

# Unsupervised Aspect Term Extraction for Sentiment Analysis through Automatic Labeling

Danny Suarez Vargas<sup>id</sup><sup>a</sup>, Lucas R. C. Pessutto<sup>id</sup><sup>b</sup> and Viviane Pereira Moreira<sup>id</sup><sup>c</sup>

*Institute of Informatics, UFRGS, Brazil*

**Keywords:** Sentiment Analysis, Unsupervised Aspect Term Extraction, Topic Models, Word-embeddings.

**Abstract:** In sentiment analysis, there has been growing interest in performing finer granularity analysis focusing on entities and their aspects. This is the goal of Aspect-based Sentiment Analysis which commonly involves the following tasks: Opinion Target Extraction (OTE), Aspect term extraction (ATE), and polarity Classification (PC). This work focuses on the second task, which is the more challenging and least explored in the unsupervised context. The difficulty arises mainly due to the nature of the data (user-generated contents or product reviews) and the inconsistent annotation of the evaluation datasets. Existing approaches for ATE and OTE either depend on annotated data or are limited by the availability of domain- or language-specific resources. To overcome these limitations, we propose UNsupervised Aspect Term Extractor (UNATE), an end-to-end unsupervised ATE solution. Our solution relies on a combination of topic models, word embeddings, and a BERT-based classifier to extract aspects even in the absence of annotated data. Experimental results on datasets from different domains have shown that UNATE achieves precision and F-measure scores comparable to the semi-supervised and unsupervised state-of-the-art ATE solutions.


## 1 INTRODUCTION


Currently, more than half of the world population has access to the Internet (International Telecommunication Union, 2018). In this scenario, the large amount of information freely available on the Web is a natural consequence. More specifically, in the e-commerce scenario, interactions between consumers that aim at sharing useful information are frequent. There are different ways of conveying information and customer reviews are one of them. Customer reviews have been gaining more attention throughout the years because of their usefulness when people wish to purchase a product or service. Thus, it is not a surprise that *online reviews* are listed among the top-four factors that help people make purchase decisions (Jorij, 2017).


In this context, *sentiment analysis* (SA) (or *opinion mining*) aims to help deal with the large amount of data that one needs to process to make informed decisions (e.g., the selection of products/services in the e-commerce scenario). Since the earlier works published on SA, and over the years, researchers in-

roduced contributions that resulted in a continuously growing set of terminologies, definitions, sub-tasks, and approaches. SA can be addressed at three levels of granularity: document-level, sentence-level, or *aspect-level*, where the latter is the most fine-grained and challenging of the three.

Regarding the granularity of the analysis, while earlier works focused on document and sentence levels, more recently, the aspect-level gained attention. Such solutions, known as *Aspect-Based Sentiment Analysis (ABSA)* (Zhang and Liu, 2014) deal basically with the tasks of extracting and scoring the opinion expressed towards entities and their aspects. The entities are the main topic mentioned in the text, and the aspects are their attributes or features. 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 “*restaurant*”. The words “*amazing*” and “*incredible*” are the opinion words with respect to the two aspects, and the polarity label for the two aspects and for the entity is *positive*. Typically, ABSA is divided into two steps: Information Extraction and Polarity Classification (PC). The first step comprises two subproblems, Opinion Target Extraction (OTE) and Aspect Term Extraction (ATE). Our focus is on the *second subproblem*, i.e., ATE. Re-

<sup>a</sup>  <https://orcid.org/0000-0002-7177-2134>

<sup>b</sup>  <https://orcid.org/0000-0003-4826-960X>

<sup>c</sup>  <https://orcid.org/0000-0003-4400-054X>

view texts can be ambiguous and contain acronyms, slang, and misspellings. Thus, extracting fine-grained information from reviews is challenging. The level of difficulty of this task can be illustrated by the fact that the best performing semi-supervised methods achieved an F1 of 0.76 (Wu et al., 2018) at the SemEval 2014 dataset on the Restaurant domain. Unsupervised methods achieve even lower scores, with the best result for the same Restaurant dataset being F1 = 0.64 (Venugopalan and Gupta, 2020).

Another issue that contributes to the difficulty is that aspects could be multiword expressions. The task asks for the exact matching of start and end boundaries of an extracted aspect. For example, consider the input sentence (1) “*The fried rice is amazing here*”. The ATE solution could extract only the word “*rice*” and ignore the adjectival modifier “*fried*”. However, if the annotated aspect for the sentence considers the two-word expression “*fried rice*” as the correct aspect, the word “*rice*” will be considered an erroneous aspect even when it represents a desired aspect.

Existing solutions suffer from two main drawbacks: supervised and semi-supervised approaches depend on annotated data and unsupervised approaches are limited by the availability of domain- or language-specific resources. Relying on annotated data is problematic because it requires significant manual effort to build. Domain- or language-specific resources (large set of rules, lexicons, and error-prone tools such as dependency tree algorithms) are not always available, or even they report poor performance in languages other than English. An unsupervised approach that relies on a few sets of rules, exploits the availability of large non-annotated data on the Web, and uses the best established Natural Language Processing (NLP) tools is a desirable solution. The unsupervised approach and the low dependency on domain and language-specific resources enable adapting the ATE solution across domains. Our main contribution is the proposal of UNsupervised Aspect Term Extractor (UNATE), a simple end-to-end unsupervised solution for ATE. For a given domain, UNATE returns their aspects. UNATE does not require annotated data as it can automatically label aspects. We ran an experimental evaluation on datasets from different domains with the goal of answering the following research questions:

- **RQ1.** Can our unsupervised approach achieve results that are comparable to the state-of-the-art unsupervised ATE methods?
- **RQ2.** Is it possible to replace the manual annotation of entities and their aspects by an automatic method and achieve comparable performance in ATE?

Our evaluation showed that UNATE (i) achieves the best F1 among all unsupervised baselines for the restaurant domain and the second-best result for the laptop domain; (ii) results for precision and recall in both domains are close to the best scoring unsupervised baseline in both domains despite using fewer resources; and (iii) outperforms the supervised baselines from SemEval.

## 2 BACKGROUND

This section introduces the terminology, tasks, and methods that are used in this work.

### 2.1 Terminology and Tasks

A *review* represents the main input to SA solutions. This input conveys one or more opinions about a given product or service (Liu, 2012). For a given review, it is possible to identify the concepts of *Domain*, *Entity*, and *Aspect* which are used to refer to different levels of granularity (Pontiki et al., 2014; Liu, 2015). The *domain* is used to label an entire dataset with the broader subject that it refers to (e.g., restaurant, electronic products, etc.). A domain is identified by a *domain word*, i.e., its name. An *entity* is used to label an individual review text or sentence. The entity can be considered as the broader topic of a given sentence (e.g., food, service, and price are topics of the restaurant domain). The third concept *aspect* is used to describe features of a given entity (e.g., pizza and sushi are aspects of the food entity). Finally, these three concepts can be *Implicit* (i.e., not present in the text) or *Explicit* (i.e., present in the text).

**Aspect Term Extraction (ATE) and Opinion Target Extraction (OTE)** are defined as the extraction of entities and their aspects (attributes and features). Specifically, for a given text, OTE extracts entities and their aspects while ATE only extracts the aspects of a given entity (Pontiki et al., 2014, 2015). For example, in the sentence  $s = \text{“Great Laptop, I love the operating system and the preloaded software.”}$ , the desired OTE output is a set of entities and aspect terms  $at = \{\text{“Laptop”, “operating system”, “preloaded software”}\}$ , while the desired ATE output for the entity “*Laptop*” is the set of aspect terms  $\{\text{“operating system”, “preloaded software”}\}$ .

### 2.2 Supporting Techniques

**Topic Models.** Topic models or probabilistic topics models are a suite of machine learning/ NLP algorithms. Their goal is to find latent semantic structures

from raw text data by using the relation between three components: documents, words, and topics. The first and second components represent the observed data, while the third component is the desired information to be extracted. Topic models work over a generative assumption: the model assumes that the topics are generated first, before the documents (Blei, 2012).

**Word Embeddings.** For a given corpus of raw data, word embeddings learn the distributional representation of words by exploiting word co-occurrences. As a result, the embeddings model the semantic and syntactic similarities between words. Thus, it is possible to measure the probability of a word in a sentence even when this sentence was not seen before during training. Several word embeddings algorithms were proposed, including Word2vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), and FastText (Bojanowski et al., 2016).

**Clustering and Silhouette Score.** For a given set of data items, clustering is an unsupervised process that aims at creating  $k$  groups of items concerning a particular property. We can compare clusters obtained for different  $k$  values by evaluating two measures: *cohesion* and *separation*. Cohesion measures how similar items of a given group are, and separation measures how different the items of different groups are. The desired characteristics clustering are high cohesion and separation. The silhouette score aggregates cohesion and separation and can be used as a single measure to assess the clustering quality.

**Anomaly detection.** This task aims to find instances of a dataset that do not follow an expected behavior. These discordant instances are often referred to as anomalies or outliers. The anomaly detection task is particularly relevant when we want to perform data exploration to exploit the knowledge embedded in these instances (Chandola et al., 2009).

**Bidirectional Encoder Representations from Transformers.** BERT (Devlin et al., 2018) is the best-known transformer-based model which helps the research community to outperform state-of-the-art in several NLP tasks. The transformer model can be described as a stack of layers where each layer processes information through multiple self-attention mechanisms to obtain an encoded representation of texts. In other words, BERT encodes information through its stack of layers capturing linguistic and world knowledge information for a large amount of text data (Rogers et al., 2020). The workflow of BERT consists of two steps (Pre-training and Fine-tuning).

### 3 RELATED WORK

This section surveys related work in ATE. We start by introducing the seminal approaches and then move on to recent methods.

There are five basic approaches for ATE, namely: frequency-based, rule-based, sequential models, topic models, and deep learning.

The main advantage of the frequency-based approach (Hu and Liu, 2004) is that it is a simple and unsupervised solution which only relies on the co-occurrence of words such as nouns or adjectives. Its disadvantage is that these words do not always represent aspects (leading to false positives). The topic model approach (Brody and Elhadad, 2010) also has the advantage of being unsupervised but, unlike the frequency-based approach, it does not take into account word order or co-occurrence information. Instead, it considers that each word in a corpus is generated independently – this is the main disadvantage of this category of approaches which is responsible for the poor quality of the extracted aspects. The rule-based approach (Qiu et al., 2011) has the advantage of being more discriminative than the topic model approach. The goal is to reduce the number of false positives, but it has the disadvantage of being semi-supervised and requiring a computationally expensive linguistic resource to generate the dependency tree of texts. The sequential model approach (Jakob and Gurevych, 2010) has the advantage of exploring the fact that the input data (text) is sequential. For years, solutions based on sequential model algorithms like Conditional Random Fields (CRF), represented the state-of-the-art. However, they are a supervised and laborious option that relies on feature-engineering. Finally, deep learning solutions (Liu et al., 2015; Luo et al., 2019; Xu et al., 2018) represent the current state-of-the-art. Their main advantage is that they can automatically infer features from large amounts of annotated data. Their main disadvantage is that they require significant computational power.

The works that are more closely related to ours are either unsupervised or semi-supervised (*i.e.*, methods that use a few annotated instances). There are four methods that fit into these categories and that we use as baselines. Next, we briefly describe them.

Garcia-Pablos et al. (2014) proposed an adaptation of the double propagation algorithm Qiu et al. (2011). The proposed approach used a set of propagation rules, a part-of-speech (POS) tagger, and a dependency parser over domain-related non-annotated data to expand an initial set of seed words (aspects and opinions).

Giannakopoulos et al. (2017) organized their work

Table 1: Comparative Table of Resource Usage of Baseline works and UNATE.

Methods	Lexicon	Seed Words	External data	Dep. Parser
Garcia-Pablos et al. (2014)	No	Yes	Yes	Yes
Giannakopoulos et al. (2017)	Yes	No	Yes	No
Wu et al. (2018)	Yes	No	Yes	Yes
Venugopalan and Gupta (2020)	Yes	Yes	Yes	No
UNATE	No	No	Yes	No

into three steps. In the first step, it performs phrase extraction by selecting the high-quality ones using their frequency on a given corpus. In the second step, aspect candidates are obtained by pruning the high-quality phrases. The pruning process uses a combination of sentiment lexicons and syntactic rules. Finally, in the third step, the aspect candidates are used to perform aspect-level annotation of sentences to train a bidirectional Long Short-Term Memory classifier, in conjunction with CRF (Bi-LSTM-CRF).

Wu et al. (2018) proposed an automatic data annotation method to train a recurrent neural network classifier. This work follows three steps. In the first step, It extracts noun phrases as aspect candidates. In the second step, domain-correlation and opinion lexicon filter out undesired candidates. Finally, in the third step, the aspect candidates are used to perform aspect-level annotation of sentences to train a recurrent neural network classifier.

Venugopalan and Gupta (2020) proposed a three-module solution. In the first module, they use Named-Entity Recognition (NER), Part-of-speech Tagging (POS), and Dependency Parsing to perform data pre-processing. In the second module, they perform the aspect candidate extraction using a predefined set of rules and a domain-specific sentiment lexicon. Finally, in the third module, they prune the aspect candidates by calculating their similarities w.r.t a predefined seed aspect terms.

The baselines share similarities. Two steps are common in all solutions: aspect term candidate generation and the subsequent pruning of the undesired aspects. External data (*i.e.*, unannotated data that are not part of the dataset) is always considered to perform at least one of the two steps, and the POS tagger is the principal resource used to find multiword aspect terms. Table 1 summarizes compares the baselines and UNATE in terms of resource usage. We can see that all baselines use at least two out of the four resources while UNATE uses only one.

## 4 UNSUPERVISED ASPECT TERM EXTRACTION

This section describes UNATE, a simple unsupervised similarity-based solution for ATE. The intuition behind UNATE is to use different strategies to generate aspect term candidates (*i.e.*, to enhance recall) and then prune to select the best candidates (improving precision). UNATE is composed of five tasks depicted in Figure 1. The first three steps in UNATE aim at ensuring high recall while the last two steps focus on precision. The next subsections describe each step.

### 4.1 Topic Extraction and Selection

The goal of this task is to automatically extract a set of representative topics (*i.e.*, the vocabulary that characterizes the domain of interest) from the raw data in a given domain. In order to achieve that, we rely on topic models and word-embeddings. The intuition is to combine the unsupervised capabilities of the former and the ability to capture semantics of the latter. Topic Extraction and Selection (Step 1) receives three inputs – the raw domain data (Input  $\mathcal{D}$ ), domain embeddings (Input  $E^{\mathcal{D}}$ ), and a parameter value  $\theta$ , which tells the number of topics to be considered. The process is as follows. First, the raw data is pre-processed through sentence splitting and tokenization. Then, the pre-processed raw data is used to extract and select a set of representative topic words (through topic extraction, clustering, and selection). Topic extraction extracts the  $\theta$  words with the highest probability of being topics. This is achieved by applying a topic modeling algorithm and directly extracting the top- $\theta$  words. Topic clustering, for a given  $\theta$ -value, is responsible for finding the best clustering configuration (the best number of clusters to group the  $\theta$  topics). First, the vector representation of each topic  $t_i$  ( $i = 1, 2, \dots, \theta$ ) is obtained. Then,  $\theta - 2$  executions of a clustering algorithm are performed and their silhouette scores are measured. Finally, the cluster configuration with the closest silhouette score in relation to

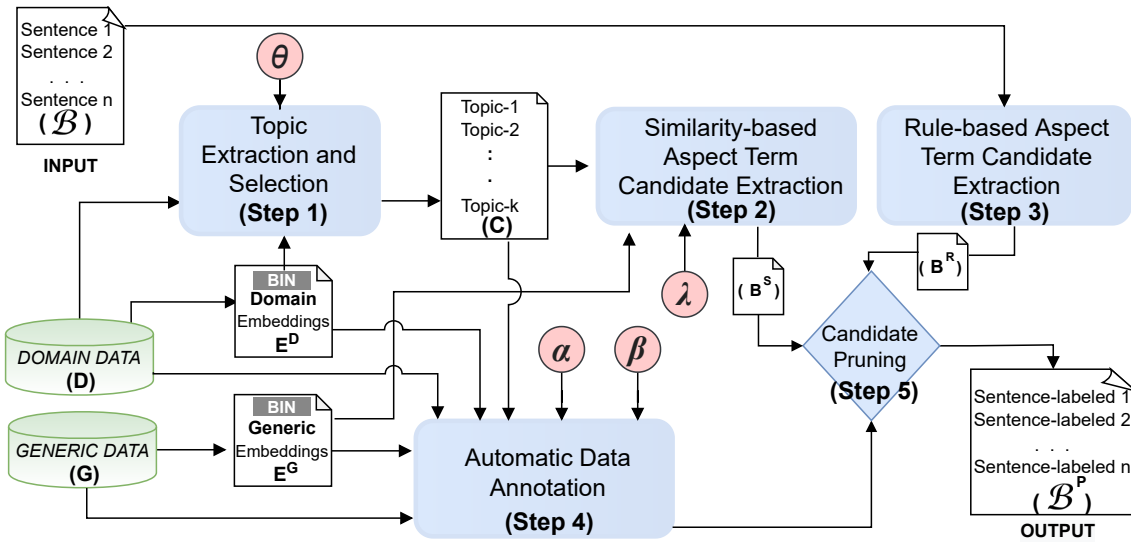


Figure 1: UNATE framework.

the mean is selected. Selection is responsible for identifying the cluster that contains the words that better represent the given domain. To do that, we propose a similarity-based algorithm, described as follows. The algorithm assigns a real value  $r$  to each cluster. For each cluster, its  $r$ -value is calculated by averaging the similarity values of all two-word combinations of the topics. Finally, the cluster with the highest  $r$  value is selected. This way, we consider each word in the selected cluster as a representative topic for a given domain. At the end of this process, we obtain *Output C*, which consists of  $k$  topic words.

## 4.2 Similarity-based Aspect Term Candidate Extraction

The goal of this step is to generate aspect term candidates relying on the similarity with the desired domain. We start by identifying single words and then search for multiwords. The inputs here are: the topics that were generated in Step 1 ( $C$ ), generic embeddings ( $E^G$ ), and a parameter  $\lambda$ , which controls the number of candidates to be generated.

The tasks in this step are detailed in Figure 2 through an example. Let  $E^G$  be a word embeddings model on the generic domain  $G$ ,  $B_{(n \times m)}$  is a matrix of  $n$  sentences and  $m$  words, and  $C = w_1^c, w_2^c, \dots, w_k^c$  is a list of  $k$  topic words. We start by obtaining two matrices,  $B_{(n \times m)}^*$  and  $\tilde{B}_{(n \times m)}$ , which represent the pre-processed and the POS-tagged versions of  $B$ , respectively. Next, we define the extraction process as the composition of three functions  $h(j(f))$ : measuring Similarity ( $f$ ), selecting single word candidates ( $j$ ),

and multiword searching ( $h$ ).

**Measuring similarity ( $f$ ).** Function ( $f$ ), for measuring similarity, takes three inputs: (i) the pre-processed version of the test sentences  $B_{(n \times m)}^*$ , (ii) the list of topic words  $C$ , and (iii) the generic embeddings  $E^G$ . The output of  $f$  is the similarity matrix  $S_{(k \times n \times m)}^G$ , where each real value  $s_{zyx}$  represents the similarity between each word  $b_{yx} \in B^*$  w.r.t. each topic word  $w_z^c \in C$  by using their vector representations in  $E^G$ .

For a given sentence  $b \in B^*$  which consists of  $m$  words, our goal here is analogous to the attention mechanism in neural networks. The difference is that while the attention mechanism assigns real values (attention) to each word in  $b$  concerning its neighborhood (context), our similarity function assigns real values (similarity) to each word in  $b$  in relation to a fixed set of topic words  $C$ . For example, for the topic word  $w_z^c = \text{“food”} \in C$ , a word  $b_{yx} = \text{“pizza”} \in B^*$ , and a word embeddings model in the Generic domain  $E^G$ ,  $f$  calculates the real value  $s_{zyx} \in S_{(k \times n \times m)}^G$  which represents the importance of  $b_{yx}$  (i.e., the word *pizza*) in relation to  $w_z^c$  (i.e., the topic word *food*).

The cosine is used to quantify the similarity of a given word  $w_i$  in relation another word  $w_j$ , or a group of words  $\mathcal{W}$ . In this function, we use two types of similarity values. The first one is the *direct similarity (DSim)*, which is obtained by the direct comparison of two words. The second one is the *contextual similarity (CSim)*, which is obtained by the comparison of two words in relation to *contextual words*, i.e., neighboring words that are used as cues to find the similarity value. Finally, the similarity values are obtained by averaging the direct and contextual similarities of

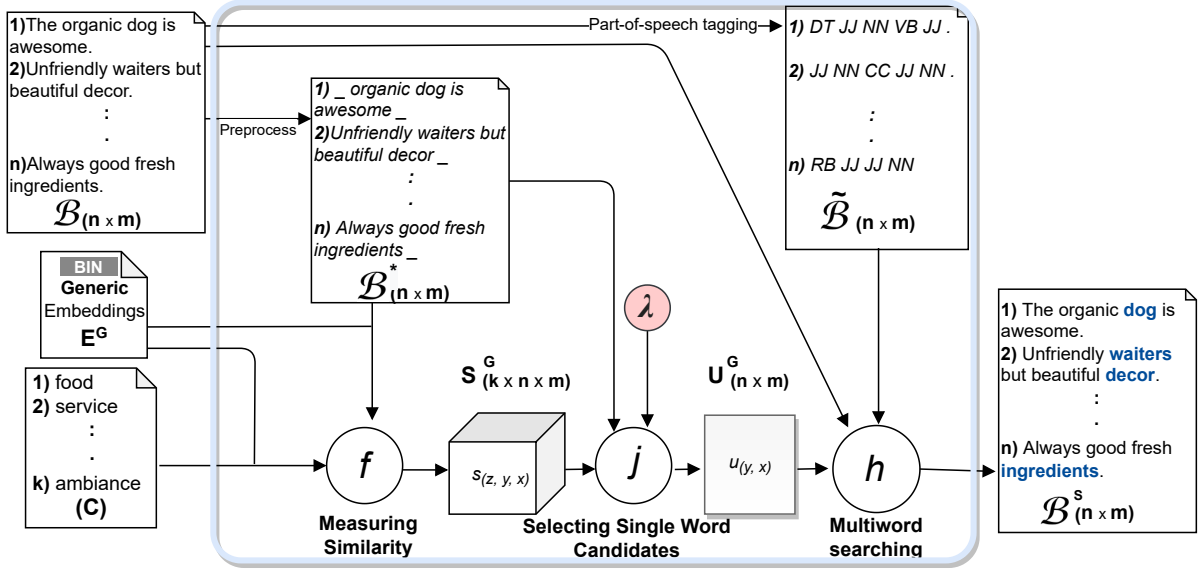


Figure 2: Example the process for similarity-based aspect term candidate extraction (step 2 in Fig. 1).

each word  $\in \mathcal{B}^*$  w.r.t. a list of topic words  $\mathcal{C}$ . For example, if we consider the *laptop* domain, the topic word “*display*”, and the sentence “*The notebook is beautiful, I love the clear resolution*” which is a row of  $m$  words  $\in \mathcal{B}^*$ , the similarity function should assign a higher score to the word “*resolution*” and lower scores to the other words. However, if an adjective such as “*clear*” always appears near our target (*i.e.*, *resolution*), direct similarity may inappropriately assign a high score to it. To avoid assigning high scores to words that are unlikely to be aspects in the domain of interest, we rely on contextual similarity. Contextual similarity solves this problem by using a set of words that represents the words that we do not want as output. These words guide the similarity algorithm into the correct direction. For contextual similarity, we use a set of positive and negative words, *i.e.*, positive words are related to the topic of interest, while the negative words belong to the remaining topics. For example, for a given topic word  $w_z^c$  from a list of topic words  $\mathcal{C}$ , and a word  $b_{yx}$  from input  $\mathcal{B}^*$ , we obtain the contextual similarity of  $b_{yx}$  w.r.t.  $w_z^c$  by considering  $w_z^c$  as the unitary set of positive words  $\mathcal{W}^p$ , and  $\mathcal{W}^n = \mathcal{C} - w_z^c$  as the set of negative words.

$$DSim.b_{yx} = \max_{w^p \in \mathcal{W}^p} \cos(b_{yx}, w^p) \quad (1)$$

$$CSim.b_{yx} = \frac{1}{|\mathcal{W}^p| |\mathcal{W}^n|} \sum_{(w^p \in \mathcal{W}^p)} \sum_{(w^n \in \mathcal{W}^n)} \cos(b_{yx}, w_a^p, w_b^n) \quad (2)$$

where  $\cos$  is the cosine similarity between two vectors. For a given word  $b_{yx}$ , a set of positive words  $\mathcal{W}^p$ , and a set of negative words  $\mathcal{W}^n$ , we use Equation 1 to measure the direct similarity and Equation 2 to measure the contextual similarity.

**Selecting Single Word Candidates ( $j$ ).** The second function ( $j$ ), for identifying single word aspect candidates, takes three inputs, the pre-processed  $n$  sentences of  $m$  words  $\mathcal{B}^*(n \times m)$ , its similarity values w.r.t.  $k$  topics  $S^G(k \times n \times m)$ , and anomaly detection parameter  $\lambda$ , and returns a matrix  $U^G(n \times m)$  which allows us to know whether a word in  $\mathcal{B}$  represents a single word aspect candidate. Function  $j$  can be described as a sequence of two sub-functions,  $j'$  and  $j''$ , where  $j'$  works at word level and  $j''$  works at sentence level. First, sub-function  $j'$  takes the three dimensional matrix  $S(k \times n \times m)$  and transforms it into a two dimensional matrix  $\bar{S}(n \times m)$ , where each  $\bar{s}_{yx} \in \bar{S}$  is obtained by aggregating the  $k$  similarity values of  $b_{yx}$  in  $S$ . Next, the second sub-function  $j''$ , builds our desired single word aspect candidate matrix  $U^G(n \times m)$  by initializing all cells with *False*. Then, for each row  $\bar{s}_y \in \bar{S}$  which consists of  $m$  cells,  $j''$  finds the best similarity rated ones and labels them with *True*. We define the first sub-function  $j'$  as the simple average aggregation of  $k$  values, and the second sub-function  $j''$  as an *anomaly detection task*. We hypothesize that the average aggregation captures the relevance of a given word in relation to  $k$  topics, and the anomaly detection algorithm extracts the most salient words in a sentence. Furthermore, we use the parameter  $\lambda$  in function  $j''$  to define the desired proportion of outliers we want to obtain as single word aspect candidates.

**Multiword searching ( $h$ ).** Finally, the third function ( $h$ ) is designed to determine whether a single word aspect  $u_{yx} \in U$  is part of a multiword. This function

takes three inputs: (i) the input sentences  $\mathcal{B}_{(n \times m)}$ , (ii) its pre-processed version  $\tilde{\mathcal{B}}_{(n \times m)}$ , and (iii) the matrix of single words  $U_{(n \times m)}^G$ . Our goal here is to use the single word aspects found by function  $j$  jointly with the POS-tag information of  $\tilde{\mathcal{B}}$ . In greater detail, for each cell  $u_{yx} \in U$  of a given row  $u_y$  which has the a True value, function  $h$  verifies whether  $u_{yx}$  is part of a multiword. This verification process relies on the POS-tag information of the surroundings cells, *i.e.*, cells on the left and right are checked in order to label them as aspect or not. In this function, we use the higher probability of nouns being aspects to extend an aspect single word to a multiword only when its surrounding cells are nouns. In summary, for each single word aspect in  $\mathcal{B}$ , function  $h$  checks the surrounding words and uses the POS-tag information to decide whether to extend an aspect to a multiword. Finally,  $h$  returns the similarity-based aspect term candidates as our output  $\mathcal{B}^S$ .

### 4.3 Rule-based Aspect Term Candidate Extraction

The subject-verb-object form is the predominant sentential form in English and many other languages (Han, 2009). Thus, one could use a small set of rules to identify aspect term candidates. This sentential form is represented in our context by the sequential rule that contains the following POS tags – Noun, Verb, and Adjective/Adverb. We name this rule *the fundamental rule*. For example, the sentence “*The food was incredible*” follows the fundamental rule. The word “*food*” is the noun, “*was*” is the verb, and “*incredible*” is the adjective modifier of the noun. To increase the utility of the fundamental rule, we model it as a triplet (Target, Link, Opinion), where *target* is the entity or aspect being evaluated, *opinion* is the modifier of the target, and *link* represents the co-dependency of opinions and targets. This model helps us consider single-word and multiword occurrences as valid components. For example, the single word “*screen*” and the multiword “*resolution screen*” are valid target components. The link and opinion components follow the same logic to consider single-word and multiword occurrences. In summary, targets could be nouns or noun phrases, links could be verbs or compound verbs, and opinions could be adjectives/adverbs or compound adjectives/adverbs. This task takes the pre-processed and the POS-tagged versions of  $\mathcal{B}$ . For each sentence in  $\mathcal{B}$ , the output  $\mathcal{B}^R$  consists of the original sentence along with the corresponding triplets.

### 4.4 Automatic Data Annotation

In any domain, it is important to distinguish aspects that indeed belong to the domain (*i.e.*, target) from aspects that belong to a generic domain. For example, in the sentences “*The food was incredible*” and “*The weather was incredible*”, *food* is a target from the restaurant domain while *weather* is not. Our assumption is that *If we have enough domain-specific data, domain-generic data, a set of domain-specific topics, and our fundamental rule, we can build a classifier capable of distinguishing between domain-specific and domain-generic aspects*. In order to train this classifier, this step automatically generates the training data. Positive (in-domain) and negative (out-of-domain) instances are required. Positive instances come from the domain data ( $\mathcal{D}$ ) and the negative instances come from the generic data ( $\mathcal{G}$ ). The triplets are obtained for both  $\mathcal{D}$  and  $\mathcal{G}$  using the fundamental rule. Using the entire set of sentence-triplets is not ideal since rules can generate false positives and false negatives that would lower the classifier performance. Thus, we only select sentence-triplets for which there is a high confidence as to which class they belong to. This selection is based on two similarity thresholds  $\alpha$  and  $\beta$ , which controls the quality of the in-domain and out-of-domain instances, respectively. Instances with similarity  $\geq \alpha$  w.r.t. the  $k$  topics are kept as positive training instances (remove false positives). For the negative instances, the idea is to discard the ones that may be aspects in the domain (remove false negatives). Thus, if their similarity with the  $k$  topics  $\geq \beta$  are discarded from the negative instances. Intuitively,  $\alpha$  should have a lower value than  $\beta$ .

### 4.5 Candidate Pruning

Once the training instances were generated in step 4, here they are used to fine-tune a BERT-based classifier. This classifier is used to prune the aspect term candidates generated in steps 2 and 3.

### 4.6 Parameters

UNATE has four parameters  $\lambda$ ,  $\alpha$ , and  $\beta$  (which range from 0 to 1) and  $\theta \in \mathbb{N}$ . The behavior of Steps 1 and 2 is controlled by  $\theta$  and  $\lambda$ . We observe that larger values for these two parameters result in a greater number of aspect term candidates and vice-versa.

Steps 4 and 5 are affected by  $\alpha$  and  $\beta$ .  $\alpha$  works over the domain data and represents the minimum similarity value of aspect term candidates w.r.t the topics selected in Step 1. The  $\beta$  value works over the generic data and represents the maximum simi-

Table 2: Dataset Statistics.

Dataset Name	Domain	# sentences	Description
Sem2014-Rest	Restaurant	3041	Train Data
		800	Test Data
Sem2014-Lapt	Laptop	3045	Train Data
		800	Test Data

Table 3: External Raw Data Statistics.

Raw Data Name	Domain	# sentences
TripRawData	Restaurant	0,73 M
AmazonRawData	Software & Celphones	1,48 M
WikiRawData	Generic	1,12 M

larity value of aspect term candidates w.r.t the topics selected Step 1. The combination of a high  $\alpha$  and a low  $\beta$  yields a better filtering of aspect term candidates. Large  $\alpha$  values mean that a given aspect term candidate should be very similar to one or more topics (to be considered as desired aspect). A lower  $\beta$  means that a given aspect term candidate should not be similar to any topic extracted in Step 1 (to be considered as a undesired aspect).

In summary, large values for  $\theta$  and  $\lambda$  increase the number of aspect term candidates; large values for  $\alpha$  increase accuracy in the selection of domain-related aspect terms; and low values for  $\beta$  guarantee good performance in distinguishing between desired and non-desired aspect terms.

## 5 EXPERIMENTS

In this section, we describe the experiments performed to test UNATE in different contexts.

### 5.1 Materials and Methods

**Datasets.** The source of data is the SemEval evaluation campaign (2014)<sup>1</sup>. This event released two datasets for the ATE task in 2014. We use these datasets in their original form to further comparison with our ATE baseline works. Each dataset contains sentences from which aspects will be extracted and it has ground truth annotations (*i.e.*, the expected outputs) to allow the calculation of the evaluation metrics. Datasets are summarized in Table 2.

**External Data.** For the *restaurant* domain, we collected *raw text data* from TripAdvisor<sup>2</sup>. For the *laptop* domain, we collected *raw text data* from a publicly available Amazon dataset<sup>3</sup>. For the *generic* do-

main, we collected *raw text data* from a publicly available Wikipedia dump<sup>4</sup>. The statistics of the external data are summarized in Table 3.

**Evaluation Metrics.** The three typical metrics used in classification tasks are Precision, Recall, and F1. *Precision* quantifies, among the instances that were classified as belonging to a class  $c_i$  what proportion indeed belongs to  $c_i$ , according to the ground truth. *Recall* measures the proportion of instances that belong to class  $c_i$ , according to the ground truth, that were classified as such. The *F1*, is the weighted harmonic between precision and recall. When both precision and recall receive the same weight, the metric is called *F1* and it is the standard metric for ATE.

**Baselines.** We compared UNATE with the following baselines:

- the supervised baseline used in SemEval 2014 provided by the task organizers (Pontiki et al., 2014), which belongs to the frequency-based approach;
- a semi-supervised baseline (Wu et al., 2018), described in Section 3; and
- three unsupervised methods (García-Pablos et al., 2014; Giannakopoulos et al., 2017; Venugopalan and Gupta, 2020) also described in Section 3.

### 5.2 Experimental Setup

Experimental results for all methods were calculated for the test data (Table 2). As external data (Input  $\mathcal{D}$ ), UNATE used TripRawData for the restaurant domain and AmazonRawData for the laptop domain. WikiRawData was used as generic data (Input  $\mathcal{G}$ ) in both cases.

We followed the same pre-processing steps for our three inputs (*i.e.*, sentence splitting, word tokenization, and POS tagging) using the default configurations from NLTK<sup>5</sup> and Stanford CoreNLP<sup>6</sup>. Word2Vec<sup>7</sup> was used to create domain embeddings ( $E^{\mathcal{D}}$ ) and ( $E^{\mathcal{G}}$ ) with the skipgram model, window size = 5, and 200 dimensions. Pre-trained embeddings (word2vec-google-news-300<sup>8</sup> were used as generic embeddings ( $E^{\mathcal{G}}$ ) in all cases. For topic extraction, the Mallet<sup>9</sup> implementation of Latent Dirichlet Allocation (LDA) was applied on the domain data using  $\theta$  values ranging from 6 to 12. Topic clustering and selection were performed using  $E^{\mathcal{D}}$  and the default

<sup>4</sup><https://dumps.wikimedia.org/>

<sup>5</sup><https://www.nltk.org/>

<sup>6</sup><https://github.com/stanfordnlp/CoreNLP>

<sup>7</sup><https://radimrehurek.com/gensim/>

<sup>8</sup><https://github.com/eyaler/word2vec-slim/raw/master/GoogleNews-vectors-negative300-SLIM.bin.gz>

<sup>9</sup><http://mallet.cs.umass.edu/topics.php>

<sup>1</sup><http://alt.qcri.org/semeval2014/task4/>

<sup>2</sup><https://www.tripadvisor.com/>

<sup>3</sup><http://jmcauley.ucsd.edu/data/amazon/>



configurations of  $k$ -means and Silhouette Score<sup>10</sup>.

Similarity-based aspect term candidates (Step 2) were generated for ten  $\lambda$  values. Rule-based aspect term candidate extraction (Step 3) does not have parameters.

For automatic data annotation (Step 4), we experimented with  $\alpha$  values ranging from 0.15 to 0.25 and with  $\beta$  values ranging from 0.45 to 0.55. We used the resulting annotated data to build instances for fine-tuning a pre-trained BERT model<sup>11</sup>. Each instance consists of a value-label pair. The value is in the form: “[CLS] Target [SEP] Link [SEP] Opinion”, where the label is 0 or 1. We tested with the number of epochs ranging from 1 to 4. The best accuracy-loss relation was achieved with 2 epochs. We built 240k and 218k automatically annotated instances for the restaurant and laptop domains, respectively. The parameters were  $\theta = 12$ ,  $\alpha = 0.25$ ,  $\beta = 0.55$ , and  $\lambda = 1.0$ .

### 5.3 Results

The results for UNATE and the baselines are shown in Table 4. In comparison with the unsupervised baselines, UNATE has the best F1 results in the restaurant domain and second best in the laptop domain. UNATE outperforms the supervised baseline by a wide margin in terms of recall and F1. The best scoring method for precision (Giannakopoulos et al., 2017) relies on a lexicon and yet its recall results are lower than UNATE’s. The best scoring method for recall Venugopalan and Gupta (2020) also requires a lexicon and seed words and its precision is lower than UNATE’s. The semi-supervised baseline (Wu et al., 2018) outperformed UNATE. However, it requires a sample of annotated data, a lexicon, and a dependency parser. Based on these results, the answer to *RQ1 - Can our unsupervised approach achieve results that are comparable to the state-of-the-art unsupervised ATE methods?* is yes.

To answer our second research question *Is it possible to replace the manual annotation of entities and their aspects by an automatic method and achieve comparable performance in ATE?*, we used our approach in conjunction with a supervised method – the baseline used in SemEval 2014 (Pontiki et al., 2014). The baseline run (SemEval<sub>S</sub>) was the same provided by the SemEval organizers. It takes the aspects that are provided in the training set and uses them to extract aspects on the test set. The unsupervised run (SemEval<sub>U</sub>) uses the training instances as Input  $\mathcal{B}$  for UNATE, but ignores the aspect labels. The entire process depicted in Figure 1) is performed. We obtain

Output  $\mathcal{B}^P$ , *i.e.*, sentences labeled with aspect information. In other words, we are automatically annotating the training instances without supervision. Finally, Output  $\mathcal{B}^P$  is fed to the baseline which performs ATE. The results are in Table 5. For the restaurant domain, our automatic method for data annotation improves the performance of SemEval<sub>S</sub> in all metrics. Recall was substantially enhanced by 15 percentage points. This happens because UNATE provides the classifier with a greater variety of aspects in different contexts that come from the external data. On the other hand, in the laptop domain, SemEval<sub>U</sub> has lower scores in all metrics compared to SemEval<sub>S</sub>. This can be attributed to the fact that the external data (*i.e.*, Amazon reviews) has important differences in relation to the test data. Amazon reviews tend to be longer, more descriptive and less opinionated than reviews in the test dataset. In fact, the scores of all methods are lower in the laptop dataset. This is due to characteristics of the domain – aspects tend to be longer *e.g.*, “*performance and feature set of the hardware*” and less frequent. Although UNATE did not improve the results of the supervised method in the laptop domain, it can still be considered as an alternative when there is no training data available.

The limitations of UNATE are the need for external data and a POS tagger. External data can be automatically collected without human intervention and POS taggers are more widely available than dependency parsers that are required by some approaches.

## 6 CONCLUSION

This paper proposed UNATE, an unsupervised approach for aspect term extraction that aims at circumventing two limitations existing approaches, namely the dependency on annotated data or the need for language, and specific resources. UNATE relies on topic models, word embeddings, and external (unlabeled) data to automatically annotate review sentences. Then, a fine-tuned BERT classifier identifies aspect terms in a given domain.

We experimented with UNATE in two standard datasets for aspect term extraction in different domains – restaurants and laptops. Our results showed that UNATE outperforms the baselines in the restaurant domain and comes in second place in the laptop domain. We also tested whether UNATE can be used to automatically label instances to train a supervised method. We found that in the restaurant domain the results of the baseline improved with UNATE’s automatic labeling but the same did not happen in the laptop domain. Still, our approach can represent a vi-

<sup>10</sup><https://scikit-learn.org/>

<sup>11</sup><https://github.com/google-research/bert>

Table 4: ATE results. Best unsupervised results in bold.

Method	Type	Sem2014-Rest			Sem2014-Laptop		
		P	R	F1	P	R	F1
SemEval (Pontiki et al., 2014)	Supervised	0.52	0.43	0.47	0.44	0.30	0.36
Wu et al. (2018)	Semi-Supervised	0.73	0.79	0.76	0.56	0.67	0.61
Garcia-Pablos et al. (2014)	Unsupervised	0.56	0.65	<b>0.61</b>	0.32	0.43	0.37
Giannakopoulos et al. (2017)	Unsupervised	<b>0.74</b>	0.42	0.54	<b>0.67</b>	0.31	0.42
Venugopalan and Gupta (2020)	Unsupervised	0.53	<b>0.82</b>	0.64	0.45	<b>0.69</b>	<b>0.55</b>
UNATE	Unsupervised	0.70	0.69	<b>0.69</b>	0.51	0.43	0.47

Table 5: ATE with an automatically annotated dataset.

Method	Sem2014-Rest			Sem2014-Laptop		
	P	R	F1	P	R	F1
SemEval <sub>S</sub>	0.52	0.43	0.47	<b>0.44</b>	<b>0.30</b>	<b>0.36</b>
SemEval <sub>U</sub>	<b>0.56</b>	<b>0.58</b>	<b>0.57</b>	0.33	0.19	0.24

able alternative when no annotation is available.

As future work, we plan to use UNATE to automatically generate a domain lexicon that can be used in tasks such as aspect classification, clustering, and visualization. Finally, like Tubishat et al. (2021), we could explore a larger set of rules.

**Acknowledgments:** This work has been financed in part by CAPES Finance Code 001 and CNPq/Brazil.

## REFERENCES

- Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *CoRR*, abs/1607.04606.
- Brody, S. and Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *NAACL*, pages 804–812.
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Garcia-Pablos, A., Cuadros, M., Gaines, S., and Rigau, G. (2014). V3: Unsupervised generation of domain aspect terms for aspect based sentiment analysis. In *SemEval*, pages 833–837.
- Giannakopoulos, A., Musat, C., Hossmann, A., and Baeriswyl, M. (2017). Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. *CoRR*, abs/1709.05094.
- Han, X. (2009). Why can’t we dispense with the subject-predicate form without losing something more? *Florida Philosophical Review*, 9(2):79.
- Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177.
- International Telecommunication Union (2018). Measuring the information society report.
- Jakob, N. and Gurevych, I. (2010). Extracting opinion targets in a single and cross-domain setting with conditional random fields. In *EMNLP*, pages 1035–1045.
- Jorij, A. (2017). Global ecommerce report 2017.
- Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Liu, B. (2015). *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.
- Liu, P., Joty, S. R., and Meng, H. M. (2015). Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*, pages 1433–1443.
- Luo, H., Li, T., Liu, B., and Zhang, J. (2019). DOER: dual cross-shared RNN for aspect term-polarity co-extraction. In *ACL*, pages 591–601.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.
- Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., and Androutsopoulos, I. (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval*, pages 486–495.
- Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014). Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval*, pages 27–35.
- Qiu, G., Liu, B., Bu, J., and Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.
- Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

- Tubishat, M., Idris, N., and Abushariah, M. (2021). Explicit aspects extraction in sentiment analysis using optimal rules combination. *Future Generation Computer Systems*, 114:448–480.
- Venugopalan, M. and Gupta, D. (2020). An unsupervised hierarchical rule based model for aspect term extraction augmented with pruning strategies. *Procedia Computer Science*, 171:22–31.
- Wu, C., Wu, F., Wu, S., Yuan, Z., and Huang, Y. (2018). A hybrid unsupervised method for aspect term and opinion target extraction. *Knowledge-Based Systems*, 148:66–73.
- Xu, H., Liu, B., Shu, L., and Yu, P. S. (2018). Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*, pages 592–598.
- Zhang, L. and Liu, B. (2014). Aspect and entity extraction for opinion mining. In *Data mining and knowledge discovery for big data*, pages 1–40. Springer.

