

Simple Unsupervised Similarity-Based Aspect Extraction

Danny Suarez Vargas, Lucas R. C. Pessutto, and Viviane Pereira Moreira

Federal University of Rio Grande do Sul – UFRGS, Porto Alegre RS, Brazil
<http://www.inf.ufrgs.br/>
{dsvargas,lrcpessutto,viviane}@inf.ufrgs.br

Abstract. In the context of sentiment analysis, there has been growing interest in performing a finer granularity analysis focusing on the specific aspects of the entities being evaluated. This is the goal of Aspect-Based Sentiment Analysis (ABSA) which basically involves two tasks: aspect extraction and polarity detection. The first task is responsible for discovering the aspects mentioned in the review text and the second task assigns a sentiment orientation (positive, negative, or neutral) to that aspect. Currently, the state-of-the-art in ABSA consists of the application of deep learning methods such as recurrent, convolutional and attention neural networks. The limitation of these techniques is that they require a lot of training data and are computationally expensive. In this paper, we propose a simple approach called SUAEX for aspect extraction. SUAEX is unsupervised and relies solely on the similarity of word embeddings. Experimental results on datasets from three different domains have shown that SUAEX achieves results that can outperform the state-of-the-art attention-based approach at a fraction of the time.

1 Introduction

Opinionated texts are abundant on the Web and its study has drawn a lot of attention from both companies and academics originating the research field known as opinion mining or sentiment analysis. The last few years have been very prolific in this field which combines Natural Language Processing and Data Mining. In a study which dates back to 2013, Feldman [1] mentioned that over 7,000 articles had already been written about the topic. Several facets of the problem have been explored and numerous solutions have been proposed.

While most of the work in the area has been devoted to assigning a polarity score (positive, negative, or neutral) to the overall sentiment conveyed by the text of an entire review, in the last few years, there has been increasing interest in performing a finer-grained analysis. Such analysis, known as *Aspect-Based Sentiment Analysis* (ABSA) [2] deals basically with the texts of extracting and scoring the opinion expressed towards an entity. For example, in the sentence “*The decor of the restaurant is amazing and the food was incredible*”, the words *decor* and *food* are the aspects of the entity (or category) *restaurant*.

ABSA is a challenging task because it needs to accurately extract and rate fine-grained information from reviews. Review texts can be ambiguous and con-

tain acronyms, slangs and misspellings. Furthermore, aspects vary from one domain to another – one word that represents a valid aspect in one domain, may not do so for another domain. For example, consider the input sentence “*The decor of the place is really beautiful, my sister loved it.*” in the *Restaurant* domain. The ABSA solution should focus its attention on the words “*decor*” and “*restaurant*”. However, if we do not explicitly set *Restaurant* as the domain, the word “*sister*” can also gain attention as an aspect term. This poses a potential problem to extraction approaches. For example, an approach that is only based on rules [3] can assume that aspects are always nouns and come exactly after the adjective. However, rigid rules can create false positives because not always a noun represents an aspect. Another approach could consider the distribution of words in texts and hypothesize that the number of occurrences of a given word is determinant to decide whether it is an aspect. Again, this could lead to false positives since high-frequency words tend to be stopwords.

A disadvantage of current state-of-the-art approaches [4–7] is that they rely on techniques that require significant computational power, such as deep neural networks. In special, neural attention mechanisms [8] are typically expensive.

In this paper, we propose SUAEX, a simple unsupervised method for aspect extraction. SUAEX relies on vector similarity to emulate the attention mechanism which that allows us to focus on the relevant information. Our main contribution is to show that a simple and inexpensive solution can perform as well as the neural attention mechanisms. We tested SUAEX on datasets from different domains, and it was able to outperform the state-of-the-art in ABSA in many cases in terms of quality and in all cases in terms of time.

2 Background and Definitions

Word-Embeddings. Representing words in a vector space is widely used as a means to map the semantic similarity between them. The underlying concept is the hypothesis that words with similar meanings are used in similar contexts. Word embeddings are a low dimensional vector representation of words which is able to keep the distributional similarity between them. Furthermore, word embeddings are able to map some linguistic regularities present in documents. Since the original proposal by Mikolov *et al.* [9], other techniques have been presented [10, 11] adding to the popularity of word embeddings.

Category and Aspect. The terms *category* and *aspect* are defined as follows. For a given sentence $S = w_1, w_2, \dots, w_n$ taken from the text of a review, the category C is the broad, general topic of S , while the aspects are the attributes or characteristics of C [12]. In other words, a category (*i.e.*, laptop, restaurant, cell phone) can be treated as a cluster of related aspects (*i.e.*, memory, battery, and processor are aspects that characterize the category *laptop*).

Reference Words are important in the context of our proposal because they aid in the correct extraction of aspects (*i.e.*, distinguishing aspects from non-aspects), and help determine whether an aspect belongs to a category. A reference word can be an aspect, or the name of the category itself (a synonym, meronym, or hyponym). For example, if we want to discover aspects from the

category “laptop”, the words “computer”, “pc” and the word “laptop” itself would be reference words.

Attention Mechanism. The attention mechanism was introduced in a neural machine translation solution [8]. The main idea was to modify the encoder-decoder structure in order to improve the performance for long sentences. For a given input sentence $S = \{w_1, w_2, \dots, w_n\}$, the encoded value e_s of S and a set of hidden layers $H = \{h_1, h_2, \dots, h_m\}$, the decoder for each output y_i considers not only the value of the previous hidden layer h_{i-1} and a general context c , but it also considers the relative importance of each output word y_i with respect to the input sentence S . For example, for a given output word y_j , it can be more important to see the words w_2, w_3 in S , while for another output word y_k , it can be more important to see only the word w_4 in S . The attribution of the relative importance is performed by an attention mechanism.

3 Related Work

Sentiment Analysis can be performed at different levels of granularity. One could be interested in a coarse-grained analysis which assigns a sentiment polarity to an entire review document (*i.e.*, document level analysis); or to each sentence of the review; or, at a finer granularity, to the individual aspects of an entity.

The aspect level is quickly gaining importance, mainly due to the relevant information that it conveys [12]. In this level of analysis, the aspects and entities are identified in natural language texts. Aspect extraction task can be classified into three main groups according to the underlying approach [2]: (*i*) based on language rules [3,13,14], (*ii*) based on sequence labeling models [15,16], and (*iii*) based on topic models [17]. However, other works do not fit in only one of these groups as they combine resources from more than one approach [18]. Furthermore, state-of-the-art approaches rely on more sophisticated architectures like recurrent neural networks such as LSTM, Bi-LSTM, Neural Attention Models, and Convolutional Neural Networks [5–7,19,20].

The work proposed by He *et al.* [19], known as Attention-based Aspect Extraction (ABAE), represents the state-of-the-art in ABSA and was used as the baseline of our work. ABAE relies on an attention neural network to highlight the most important words in a given text by de-emphasizing the irrelevant words. ABAE is a three-layer neural network. The input layer receives a given sentence $S = \{w_1, w_2, \dots, w_n\}$, S is represented as a set of fixed length vectors $e = \{e_w1, e_w2, \dots, e_wn\}$. These vectors are processed by the hidden layer setting the attention values $a = \{a_1, a_2, \dots, a_n\}$ related to a given context y_s . The context y_s is obtained from the average of the word vectors in e . After performing the attention mechanism, the input sentence is encoded as z_s and a dimensionality reduction is performed from the word-embedding space to the aspect-embedding space. In other words, the input sentence is represented only by the most relevant words in r_s . In addition, the process of training a neural network needs an optimization function. The output layer is the sentence reconstruction r_s and the function to optimize aims to maximize the similarity between z_s and r_s . Finally, the mathematical definition of ABAE is the following:

$$\begin{aligned}
r_s &= T^T \cdot p_t & p_t &= \text{softmax}(W \cdot z_s + b) \\
z_s &= \sum_{i=1}^n a_i e_{w_i} & a_i &= \frac{\exp(d_i)}{\sum_{j=1}^n \exp(d_j)} \\
d_i &= e_{w_i}^T M y_s & y_s &= \frac{1}{n} \sum_{i=1}^n e_{w_i} \\
J(\theta) &= \sum_{s \in D} \sum_{i=1}^m \max(0, 1 - r_s z_s + r_s n_i)
\end{aligned}$$

where z_s encodes the input sentence S by considering the relevance of its words. a_i is the relevance of the i^{th} word in S . d_i is the value that expresses the importance of the i^{th} word related to the context y_s . Finally, $J(\theta)$ is the objective function which is optimized in the training process.

In summary, each group of solutions for ABSA have advantages and disadvantages. The methods based on language rules are simple but require manual annotation to construct the initial set of rules. Furthermore, these rules are domain-specific – a new set of rules is needed for each domain. The methods based on sequence models, topic models and even some based on neural networks are supervised machine learning solutions. So, their quality is directly proportional to the amount of annotated data. Finally, methods based on unsupervised neural networks, such as our baseline [19], achieve good results but at a high computational cost.

4 Simple Unsupervised Aspect Extraction

This section introduces SUAEx, a simple unsupervised similarity-based solution for ABSA. SUAEx relies on the similarity of vector representations of words to emulate the attention mechanism used in the state-of-the-art. Since our proposed solution does not need to train a neural network, SUAEx is computationally cheaper than state-of-the-art solutions and, as demonstrated in Section 6, it achieves results that can surpass the baseline at a fraction of the time.

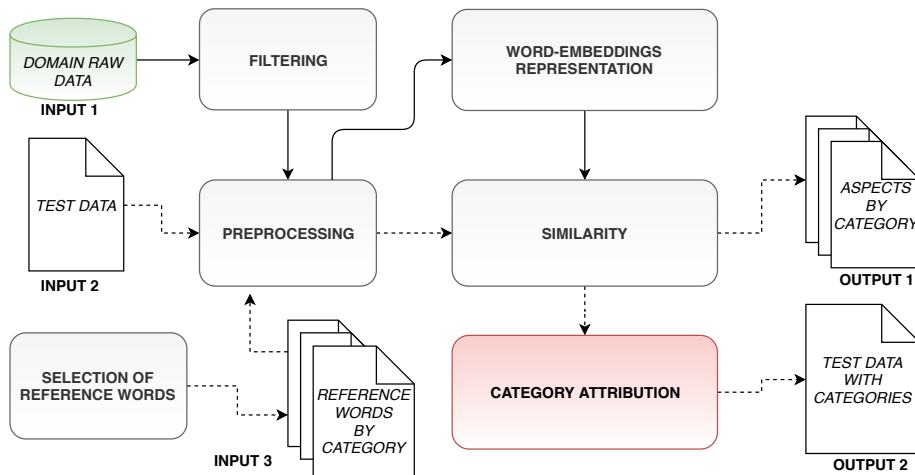


Fig. 1. SUAEx framework. The continuous arrows represent the path taken by Input 1, while the dashed arrows represent the path followed by Inputs 2 and 3.

SUAEx¹ consists in six modules depicted in Fig. 1: (i) Filtering, (ii) Selection of Reference Words, (iii) Preprocessing, (iv) Word-Embeddings Representation, (v) Similarity, and (vi) Category Attribution. SUAEx requires three inputs and generates two outputs. The inputs are: **Input1** – the *raw data* expressed as free text from a given domain (which is used to build the domain-specific word embeddings); **Input2** – the *test data* with the reviews for which the aspects and categories will be extracted; and **Input3** – the *reference words*, which are used to determine the categories as well as extract the aspects related to each category. Next, we describe the components of SUAEx.

The **Selection of Reference Words** module is responsible for choosing the representative words for each category. In other words, if we want k categories as output, we need to select k groups of reference words. The selection of the words for each group can be performed in three different ways: manual, semi-automatic, and automatic. The manual selection can be done by simply selecting the category words themselves as reference words. The semi-automatic selection can be performed by expanding the initial manually constructed groups of reference words. The expansion can be done through the search for synonyms or meronyms of the words that represent the category name. Finally, an automatic selection mechanism can be performed by considering a taxonomy of objects [21].

The **Filtering** module aims to select the domain related part from raw data. This module is optional but it is particularly useful when we want to delve into a certain category and we only have raw data for the general topic. For example, if we just performed the aspect extraction for the category “Electronics”, we have the raw data for it. If now, we want to perform the aspect extraction of the category “Laptop”, we need only raw data about this new more specific category. This module is in charge of selecting the right raw data from a large corpus, thus manual filtering is unfeasible. Filtering can be implemented as a binary text classifier for the domain of interest, or simply by choosing reviews that mention the category name.

The **Preprocessing** module normalizes the input data and reduces the amount of raw data needed to construct the vector representation of words (word-embeddings). The amount of data needed to train word-embeddings is directly proportional to the size of the vocabulary of the raw data. Since preprocessing reduces the size of the vocabulary, it has the effect of reducing the amount of raw data needed. This module encompasses typical preprocessing tasks such as tokenization, sentence splitting, lemmatization, and stemming.

The **Word-Embedding Representation** module is responsible for creating the vector representation of words which will be used to measure word similarity in the Similarity module. It receives the preprocessed raw data, transforms it into a vector representation and returns a domain-specific model. The model can be generated using well-known tools such as Word2vec [9], Glove [10], or Fastext [11]. This module is particularly important because SUAEx relies solely on the similarity of domain-specific word-embeddings.

¹ The code for SUAEx is available at <https://github.com/dannysvof/SUAEx.git>.

The **Similarity as an Attention Mechanism** module receives two types of inputs, the preprocessed reference words and the test data. The goal of this module is to emulate the behavior of the attention mechanism in a neural network by assigning attention values to each word in an input sentence in relation to a given set of reference words. For each group of reference words, (which are in the same number as the categories desired as output), it returns an attention-valued version of the test data. This output can be used in two ways: to identify the aspects for each category or as an input to the Category Attribution module. A vector similarity measure, like the cosine similarity, is used to attribute the relevance of a given word x in relation to another word y or related to a group of words c . In this module, we can test with two types of similarity values. The similarity obtained from the direct comparison of two words (direct similarity) and the similarity obtained from the comparison of two words in relation to some contextual words (contextual similarity). Finally, the attention values are obtained by applying the *softmax* function to the similarity values.

Output1 is the test data together with the values for attention and similarity assigned by SUAEX. For example, if we consider three groups of reference words in the input, "food", "staff", and "ambience", Output1 consists in three attention-valued and similarity-valued versions of the test data (one set of values for each category).

The **Category Attribution** module uses the output of the Similarity module to assign one of the desired categories to each sentence in the test data. In this module, we can test different ways to aggregate the similarity values assigned to the words. For example, one could use the average for each sentence or only consider the maximum value [22]. If the average is used, it means that there are more than one relevant word to receive attention in the sentence. However, if only the maximum value is used the word with the highest score will get all the attention.

Output2 is the main output which contains all the sentences of the test data with the categories assigned by the Category Attribution module.

5 Experimental Design

The experiments of this aim at evaluating SUAEX for aspect and category extraction both in terms of quality and runtime. Our tests are done over datasets coming from three different domains.

The experiments are organized in two parts. In the first experiment, the goal is to compare SUAEX to our baseline ABAE [19]. We hope to answer the following questions: *Can a simple approach like SUAEX achieve results that are close to the state-of-the-art in aspect extraction?* and *How does SUAEX behave in different domains?* The second experiment performs a runtime analysis of the two approaches. Below, we describe the datasets, the tools and resources used.

Datasets. The datasets used in our experiments are summarized in Table 1 and come from different sources and different domains to enable a broad evaluation. They are all freely available to allow comparison with existing approaches.

ABAE datasets One of our sources of data is our baseline [19], which made two

Table 1. Dataset Statistics

Dataset Name	file	# sentences
CitySearch	train	281,989
CitySearch	test	3,328
BeerAdvocate	train	16,882,700
BeerAdvocate	test	9,236
Sem2015-Restaurant	train	281,989
Sem2015-Restaurant	test	453
Sem2015-Laptop	train	1,083,831
Sem2015-Laptop	test	241

datasets available – one in the domain of *Restaurant*, known as *CitySearch* and another in the domain of *Beer* (originally presented in [23]), known as *BeerAdvocate*. Each dataset consists of two files: one for training the vector representation of words and one with test data. We used these datasets to test and compare our method to the baseline under the same conditions. For each sentence in the test file, the datasets have annotations that indicate the expected category.

SemEval datasets Our second source of data is the SemEval evaluation campaigns, specifically from an ABSA task². The reviews are on the domains of *Restaurant* and *Laptop*. We used the train and test files which contain the text of the reviews, the Aspect Category, as well as the aspect words, aspect word position in text, and their polarity. In our experiments, we considered the category entities as the categories for each review text. For example, if the review text is “*The pizza was great*”, the category for the aspect cluster is the word “*food*”. The SemEval datasets were used with the goal of testing adaptability of our solution across domains. In order to run the SUAEx with the SemEval datasets, some modifications on the *test data* had to be made: (i) we removed the instances which could be classified more than one category; (ii) we considered only the entities as category labels; and (iii) we discarded the categories with very few instances. The same modified version was submitted to both competing systems. For the datasets on the *Restaurant* domain, we selected three groups of reference words, {“*food*”}, {“*staff*”}, and {“*ambience*”}. For BeerAdvocate, we selected the groups {“*feel*”}, {“*taste*”}, {“*look*”}, and {“*overall*”}. And for the dataset Sem2015-Laptop, we selected the groups {“*price*”}, {“*hardware*”}, {“*software*”}, and {“*support*”}.

Tools and Parametrization. NLTK³ was used in the preprocessing module to remove stopwords, perform tokenization, sentence splitting, and for lemmatization. The domain-specific word-embeddings were created with Word2Vec⁴ using the following configuration: CBOV module, window size of 5 words, and 200 dimensions for the resulting vectors. The remaining parameters were used with the default values (negative sampling = 5, number of iterations = 15). The similarity between word vectors was measured with the cosine similarity in Gen-

² <http://alt.qcri.org/semeval2015/task12/>

³ <https://www.nltk.org/>

⁴ <https://code.google.com/archive/p/word2vec/>

sim [24] which reads the model created by the Word2vec. Scikit-learn⁵ provided the metrics for evaluating the quality of the aspects and categories extracted with the traditional metrics (precision, recall, and F1). *Amazon raw data* was taken from a public repository⁶ to be used as an external source of data. This data was necessary because for the *laptop* domain, since the training file provided by SemEval is too small and insufficient to create the domain word-embeddings.

Baseline. Our baseline, ABAE [19], was summarized in Section 3. Its code is available on a github repository⁷. We downloaded it and ran the experiments using the default configurations according to the authors’ instructions (*i.e.*, word-embeddings dimension = 200, batch-size = 50, vocabulary size = 9000, aspect groups = 14, training epochs = 15). Despite the authors having released a pre-trained model for the restaurant domain, we ran the provided code from scratch and step-by-step in order to measure the execution time.

6 Results and Discussion

Results for the quality of the aspects and categories extracted. The evaluation metrics precision, recall, and F1 were calculated by comparing the outputs generated by both methods against the gold-standard annotations. Fig. 2 shows the results of the evaluation metrics for both approaches averaged across categories. As an overall tendency, SUAEX achieves better recall than the baseline in all datasets. ABAE tends to have a better precision in most cases (three out of four datasets). This is expected and can be attributed to the contrast in the way the two solutions use the attention mechanism. While ABAE only considers the highest attention-valued word in the sentence, SUAEX uses all the attention values in the sentences. This difference can be seen in the example from Fig. 3. SUAEX considers the reference words as a type of context to

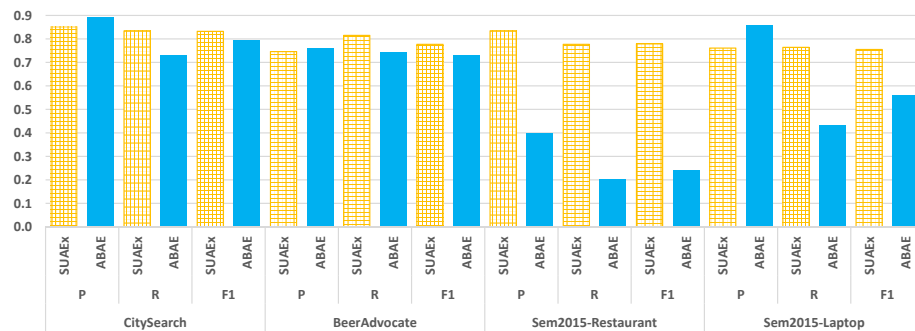


Fig. 2. Overall results averaged across categories

⁵ <https://scikit-learn.org/stable/>

⁶ <http://jmcauley.ucsd.edu/data/amazon/>

⁷ <https://github.com/ruidan/Unsupervised-Aspect-Extraction>

guide category attribution. Basically, for a given sentence, ABAE tries to be precise by focusing on a single word, while SUAEX tries to be more comprehensive by considering more words. Our recall improvement was superior to our decrease in precision, so our F1 results were better in all datasets. The results demonstrate the adaptability of SUAEX to different domains. On the SemEval datasets, SUAEX outperformed the baseline in nearly all cases. We can attribute ABAE’s poor results in the SemEval datasets to the dependence on the training (*i.e.*, raw) data. While SUAEX only uses the training data to generate the word-embeddings representation, ABAE also uses it in the evaluation module because it clusters the training data.

The results of the evaluation metrics per aspect category are shown in Table 2. SUAEX scored lower in recall for Ambience, Smell, and Look because the reference words (*i.e.*, the names of the categories themselves) are not as expressive as the reference words for the other categories. We can find more similar words for general terms like *food* or *staff* than for specific terms like *smell*.

Other works have also used the CitySearch and BeerAdvocate datasets. Thus we are also able to compare SUAEX to them. These works have applied techniques such as LDA [25], biterm topic-models [26], a statistical model over seed words [27], or restricted Boltzmann machines [28]. The same tendency found in the comparison with the baseline remains, *i.e.*, SUAEX achieves better recall in all cases except for the categories *Ambience* and *Smell*. In terms of F1, SUAEX is the winner for *Food*, *Staff*, *Taste*, *Smell*, and *Taste+Smell*.

Fig. 3 presents an example in which SUAEX assigns more accurate attention-values to an input sentence. This happens because the word *pizza* is more similar to our desired category *food* than the word *recommend*. However, our method is dependent on the reference words, and in some cases it can assign high attention values to adjectives (which typically are not aspects). This happens with the word *higher* in our example, which is an adjective and received the second highest score. This could be mitigated with a post-filter based on part-of-speech tagging.

In Table 3, we show the aspect words extracted for the CitySearch dataset. The extraction was performed by selecting the highest attention-valued words of each sentence and by considering the category classification results. This extraction can be used as an additional module in our framework (Fig. 1).

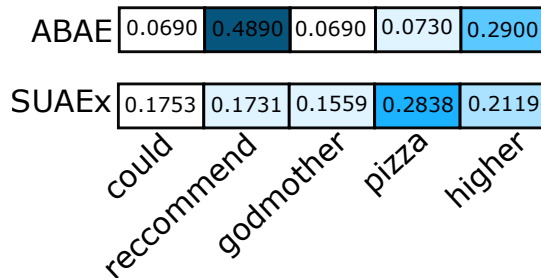


Fig. 3. An example of the attention values assigned to the input sentence “*I could n’t recommend their Godmother pizza any higher*” by the two solutions .

Table 2. Results for the quality of aspect category extraction.

Category	SUAEx			ABAE		
	P	R	F1	P	R	F1
CitySearch						
Food	0.917	0.900	0.908	0.953	0.741	0.828
Staff	0.660	0.872	0.752	0.802	0.728	0.757
Ambience	0.884	0.546	0.675	0.815	0.698	0.740
BeerAdvocate						
Feel	0.687	0.832	0.753	0.815	0.824	0.816
Taste	0.656	0.794	0.718	0.637	0.358	0.456
Smell	0.689	0.614	0.649	0.483	0.744	0.575
Taste+Smell	0.844	0.922	0.881	0.897	0.853	0.866
Look	0.876	0.849	0.862	0.969	0.882	0.905
Sem2015-Restaurant						
Food	0.953	0.674	0.789	0.573	0.213	0.311
Staff	0.882	0.714	0.789	0.421	0.159	0.213
Ambience	0.627	0.967	0.760	0.107	0.206	0.141
Sem2015-Laptop						
Price	0.750	0.915	0.824	0.895	0.576	0.701
Hardware	0.785	0.797	0.791	0.914	0.481	0.631
Software	0.714	0.455	0.556	0.714	0.114	0.196
Support	0.667	0.800	0.727	0.083	0.200	0.118

Table 3. Aspects extracted by SUAEX for the CitySearch dataset

Category	Aspect Terms
Staff	<i>replied, atmosphere, answering, whoever, welcomed, child, murray, manager, existence, cold, staff, busy, forward, employee, smile, friendly, gave, woman, dessert, early, kid, lady, minute, bar, helpful, wooden, always, greeting, server, notified, busier, nose, night, guy, tray, seating, everyone, hour, crowd, people, seat, lassi, proper, divine, event, folk, even, waitstaff, borderline, ice</i>
Ambience	<i>antique, atmosphere, outdoor, feel, proximity, scene, sleek, bright, weather, terrace, dining, surroundings, music, calm, peaceful, ambience, cool, sauce, location, ceiling, garden, painted, relaxed, dark, warm, artsy, excellent, tank, furniture, dim, bar, romantic, level, inside, parisian, air, architecture, aesthetic, adorn, beautiful, brightly, neighborhood, ambience, alley, elegant, decour, leafy, casual, decor, room</i>
Food	<i>seasonal, selection, soggy, chinese, cheese, penang, dosa, doughy, corned, sichuan, mojito, executed, innovative, dish, chicken, calamari, thai, butter-nut, bagel, northern, vietnamese, paris, menu, technique, dumpling, dhal, better, location, congee, moules, rice, sauce, ingredient, good, straightforward, mein, food, dessert, overdone, appetizer, creatively, fusion, know, unique, burnt, minute, panang, risotto, shabu, roti</i>

Runtime Results. In order to obtain the runtime results, we ran both methods on the same configuration (Intel Core i7-4790 CPU, GeForce GTX 745 GPU, and 16GB of main memory). Since both methods went through the same preprocessing steps, our comparison focused on the attention mechanisms (*Attention Neural Network training* in ABAE and the *Similarity as Attention Mechanism* module in SUAEX). Table 4 shows the runtime results. The differences are remarkable – ranging from one thousand to almost ten thousand times. For the BeerAdvocate dataset, we were unable to obtain the runtimes for the baseline because the number of training sentences was too large. One could argue that, in practice, these differences are not so significant because they concern the training phase, which can be performed once only. However, training has to be repeated for each domain and, from time to time, to cope with how the vocabulary changes.

Table 4. Runtime for both methods in all datasets

	CitySearch	BeerAdvocate	Sem2015-Restaurant	Sem2015-Laptop
SUAEX	42 sec	7 min	13 sec	36 sec
ABAE	12 hours	undefined	12 hours	98 hours

7 Conclusion

This paper proposes SUAEX which is an alternative approach to deep neural network solutions for unsupervised aspect extraction. SUAEX relies on the similarity of word-embeddings and on reference words to emulate the attention mechanism used by attention neural networks.

With our experimental results, we concluded that SUAEX achieves results that outperform the state-of-the-art in ABSA in a number of cases at remarkably lower runtimes. In addition, SUAEX is able to adapt to different domains.

Currently, SUAEX is limited to dealing with aspects represented as single words. As future work, we will extend it to treat compound aspects such as “*wine list*”, “*battery life*”. Also, we will improve the selection of reference words by using hierarchical data such as subject taxonomies.

Acknowledgments: This work was partially supported by CNPq/Brazil and by CAPES Finance Code 001.

References

1. Feldman, R.: Techniques and applications for sentiment analysis. *Commun. ACM* **56** (2013) 82–89
2. Zhang, L., Liu, B.: Aspect and entity extraction for opinion mining. In: *Data mining and knowledge discovery for big data*. Springer (2014) 1–40
3. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. *Computational Linguistics* **37** (2011) 9–27
4. Yin, Y., Wei, F., Dong, L., Xu, K., Zhang, M., Zhou, M.: Unsupervised word and dependency path embeddings for aspect term extraction. In: *IJCAI*. (2016) 2979–2985

5. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Recursive neural conditional random fields for aspect-based sentiment analysis. In: EMNLP. (2016) 616–626
6. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: AAAI. (2017) 3316–3322
7. Giannakopoulos, A., Musat, C., Hossmann, A., Baeriswyl, M.: Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. In: WASSA. (2017) 180–188
8. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR **abs/1409.0473** (2014)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26. (2013) 3111–3119
10. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. (2014) 1532–1543
11. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. CoRR **abs/1607.04606** (2016)
12. Liu, B.: Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* **5** (2012) 1–167
13. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: SIGKDD. (2004) 168–177
14. Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: SocialNLP. (2014) 28–37
15. Jin, W., Ho, H.H., Srihari, R.K.: A novel lexicalized HMM-based learning framework for web opinion mining. In: ICML. (2009) 465–472
16. Jakob, N., Gurevych, I.: Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In: EMNLP. (2010) 1035–1045
17. Moghaddam, S., Ester, M.: ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In: SIGIR. (2011) 665–674
18. Toh, Z., Su, J.: NLANGP at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In: SemEval-2016. (2016) 282–288
19. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An unsupervised neural attention model for aspect extraction. In: ACL. Volume 1. (2017) 388–397
20. Poria, S., Cambria, E., Gelbukh, A.: Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* **108** (2016) 42 – 49
21. Salazar, C.: Una taxonoma para los objetos. *Revista Universidad de Antioquia* (2014) 78–82
22. Vargas, D.S., Moreira, V.P.: Identifying sentiment-based contradictions. *JIDM* **8** (2017) 242–254
23. McAuley, J., Leskovec, J., Jurafsky, D.: Learning attitudes and attributes from multi-aspect reviews. In: ICDM. (2012) 1020–1025
24. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: LREC Workshop on New Challenges for NLP Frameworks. (2010) 45–50
25. Zhao, W.X., Jiang, J., Yan, H., Li, X.: Jointly modeling aspects and opinions with a maxent-lda hybrid. In: EMNLP. (2010) 56–65
26. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: WWW. (2013) 1445–1456
27. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: ACL. (2012) 339–348
28. Wang, L., Liu, K., Cao, Z., Zhao, J., de Melo, G.: Sentiment-Aspect Extraction based on Restricted Boltzmann Machines. In: ACL. (2015) 616–625