sentences and Documents, in: Proceedings of the 31st International Conference on Machine Learning (ICML 2014), volume 32, JMLR: W&CP, 2014, pp. 1188–1196.

[28] Y. Kim, C. Denton, L. Hoang, A. M. Rush, Structured Attention Networks, in: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), volume arXiv/1702.00887, 2017.

[29] C. Yang, H. Zhang, B. Jiang, K. Li, Aspect-based sentiment analysis with alternating coattention networks, Information Processing & Management 56 (2019) 463–478.

[30] D. Tang, B. Qin, T. Liu, Aspect Level Sentiment Classification with Deep Memory Network, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), ACL, 2016, pp. 214–224.

[31] S. Ruder, P. Ghaffari, J. G. Breslin, A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), ACL, 2016, pp. 999–1005.

[32] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for Target-Dependent Sentiment Classification, in: Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016), ACL, 2016, pp. 3298–3307.

[33] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for Aspect-level Sentiment Classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), ACL, 2016, pp. 606–615.

[34] D. Ma, S. Li, X. Zhang, H. Wang, Interactive Attention Networks for Aspect-level Sentiment Classification, in: Proceedings of the 26th Inter-

national Joint Conference on Artificial Intelligence (IJCAI 2017), AAAI Press, 2017, pp. 4068–4074.

[35] A. Hogenboom, P. van Iterson, B. Heerschop, F. Frasincar, U. Kaymak, Determining negation scope and strength in sentiment analysis, in: IEEE International Conference on Systems, Man, and Cybernetics 2011 (SMC 2011), IEEE SMC Society, 2011, pp. 2589–2594.

[36] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv abs/1609.08144 (2016).

[37] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (1997) 1735–1780.

[38] A. V. Olgaç, B. Karlik, Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks, International Journal of Artificial Intelligence and Expert Systems 1 (2011) 111–122.

[39] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, Y. Goldberg, Fine-grained Analysis of Sentence Embeddings Using Auxilary Prediction Tasks, CoRR abs/1608.04207 (2016). `arXiv:1608.04207`.