A Deep Learning Approach to Classify Aspect-Level Sentiment using Small Datasets

João Paulo Aires*, Carlos Padilha†, Christian Quevedo‡, Felipe Meneguzzi‡
Polytechnic School
Pontifical Catholic University of Rio Grande do Sul
Av. Ipiranga, 6681, 90619-900, Porto Alegre, RS, Brazil
* Email: joao.aires.001@acad.pucrs.br
† Email: cpadilha@motorola.com
‡ Email: {christian.quevedo, felipe.meneguzzi}@pucrs.br

***Abstract*—Sentiment analysis is an important technique to interpret user opinion on products from text, for example, as shared in social media. Recent approaches using deep learning can accurately extract overall sentiment from large datasets. However, extracting sentiment from specific aspects of a product with small training datasets remains a challenge. The automatic classification of sentiments at aspect-level can provide more detailed feedbacks about product and service opinions avoiding manual verification. In this work, we develop two deep learning approaches to classify sentiment at aspect-level using small datasets.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Measuring how people feel about a certain product or service gives a notion of the current product success and is a predictor of future business [1]. Social media and shopping websites collect a large amount of opinions about personal experiences in their purchases. To understand the information obtained from these sources, one needs to employ automated methods that read and classify the expressed sentiment as good or bad, a task known as sentiment analysis. Sentiment analysis is the process of identifying and classifying customer reviews by attributing to it a degree of positiveness or negativeness, with one of the simplest ways of representing such degree as one out of three classes: positive, negative, and neutral [2]. Aggregating these classifications provides important insights over the success or failure of a product and helps product designers improve their products by identifying those with substantial negative reviews or just identifying products whose reviews are mostly positive.

Sentiment analysis can be performed at roughly three levels: document, sentence, and aspect [3]. Document and sentence-level analysis provide the overall sentiment about these respective portions of text, whereas aspect-level analysis consists of identifying and classifying the sentiment of *semantically distinct* and identifiable aspects within the text. Thus, aspect-level sentiment analysis is a challenging task that often requires natural language processing to identify the portion of text in which the aspect is located and then classify its sentiment. Although there are many approaches to classify aspect-level reviews for the English language [4]–[6], other languages such as Portuguese have a comparatively smaller number of approaches capable of identifying sentiment on texts [7]–[9]; as well as fewer still publicly available datasets. In this paper, we develop an approach to deal with aspect-level sentiment analysis using small datasets. Our main contribution in this paper is a set of novel approaches for aspect-level sentiment analysis, including the use of deep learning approaches for both identification and classification of aspects and a preprocessing step that boosts the results for aspect identification. We use two datasets with Portuguese opinions annotated at document and aspect-level to train and test our approaches, and use an English dataset to gauge our approach for larger datasets.

II. Background

In what follows, we briefly summarize aspect-level sentiment analysis and proceed to review applicable deep learning techniques for text processing.

A. Aspect-level Sentiment Analysis

We adopt the formalism of Liu [1] and represent the sentiment analysis problem as a quadruple (g, s, h, t), where g is the sentiment target (a document, sentence, or aspect), s the sentiment of the target, h is the holder, which is the one who expresses the sentiment, and t is the time in which the sentiment is expressed. Most research in the area focuses on the tuple (g, s), which is only the sentiment classification of a target [10] since using only this information one can extract the user opinion. The process of sentiment analysis involves inferring the sentiment expressed in texts as either positive negative or neutral. Companies started to use sentiment analysis in social media to monitor customer sentiment over products [11]. The notion of sentiment from customers allows companies to understand the impact of their products.

Sentiment analysis can be divided into various levels, depending on the granularity of the textual target being analyzed, including document, sentence, and aspect-level analysis [10], [11]. In document-level sentiment analysis, the task consists of identifying the sentiment over a large portion of text (e.g. , a paragraph or an entire document), which allows the identification of an overall sentiment of the text. In the sentence-level analysis, the task is to identify the sentiment of each sentence in the text, allowing one to reason about short texts usually found in social media with a limited number of characters, such as Twitter. Finally, aspect-level analysis classifies the sentiment of specific *aspects* in the text. An aspect is a semantically distinctive characteristic about which one may have an opinion, for example, product features such as a phone's screen or battery capacity, specific elements of a hotel room such as its bed or bathroom, among others. The analysis of such aspects can provide fine-grained information about the interlocutor's opinion of the target. Consider the following cellphone review:

Example 1. 1: I'm very satisfied with the XYZ phone, the **camera** is great, **battery** lasts a long time but the **screen** is too small.

In the example 1, we have three main aspects, and each one has an individual sentiment. Whereas the text conveys a positive sentiment for **camera** and **battery**, it conveys a negative sentiment about the **screen**.

B. Deep Learning Approaches

Deep learning is a branch of machine learning that tries to solve learning tasks using deep neural networks (DNNs), i.e.; it relies on the power of a vast number of layers to extract features from the input data. The most common way to understand deep neural networks (DNNs) structure is to think of a regular neural network with a large number of hidden layers following a hierarchy for feature extraction from the input. In such hierarchy, the first layers extract abstract features whereas the last ones have specific ones. The premise for such number of layers is that as layers become deeper, richer information can be exploited from the input, which helps in the classification [12]. The most remarkable characteristic of DNNs is that they can extract features from raw data, i.e., without the need of previous feature selection [12]. This automatic extraction of features allows the algorithm to choose the best ones for each problem during training. Deep learning has been widely used for different tasks involving natural language processing, such as machine translation [13] and part-of-speech tagging [14]. Current approaches using deep learning to solve sentiment-analysis problems involve the use of both convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Such networks are able to extract meaningful information from the input boosting text classification.

1) Convolutional Neural Networks: Convolutional neural networks (CNNs) are a type of deep neural network that has one or more layers performing a convolution operation. CNNs are widely used to process images and videos as they are designed to receive multidimensional arrays as input, bidimensional for images and tridimensional for videos [12]. In a CNN, in the first layers the raw input is processed by convolutional layers Convolutional layers (see Figure 1) apply a series of filters over the input generating feature maps. Each filter goes through the entire input multiplying its weights by the input values, and the result is passed to a nonlinear activation function. The feature maps generated by the filters highlight different parts of the input.

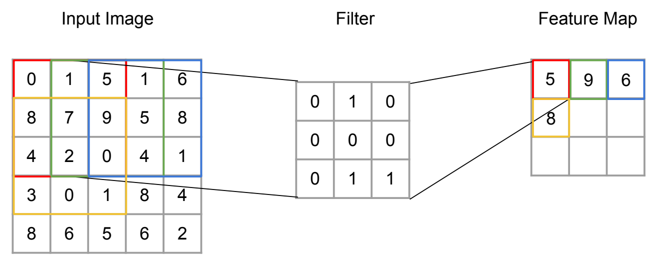

Fig. 1. Convolution example

Although CNNs are widely employed in image processing, recent approaches start to apply them in text. Two well-known approaches use CNNs to classify sentiment for sentence-level analysis, namely Zhang and LeCun [15] and Kim [16]. Zhang and LeCun propose an approach to use CNNs to solve a set of natural language processing tasks, such as sentiment analysis and text categorization. In order to process text in a CNN, a pre-processing step converts a string of characters into a matrix representation. The matrix structure uses the sentence characters as rows and the alphabet letters as columns; Then, they fill in the matrix with one (1) where cells have the same letters in row and column and zero (0) otherwise. Figure 2 exemplifies the output matrix from Zhang and LeCun approach. Zhang and LeCun's CNN extracts information from the relation between characters and their sequences, which enables the classifier to overcome typos and slangs typically found in online reviews.

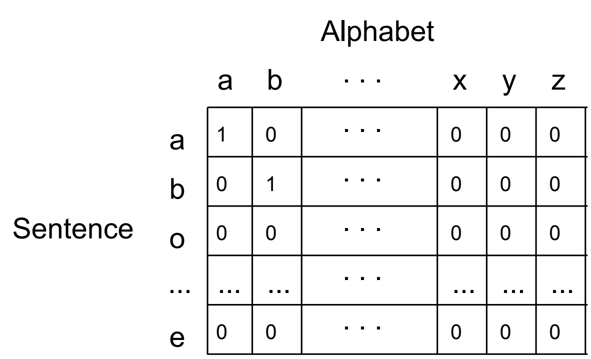

Fig. 2. Sentence to matrix representation from Zhang and LeCun [15]

Kim [16] develops a CNN to analyze sentiment at the sentence-level. This approach employs a matrix representation of the text that uses the sentence words as rows. Each row in the matrix is a word embedding, i.e., a dense representation from the word. Thus, the matrix has a dimension of $n \times m$, where n is the number of words in the sentence and m is the length of the word embedding. The author tested different embeddings for the words and found the best results using

Word2Vec [17]. Figure 3 illustrates the resulting matrix. Different from Zhang and LeCun CNN, Kim's CNN extracts features from the relation between words in a sentence. With the help of the word embeddings, he obtains semantic information to classify the sentiment from texts. He tests and compares his CNN using a series of existing datasets concerning sentiment analysis and general text classification obtaining new state of the art.

Word embedding

Sentence							
	Really	0.9	0.5	...	0.6	0.1	0.2
	loved	0.1	0.2	...	0.5	0.5	0.6
	the	0.1	0.3	...	0.2	0.9	0.1
	new	0.7	0.3	...	0.4	0.1	0.2
	feature	0.6	0.4	...	0.8	0.3	0.7

Fig. 3. Sentence to matrix conversion performed by Kim [16]

2) Recurrent Neural Networks: (RNNs) are a type of DNN that deals with sequences of inputs, which makes them useful to deal with speech and language problems. An RNN processes an input sequence one element at time and it keeps a history with information about previous inputs in its hidden units [12]. The architecture of an RNN may vary from fully interconnected to partially connected neural nets. In fully interconnected, the input of each node is the output of all others and the network input can be any node. In partially connected networks, nodes follow a similar structure of an ANN but some of them have more than one input and output [18]. As RNNs preserve information from previous nodes, they can use it to predict the next input from the context of previous inputs. Figure 4 illustrates a node in an RNN in which its output is both a predicted class to the input and the input of the next node. When dealing with long sequences, common RNNs have problems on predicting labels that need information from a long past. To solve such problem, Hochreiter and Schmidhuber [19] introduced the Long Short-Term Memory (LSTM) network, a type of RNN that manages the internal information with a series of memory mechanisms. The main difference between a standard RNN and an LSTM is that the latter has a cell state that goes through the entire network. In this cell, LSTM adds and deletes new information according to the importance of it. Such mechanism ensures that the information needed to classify the sequence travels through the network giving context to further decisions.

III. Aspect-Level Sentiment Analysis Method

Performing aspect-level sentiment analysis involves the identification of a certain aspect target in the text and then its classification. This double-step process triggers two subprocesses: the correct identification of an aspect and the definition of the correct amount of text that must be used to classify

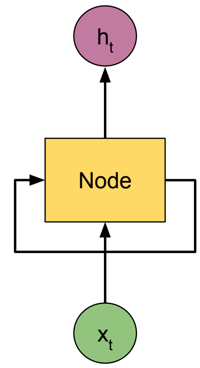

Fig. 4. Example of RNN with a loop.

its sentiment. Since we are dealing with natural language, an aspect may be written in many ways making it difficult to identify it. For example, "camera" may appear in a text as 'frontal', 'photo', 'video', among others. Besides aspect identification, defining the stretch of text we must use to classify the aspect is also a challenging task since there are multiple combinations of words around the aspect we can use to classify the aspect. In order to avoid the creation of rules to identify aspects as well as which parts of the text to consider in the sentiment classification, we propose two architectures each containing two modules that automatically identify and classify aspects. In our first architecture (SA I), we train a model by aspect to make the identification of aspects in the sentences of a document. When an aspect is identified, we pass the sentence as input for a second model that makes an overall classification of the sentence. The resulting sentiment is considered the aspect sentiment. Figure 5 illustrates the SA I architecture in which the input review is broken into sentences and each sentence becomes input to the models of the first module (Aspect Identifier). Finally, the sentences identified as containing an aspect become input to the second module (Aspect Classification) that classify the aspect sentiment.

Our second architecture (SA II) has a structure similar to the first one consisting of two modules. Their main difference is in the classification module, which in SA II has separately-trained models for each aspect, making the classification strictly specific to the selected aspects. Thus, in SA II we have the same number of identifier and classifier models. Figure 6 illustrates the SA II architecture in which the input review is preprocessed in order to obtain meaningful words from it. These words become input to the first module (Aspect Identifier) and when an aspect is identified, the input goes directly to the second module (Aspect Classifier) where the sentiment of each aspect is classified.

A. Aspects

Since identifying sentiment at aspect-level involves the specific definition of aspect targets, we selected 10 cellphone-related aspects to make the sentiment classification. This selection takes into consideration the presence of such aspects

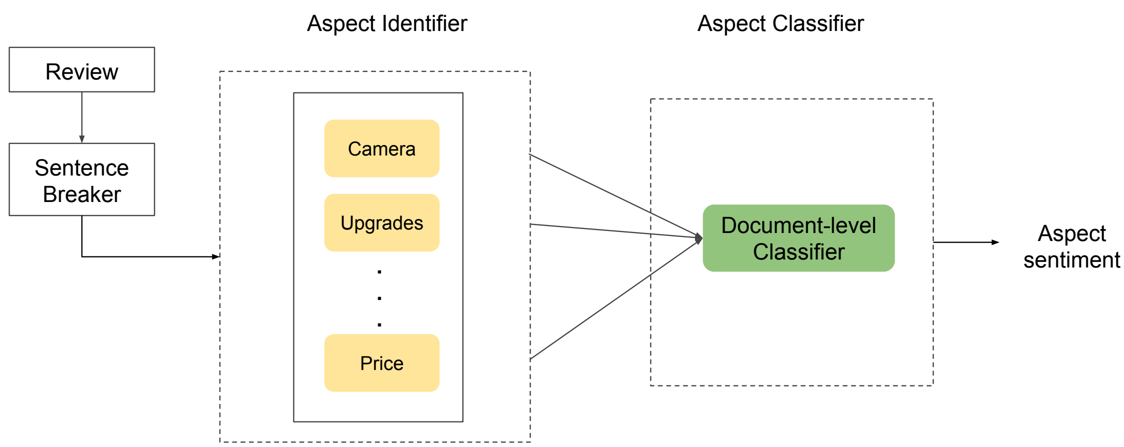

Fig. 5. SA I architecture

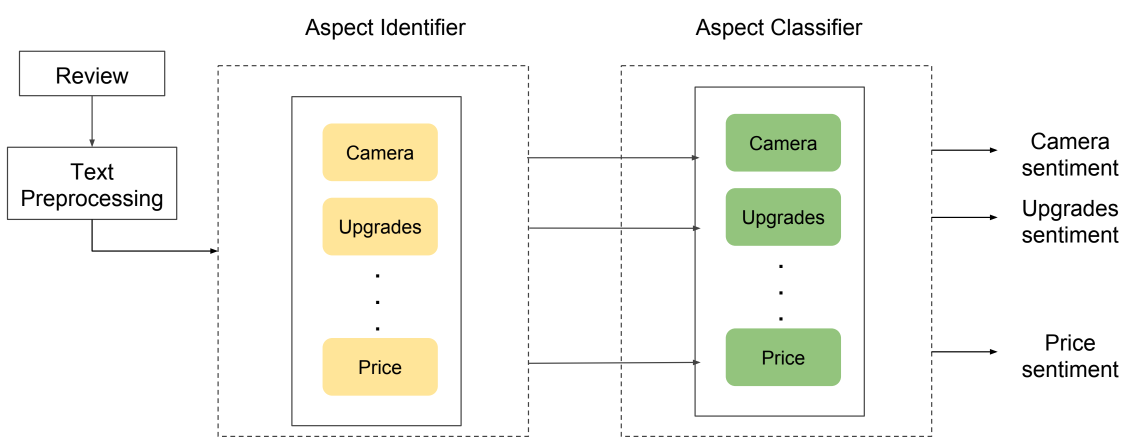

Fig. 6. SA II architecture

in our dataset, and these aspects are: battery, camera, design, display, memory, price, processor, sales and services, temperature, and upgrades.

B. Dataset

We used multiple manually-annotated datasets to train and test our models. These datasets are the result of our effort to create a Portuguese dataset for document and aspect-level sentiment classification. The first dataset (DS I) consists of 32,000 reviews from different online stores annotated with overall sentiment at the document level, as well as the sentiment for individual aspects present in the review. The process of data gathering involved the creation of a web application in which volunteers annotate the sentiment of reviews at document-level. Besides classifying the document-level, we asked volunteers to classify the aspect-level if it exists. In order to facilitate and speed up the job of annotators in identifying aspects, we used a simple rule-based mechanism to identify candidate of aspects in the text, with text-matching rules for aspect names and some of their syntactic variations. Thus, the annotation interface offers the human annotator a suggestion consisting of the aspects identified by the rule and asks the annotator to assign a sentiment to it. If our rules generate a non-existent aspect, the volunteer can remove it, and if we fail at identifying some aspect, the volunteer is able to add it to the classification along with the corresponding sentiment. Finally, to obtain a reliable annotation over the reviews, we consider each volunteer annotation as a sentiment vote, and offer the annotator reviews semi-randomly with a bias for reviews with few annotations. Then, we assume that the winning sentiment for each review has a simple majority of at least two more votes than the other sentiments in document-level. Table I summarizes the aspects most often found and annotated collected in our dataset. We use a second dataset (DS II) containing 30,000 reviews with aspects annotated by a single professional within the company as the testing benchmark against which our approaches are compared. This dataset has a larger number of annotated aspects, as we can see in Table II.

TABLE I
MOST FOUND ASPECTS IN DS I

Aspect	# of reviews
Battery	1,000
Camera	3,700
Design	468
Display	719
Memory	286
Price	1,387
Processor	433
Sales and services	-
Temperature	23
Upgrades	8,498

TABLE II
MOST FOUND ASPECTS IN DS II

Aspect	# of reviews
Battery	855
Camera	7,400
Design	649
Display	460
Memory	358
Price	907
Processor	189
Sales and service	795
Temperature	106
Upgrades	2,533

C. Text Preprocessing

Both approaches preprocess the input text prior to its classification. In SA I, we use a sentence splitter to divide a document into sentences. We use three punctuation marks (.?!) as sentence breakers. As result, each sentence of a review is an individual input to the architecture. In SA II, we perform a series of NLP operations in order to do a data text cleaning on the user reviews, as follows:

1) **Removal of HTML tags**: since our data was obtained from online shopping websites, some reviews contain HTML entities such as paragraphs (<p>), line breaks (
), and so on.
2) **Translation of emojis to text**: in this step, we map each emoji in the text to one or more words that describe the sentiment expressed by the emoji, for example, as shown in Figure 7;

Fig. 7. Translating emojis to text

3) **Decoding data**: we convert all data into UTF-8 encoding to preserve a standard text format and avoid having unknown/undefined characters as input.
4) **Removal of special characters**: in this step, we remove accentuation and punctuation.
5) **Removal of non-letters**: we keep only the words present in the text, removing the numbers too.
6) **Word standardization**: words containing any character redundancy are fixed here, e.g "I looveeeed it" is fixed to "I loved it". This step allows us to decrease the number of features by unifying all redundancy words to a single one.
7) **Removal of stop-words**: here we remove all the stop-words, except the word "not" as this word strong affects the sentiment expressed in the text;

This process aims to leave only the meaningful words in the review.

D. Aspect Identification

The aspect identification step is part of both approaches but has small differences in its definition depending on the architecture. In SA I, we trained a Support Vector Machine (SVM) [20] classifier for each aspect resulting in 10 different classifiers using DS II. The input reviews are the entire document, which may contain additional aspects beyond the target aspects. We divide the dataset by aspect and performed a 10-fold cross-validation. We show the results of this training in the Identification column in the "SA I" column in Table IV. In the pipeline, we use the trained models to discover the aspects in a review. The model's input is the resulting sentences from the preprocessing step. Thus, when a sentence is classified as containing an aspect, we use it as input to the aspect classification step.

In SA II, we have a similar approach in which we have the same number of models as the aspects to perform the identification. In this case, we train an LSTM network with a single layer containing 100 nodes for 40 epochs. We show the results in the Identification column inside the SA II column in Table IV.

E. Aspect classification

Once the previous process identified the specific aspect from a piece of text, our architectures proceed to classify the sentiment polarity of the text with regards to the identified aspect. In SA I, we use a single classification model to perform the sentiment classification. We trained this model to perform document-level sentiment analysis. Using the data from DS I, which contains 32 thousand annotated reviews, we tested several model combinations. For each model, we performed a 10-fold cross validation, and trained them over 20 epochs. Table III shows a comparison between the models applied to binary classification (positive and negative classes only) and multi-class classification (positive, negative, and neutral classes) for the following models: an SVM; an LSTM with one layer containing 256 nodes; the CNN proposed by Zhang and LeCun [15] (Z CNN); the CNN proposed by Kim [16] with an LSTM layer containing 256 nodes (K CNN + LSTM); and a preprocessing step (described in Section III-C as the one used in SA II) followed by an LSTM with one layer containing 100 nodes (P + LSTM). We use the best model (P + LSTM) as our sentiment classifier, it receives a sentence selected from the aspect identification step and classifies attributing to it a positive, negative, or neutral sentiment.

Unlike SA I, the SA II classification process has an independently-trained model per aspect, thus, we have a total of 10 sentiment classifier models. The classifier model employed is the same as the identification process, i.e., an LSTM network with one layer containing 100 nodes trained over 40 epochs.

IV. Results

The comparison between SA I and SA II involves evaluating the two steps (identification and classification) in both architectures. We use DS II to compare the results and find which architecture has the best performance. Table IV shows the comparison between SA I and SA II for each aspect in our list of selected ones. We highlight the best results for both identification and classification between SA I and SA II in bold. As results show, SA II has a better performance over SA I in almost every aspect. Although the classification step in SA I is not restricted to a selected list of aspects, the results have shown accuracies lower than 70%, which means that the aspect-specialization in classification improves task performance. Results for both identification and classification in SA II indicate accuracies over 90%, which suggests that the use of singular LSTM networks for each aspect works better for the aspect-level classification task when compared to SA I.

In order to test our best approach over a larger dataset and in a different language, we use a third dataset with manually annotated aspects in English. This third dataset (DS III) contains a total of 616,976 reviews and we describe the distribution between aspects in Table V The results for both identification and classification are displayed in the column "SA II English" in Table IV. Sentiment analysis for this dataset has accuracy above 90%, indicating that our approach can be used in different contexts obtaining similar results (i.e. increased data availability did not improve results). x

V. Related Work

A number of approaches attempt to perform sentiment analysis of Portuguese opinions, and, in this section, we review the most recent ones, comparing them to our approach. Siqueira and Barros [7] propose an approach to perform domain-free feature extraction over user opinions. They use a four-step approach in which first they break the opinion into sentences and extract the POS tags from the text to find the most frequent nouns. Second, they select non-frequent nouns that can be relevant by analyzing the adjectives that transform them. Third, they map synonyms to a unique name in order to avoid redundant features. Finally, they remove unrelated nouns by their frequent among the candidates. The authors use a Brazilian Portuguese dataset containing 2,200 user opinions. They used 2,000 reviews to obtain knowledge to the proposed solution and validated over 200 opinions. Using the four steps, they obtain an f-measure of 83%. In our work, we propose an approach that performs both the identification and classification of aspects over user opinions. Although we limit the approach to a closed list of aspects, we have a minimum use of natural language processing during our process, which gives us an automatic extraction of features from data.

Freitas and Vieira [8] propose an approach to identify sentiment from Portuguese user reviews described in domain ontologies. They propose a method with four steps using a series of natural language processing techniques. First, they receive and preprocess a set of reviews by applying sentence splitters, tokenizers and lemmatizers. Second, the authors use an ontology to perform a feature identification over the preprocessed reviews. Third, they identify the polarity by using a dictionary with adjectives, verbs, and nouns and their corresponding polarities. Finally, they use windows of words and linguistic rules to generate an output with each feature in the review and its corresponding polarity. To test their approach, the authors use two datasets with different contexts, one containing 180 accommodation Portuguese reviews and the other with 150 reviews about hotels. They obtain an f-measure of 58% in the first dataset and 62% in the second one. In our work, we avoid using rules by applying deep neural networks to select what is most relevant among the review words.

Czech, like Portuguese, has few approaches to sentiment analysis. Steinberg *et al.* [21] propose an approach to classify aspect-level sentiment in Czech reviews. To train and test the approach, they created a dataset containing a total of 1,244 labeled Czech reviews at aspect-level. Using the dataset, the authors train a two-module approach with identification and classification modules. In the identification module, they use conditional random fields to detect aspects from a preprocessed review For the classification module, they train a model using a Maximum Entropy classifier. As input to the model, they use a window with 10 words from both sides of the target aspect assigning weights to the words according to the proximity. In the first module, they obtain an f-measure of 68%, while in the second module they obtain 66% of f-measure. We also use classifiers for both identifying and classifying aspects, however, we avoid relying on a window of words to classify the aspect. Instead, we use either the entire sentence or the entire document in which the aspect is identified.

For English, we have several approaches involving different approaches from rule-base to deep learning. Jo and Oh [4] introduce an approach to identify and classify aspects. They propose two models, the first one is an extension of the Latent Dirichlet Allocation (LDA) [22] algorithm named *Sentence-LDA*. This modification considers that words in a sentence are all about the same topic. The second model is an extension of first one named Aspect Sentiment Unification Model. In this model, they calculate a probability of a certain sentiment given an aspect. They apply the first model to the aspect discovery task and the second one to sentiment classification task. The authors use two datasets, the first one is a collection of electronic device reviews from Amazon (22,000 reviews) and the second is about restaurants (30,000 reviews). On the

TABLE III

Comparison between the resulting accuracies of document-level approaches divided by binary and multi-class sentiment classification

Classification	SVM	LSTM	Z CNN [15]	K CNN [16] + LSTM	P + LSTM
Binary	96.3%	96.35%	93.73%	96.65%	**98.3%**
Multi-class	91.52%	91.1%	93.65%	93.39%	**93.78%**

TABLE IV

Accuracy comparison between Approach 1 and 2 for the task for aspect-level classification

	SA I		SA II		SA II English	
Aspect	Identification	Classification	Identification	Classification	Identification	Classification
Battery	95%	80%	**99**%	**99**%	99%	93%
Camera	83%	88%	**99**%	**98**%	98%	89%
Design	96%	89%	**99**%	**99**%	99%	91%
Display	98%	60%	**99**%	**98**%	99%	90%
Memory	98%	84%	**99**%	**98**%	99%	94%
Price	95%	94%	**99**%	**99**%	98%	95%
Processor	**99**%	89%	**99**%	**100**%	99%	93%
Sales and Services	94%	72%	**99**%	**96**%	99%	92%
Temperature	**99**%	62%	**99**%	**98**%	99%	88%
Upgrades	89%	76%	**96**%	**93**%	99%	88%

TABLE V

Most found aspects in DS III

Aspect	# of reviews
Battery	95,370
Camera	94,725
Design	50,249
Display	47,698
Memory	40,502
Price	143,072
Processor	14,232
Sales and service	60,964
Temperature	33,563
Upgrades	36,601

sentiment classification task, they obtain an accuracy of 84% for the electronics dataset and 86% on the restaurants dataset. Different from our work, their approach is not limited to a set of aspects. However, we can achieve better results when we know the target aspects.

Poria *et al.* [5] introduce an approach to extract features and classify the sentiment from short texts using a deep convolutional network. They use a CNN similar to the one proposed by Kim [16] to extract features resulting from the convolutional layer. The authors use the features as input to an SVM or multi-kernel learning. To train and test their approach, they use a dataset with 498 short video fragments in which a person utters one sentence. They obtain 79% of accuracy in a unimodal approach using only text and 85% using a bimodal approach, which considers both the text and video. The main difference between our approach and theirs is that in our work we have the goal of obtaining the aspect-level sentiment while they classify document-level using deep neural networks.

VI. Conclusion

In this paper, we introduce two architectures to classify the sentiment of aspects in user opinions. We divide both architectures into two modules, the first one identifies the aspect in the text and the second step classifies the sentiment of the identified aspect. In the first architecture, we classify the aspect sentiment using a document-level model that receives a sentence as input. This architecture can be used when we have too little data to train a specific aspect. In the second architecture, we apply a preprocessing step over the input and classify the aspect sentiment training a model for each aspect. As our results show, the second architecture achieves higher accuracies when compared to the first one, which confirms that using a preprocessing step plus a model by aspect improves the results.

As future work, we aim to refine the type of information aspect-level sentiment analysis is able to provide. Specifically, we aim to develop new architectures to discover aspects dynamically in the text making it possible to process aspect-level sentiment without a previous fixed list of aspects.

Acknowledgments

The authors would like to acknowledge Motorola Mobility for funding this research. Felipe thanks CNPq for partial financial support under its PQ fellowship, grant number 305969/2016-1.

References

[1] B. Liu, *Sentiment Analysis and Opinion Mining*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012. [Online]. Available: https://doi.org/10.2200/S00416ED1V01Y201204HLT016

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008. [Online]. Available: http://dx.doi.org/10.1561/1500000011

[3] V. K. Singh, R. Piryani, A. Uddin, and P. Waila, "Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification," in *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, March 2013, pp. 712–717.

[4] Y. Jo and A. H. Oh, "Aspect and sentiment unification model for online review analysis," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011, pp. 815–824. [Online]. Available: http://doi.acm.org/10.1145/1935826.1935932

[5] S. Poria, E. Cambria, and A. F. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015, pp. 2539–2544. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1303.pdf

[6] V. Parkhe and B. Biswas, "Sentiment analysis of movie reviews: finding most important movie aspects using driving factors," *Soft Computing*, vol. 20, no. 9, pp. 3373–3379, Sep 2016. [Online]. Available: https://doi.org/10.1007/s00500-015-1779-1

[7] H. Siqueira and F. Barros, "A feature extraction process for sentiment analysis of opinions on services," in *Proceedings of International Workshop on Web and Text Intelligence*, 2010, pp. 404–413.

[8] L. A. Freitas and R. Vieira, "Ontology-based feature level opinion mining for portuguese reviews," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13 Companion. New York, NY, USA: ACM, 2013, pp. 367–370. [Online]. Available: http://doi.acm.org/10.1145/2487788.2487944

[9] M. S. Chaves, L. A. de Freitas, M. Souza, and R. Vieira, *PIRPO: An Algorithm to Deal with Polarity in Portuguese Online Reviews from the Accommodation Sector*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 296–301.

[10] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, March 2016.

[11] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, March 2013.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] J. Zhang and C. Zong, "Deep neural networks in machine translation: An overview," *IEEE Intelligent Systems*, vol. 30, no. 5, pp. 16–25, 2015. [Online]. Available: https://doi.org/10.1109/MIS.2015.69

[14] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 160–167. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390177

[15] X. Zhang and L. Cun, "Text understanding from scratch," *CoRR*, vol. abs/1502.01710, 2015.

[16] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1746–1751.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[18] L. C. Jain and L. R. Medsker, *Recurrent Neural Networks: Design and Applications*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1999.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[20] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, July 1998.

[21] J. Steinberger, T. Brychcín, and M. Konkol, "Aspect-level sentiment analysis in czech." in *WASSA@ ACL*, 2014, pp. 24–30.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003. [Online]. Available: http://www.jmlr.org/papers/v3/blei03a.html