UWaterloo at SemEval-2016 Task 5: Minimally Supervised Approaches to Aspect-Based Sentiment Analysis

Olga Vechtomova and **Anni He**
University of Waterloo, Canada
{ovechtomova, anni.he}@uwaterloo.ca

Abstract

This paper describes our system for Aspect-Based Sentiment Analysis (ABSA), task 5 of SemEval 2016. To conduct sentence level ABSA, we employed minimally supervised approaches for each type of extracted information. The system uses Word2Vec to derive word semantic similarities, and relies on external review corpora as training data. The results of the 2016 evaluation are discussed and suggestions for improvements are given.

1 Introduction

In this paper, we describe our system for Aspect-Based Sentiment Analysis (ABSA), Task 5 of SemEval 2016. The task involves classifying consumer reviews into existing aspect category labels (slot 1), identifying the opinion target expression (OTE) corresponding to the aspect category label (slot 2), then assigning sentiment polarity expressed about the aspect category (slot 3). The aspect category (slot 1) consists of two parts: Entity (e.g. food, ambience, restaurant) and Attribute (e.g. price, quality).

Our system tackles the problem in three stages addressing each slot individually for the restaurant domain and laptop domain (slot 3 only). The strengths of the system include no lexicon resources and minimal amount of labeled corpora. For the aspect category identification and OTE extraction, the approach is semi-supervised utilizing Word2Vec models (Mikolov et al., 2013) to derive word semantic similarities. The lists of words ranked by their semantic similarities to each entity and each attribute are generated for the aspect category identification. For the polarity assignment, we build from the approach of (Vechtomova et al., 2014) using corpora of consumer-rated electronics and restaurant reviews. The method does not require any sentiment labels at the word or sentence level or sentiment lexicons.

The paper describes each stage of the system in detail in section 2. In section 3, results of the system are presented and conclusions are drawn in section 4.

2 System Description

2.1 Corpora

At different stages of our system we used various resources we automatically generated from two corpora, described below. In both of these corpora we used only the original text of the review and the overall review ratings assigned by the consumer.

For the Restaurant domain, we used a corpus of 157,865 restaurant reviews from one of the major business review websites (Vechtomova, 2014). The collection contains reviews for 32,782 restaurants in the United States. The average number of words per review is 64.7. We will refer to this corpus as the Restaurant corpus throughout the paper.

For the Laptop domain, we used a subset of the Amazon corpus (Jindal and Liu, 2008), containing 138,504 reviews of products in the category Consumer Electronics. We will refer to this corpus as the CE corpus throughout the paper.

2.2 Aspect Category Detection

For this slot, the goal is to identify all entity and attribute pairs expressed by the given review sentence.

Proceedings of SemEval-2016, pages 372–377,
San Diego, California, June 16-17, 2016.

Prior to the two-stage entity and attribute identification pipeline, we obtain ranked lists of words for each entity type (entity ranked lists) and some of the attribute labels (attribute ranked lists). Extraction of OTEs by the process described in section 2.3 must be done prior to the feed into the category identification step.

We participated in Slot 1 in the Restaurant domain only. The entity ranked lists are generated from a Word2Vec model trained on the Restaurant corpus with a vocabulary size of 22118 words. The ranked lists are generated using a semi-supervised approach by using OTEs from the 2015 ABSA Restaurant training set as seed words. A list of top *n* similar words (*n*=500) is computed for each seed word by cosine similarity, then the lists are merged to form one ranked list per entity. The top *n* similar words parameter is chosen empirically to be sufficiently large relative to the vocabulary size in order to generate a fair-sized merged list. A weight of 5 is applied to the seed words to boost their rank in the merged list. The similarity scores in the merged lists are normalized to a range of [0,1]. When the entity ranked lists are utilized in the entity identification step, we apply a similarity threshold of 0.05 to discard any words below the threshold. The trained Word2Vec model only detects single words, therefore the ranked lists contain no multiword phrases.

The entity identification is based on the previously extracted OTEs for each review sentence and the generated entity ranked lists from Word2Vec. For each identified OTE of each sentence, its cosine similarity score is obtained from each entity ranked list. The entity with the highest similarity to the OTE is assigned. The OTE is further tokenized if it is a MWU. Each token goes through the same entity assignment as above, then the entity with the highest similarity is assigned to the entire MWU-OTE. If the OTE does not appear in any of the ranked lists, then the OTE is discarded, and no entity or attribute is assigned.

In order to identify the attribute component of the entity-attribute pair, we first generate a set of seed words from the ABSA 2015 training set. This process is done for each entity category that has more than one attribute (e.g. FOOD, DRINKS). For every entity-attribute pair (e.g. FOOD#PRICES) we parse all sentences in the training set that are labeled with this pair, and extract all words that have JJ (adjective) and VBN (verb, past participle) POS tags, have frequency greater than one, and have one of the following relationships with at least one noun or pronoun: adjectival modifier, relational clause modifier, nominal subject or passive nominal subject. These attribute seed words are then used to obtain a ranked list of words (attribute ranked list) using Word2Vec in a similar process as the entity ranked lists.

After an entity is identified, attributes are assigned based on the information from the review sentence. Each review sentence is tokenized, singularized with stop words removed. Intersection of all the tokens is computed with each attribute ranked list. The attribute with the highest similarity is assigned. If none of the ranked list attributes are assigned, then the attribute with the highest prior probability in the ABSA 2015 training set is assigned.

2.3 Extraction of Opinion Target Expressions

We approach the problem of opinion target extraction with a semi-supervised method. The opinion target expression (OTE) is Slot 2 in the ABSA task, and is only set for the Restaurant domain.

First, seed words are extracted for each Entity from the set of opinion target expressions in the ABSA 2015 training dataset. These are ranked by frequency, and top *n* are used. We evaluated different values for *n*, with 5 showing the best results on the ABSA 2015 test data. For each seed word we generate a ranked list of words using Word2Vec models, and create a merged entity ranked list as explained in Section 2.2.

The next stage in the OTE extraction process is identification of the boundaries of nominal multiword units (MWUs) representing OTEs in the sentences. We use an algorithm developed by (Vechtomova, 2014) that builds MWUs in a bottom-up manner. Each sentence in the dataset is parsed using Stanford dependency parser (De Marneffe et al., 2006). The process begins by identifying all single nouns, which are governors in at least one syntactic dependency relation. By following a set of syntactic rules, we merge each of these nouns with the adjacent words (e.g. adjectives, other nouns) in an iterative manner, one at a time. The merging proceeds in two stages: in the first stage, the algorithm iteratively merges pairs of words that have either adjectival modifier (e.g. salty fish), nominal compound modifier (e.g. garden salad), or possession modifier (e.g. Chef's choice) dependency relationship. In the second stage, it iteratively merges pairs of words that have a prepositional modifier (e.g. fish with rice) or a conjunct (e.g. fish and chips) relationship. In addition to applying syntactic rules, before each merge, the algorithm calculates Normalized Pointwise Mutual Information (NPMI) between two strings to be merged. Only if all the syntactic rules are satisfied and the NPMI is above the specified threshold, the two strings are merged. The algorithm is described in detail in (Vechtomova, 2014). The output of this stage is a set of nominal MWUs (which may include single nouns) for each sentence.

Finally, the system calculates a score for each MWU–Entity pair by summing the scores of the matching words in the corresponding entity ranked list. If an MWU has a score (*s*) of zero in all categories, it is discarded. If the score is $0 < s < 0.1$, the target is changed to NULL. If $s \geq 0.1$, it is retained as OTE.

2.4 Polarity Identification

We address the problem of polarity detection with an approach that only uses the texts of consumer reviews and overall review ratings assigned by consumers. The attractiveness of the method is that it does not require any sentiment lexicons or sentiment labels at the word or sentence levels. We extend a method described in (Vechtomova et al., 2014). Due to the growing popularity of online product/business reviewing, there exist vast repositories of reviews in many categories of consumer products and businesses. Most of the online review sites require users to rate products numerically on some scale (e.g. a 5-star rating scheme in Amazon). We leverage these resources in our methods. The Restaurant corpus that we used has ratings on a 10-point scale, while the CE corpus, on a 5-point scale. For the Restaurant domain we generate a set of negative reviews by pooling all reviews with ratings of 1 and 2, and a set of positive reviews by pooling all reviews with the rating of 10 from the Restaurant corpus. In the Laptop domain, reviews with ratings 1 and 2 were used for the negative set, while reviews with the rating of 5 for the positive set.

In summary, the process consists of the following steps. First, two vectors of context features are created for each adjective or verb (*w*) that has a dependency relationship with a noun or personal pronoun. One vector *posV* is built based on all occurrences of *w* in the positive set, and the second vector *negV* is built based on its occurrences in the negative set. Next, polarity of the occurrence of *w* in a previously unseen sentence *s* is determined by building a vector *evalV* based only on the context of *w* in *s*, and computing a pairwise similarity of *EvalV* to *posV* and *negV*.

In more detail, the following steps are performed on each of the two sets: positive and negative. Each sentence in a positive/negative set is processed by using a dependency parser in the Stanford CoreNLP package. In each sentence, we first locate all nouns or personal pronouns (*n*). Then, for each *n*, its dependency triples with adjectives and verbs (*w*) are extracted, where the dependency relation is either an adjectival modifier, nominal subject, passive nominal subject, direct object or relative clause modifier. An example of a dependency triple is *nsubj(pizza, hot)*, where pizza is a governor, while hot is a dependent. We also identify dependency relations of adjectival complements, clausal complements and open clause complements and merge them with the nominal subject relationship sharing the same verb, e.g. *nsubj(menu, looks)* and *acomp(looks, great)* are merged into *nsubj_acomp(menu, look_great)*.

We also created a set of rules to determine whether the context containing an instance of *w* is negated or not. For each occurrence of *w* the following information is recorded: negation (1 – *w* is negated; 0 – *w* is not negated); dependency relation of *w* with *n*; *w* lemma (output by Stanford CoreNLP). These three pieces of information form a pattern *p*, e.g., "negation=0; amod; better". A context feature vector (*PosV* and *NegV*) is built for each *p*, as follows: for each instance of *w* matching this pattern in the corpus (positive or negative, respectively) we extract all dependency relations containing it. Each of them is transformed into a context feature *f* of the form:"lemma; Part Of Speech (POS); dependency relation". For instance, if adjective "hot" occurs in a dependency triple *advmod(hot, too)*, the following feature is created to represent "too" and its syntactic role (adverbial modifier) with respect to "hot": "too, RB, advmod". We also build

Pattern	Context
NEGATION=0, nsubj, cold, JJ	also, RB, advmod be, VBZ, cop bread, NN, nsubj horrible, JJ, conj_and it, PRP, nsubj not, RB, conj_and particularly, RB, advmod quickly, RB, advmod tired, JJ, conj_and

Table 1: Example of a Pattern and its Context Features.

composite features by joining up to four dependency relations by traversing the dependency graph.

For each f we record its frequency of co-occurrence with p (used as TF in Eq. 1). Table 1 contains an example of a pattern and a subset of its context features. The same algorithm is used to build a feature vector (*EvalV*) for each pattern extracted from each sentence in the ABSA test dataset.

Next, for each pattern p found in the test sentence, we compute pairwise similarity between its $Eval_p$ vector and $posV_p$ and $negV_p$ respectively. For the purpose of computing similarity we evaluated two similarity functions: BM25 Query Adjusted Combined Weight (QACW) (Sparck Jones et al., 2000) and TF.IDF. QACW was first used to compute term-term similarity in (Vechtomova and Robertson, 2012). The $EvalV_p$ is treated as the query, while $posV_p$ and $negV_p$ as documents (V_p in Eq. 1)

$$Sim(EvalV_p, V_p) = \sum_{f=1}^{F} \frac{TF(k_1+1)}{K+TF} \times QTF \times IDF_f \tag{1}$$

Where: F – the number of features that $EvalV_p$ and V_p have in common; TF – frequency of feature f in V_p; QTF – frequency of feature f in $EvalV_p$; K = *k1((1b)+bDLAVDL)*; k_1 – feature frequency normalization factor; b – *Vp* length normalization factor; *DL* – number of features in V_p; *AVDL* – average number of features in the vectors V for all patterns p in the training set (positive or negative). The b and k_1 parameters were set to 0.1 and 50 respectively, as these showed best performance on the ABSA 2016 Restaurant training dataset. The *IDF* (Inverse Document Frequency) of the feature f is calculated as IDF_f = log(N/n_f), where, n_f – number of vectors V in the training set (positive or negative) containing f; N – total number of vectors V in the training set.

Finally, if $Sim(EvalV_p,\ posV_p) > Sim(EvalV_p,\ negV_p)$, we assign positive polarity to the given instance of p in the test sentence, otherwise, negative.

Polarity detection in ABSA is done as a Phase B evaluation, i.e. after the Phase A goldset (containing OTEs and Entity-Attribute pairs) is released. The next steps differ for the Restaurant and Laptop domains, since the former has OTEs, while the latter does not.

For the Restaurant domain, for each OTE in the goldset, the method determines the majority polarity based on all p that contain the OTE or any of its words if the OTE is an MWU. If there is equal number of positive and negative cases, then neutral polarity is assigned. If no word matching OTE or any of its constituents is found by our method, then average polarity is calculated based on the current sentence. This method was also used for cases with NULL OTEs. If no words have been extracted by our method for the given sentence, then average polarity based on the entire review is calculated.

For the Laptop domain, we tried two methods. In Method 1 (submitted), the average polarity is calculated based on the entire sentence, and if no opinion words have been extracted for the given sentence, then we assign the average polarity calculated based on the entire review. In Method 2, for each Entity type in the goldset (e.g. Battery, Memory) we first built a set of related words using Word2Vec given the entity name as the seed word. Top 500 words ranked by cosine similarity to the seed word were used. For each Entity-Attribute pair in each sentence of the Phase A goldset, we first determine if any opinion target extracted by our method matches a word in the ranked list for the corresponding entity. If it does, then we assign its polarity to the corresponding Entity-Attribute tuple(s) for that sentence. If no matching word for an entity is found, then we calculate polarity based on all words that did not match any other category. If no such words exist, we default to Method 1.

3 Results

In this section we present results of the runs on the ABSA 2016 training and test datasets.

Tables 2 through 6 show results for entity-attribute identification and OTE extraction. The training set performs better than the test set in all

Dataset	Precision	Recall	F-Measure
TRAIN 2016	0.5334	0.5168	0.5250
TEST 2016	0.4980	0.4966	0.4973

Table 2: Slot 1 (Aspect category) results (restaurant).

Dataset	Precision	Recall	F-Measure
TRAIN 2016	0.6331	0.6676	0.6499
TEST 2016	0.6100	0.6457	0.6273

Table 3: Slot 1 results (restaurant) evaluating on entity only.

three evaluations. This can be attributed to the fact that the entity/attribute ranked lists were built using Word2Vec models generated from the 2015 training set which is a subset of the 2016 training set.

Motivation of the ranked lists process stems from the minimal amount of annotated training data. It is hypothesized that the OTEs from the training data provide strong signals for entity labels. For example, if the OTE is “waiter”, then the entity is most likely “SERVICE”. Once the entity is identified, attribute labels are assigned using tokenized words in the sentence besides the OTE. It is thought that the other words in the sentences are better predictors for attribute labels. For example, the review sentence “the food was not worth the price” has OTE “food” and the entity-attribute label “FOOD#PRICES”. The attribute label can easily be determined from the word price ranked at the top of the #PRICES attribute ranked list. There are several consequences impacting the performance of the system as a result of the two-stage design as explained below.

A comparison can be made between the results for entity-attribute pair identification (Table 2) and entity-only identification (Table 3) to show progression of the system performance between the two stages. The system performs more than 10% better in the entity-only case for both test and train datasets. This is expected because attribute identification using tokenized words as features from the review sentence is less reliable than using the OTE as a single feature.

Another weakness of the system is in the aspect category identification for NULL OTEs. The entity/attribute ranked lists procedure is not designed for NULL targets due to the lack of OTEs. Currently the NULL target sentences require a different process whereby words in the sentence are tokenized and compared to both the entity ranked lists and attribute ranked lists to identify the label pair. The system evaluated without the NULL targets shows about a 4% improvement in F-measure for the test and train datasets. To improve the system, the NULL target aspect category identification may need to use additional features from the previous and following sentences of the review.

Dataset	Precision	Recall	F-Measure
TRAIN 2016	0.5000	0.6616	0.5696
TEST 2016	0.4822	0.6273	0.5453

Table 4: Slot 1 results (restaurant) with NULL OTEs removed.

Dataset	Precision	Recall	F-Measure
TRAIN 2016	0.4826	0.7143	0.5760
TEST 2016	0.4804	0.7026	0.5707

Table 5: Slot 2 (Opinion Target Expression) results (restaurant dataset).

Dataset	Precision	Recall	F-Measure
TRAIN 2016	0.3582	0.3934	0.3750
TEST 2016	0.3240	0.3667	0.3440

Table 6: : Slots 1&2 results (restaurant dataset).

Table 5 shows the results of the evaluation for Slot 2 (OTE). While recall is good, the precision is not high. A better method of computing similarity between a candidate OTE and the Entity categories is needed. Also, a better method to detect NULL targets is needed. Out of 198 NULL OTEs in the test gold set, our method missed 164. As expected, the combined results of slots 1& 2 is rather poor as shown in Table 6.

Our approach to polarity identification showed promising results. QACW performed slightly better on both the training and test Restaurant datasets. As can be seen from Table 7, Method 1 in the laptop polarity task performed better than Method 2. This is likely due to the fact that using one word per entity as seed is not sufficient to generate a good set of entity-related words. In the future, it will be good to explore how to generate a better set of seed words semantically representing the entity.

In our submission for the Laptop Slot 3, we did not use features based on adjectival, clausal and open clausal complements merged with the nominal subject modifier, e.g. *nsubj_acomp(menu, look_great)* (for the description of features see Section 2.4). When we added these relations to the

Dataset/Method	Accuracy
TRAIN 2016 – REST TF.IDF (submitted)	0.7801
TRAIN 2016 – REST QACW	0.7825
TEST 2016 – REST TF.IDF (submitted)	0.8033
TEST 2016 – REST QACW	0.8056
TRAIN 2016 – LAPT TF.IDF, Method 1	0.7417
TRAIN 2016 – LAPT TF.IDF, Method 2	0.7383
TEST 2016 – LAPT TF.IDF, Method 1 (submitted)	0.7129
TEST 2016 – LAPT TF.IDF, Method 2	0.7066

Table 7: Slot 3 (polarity) results.

Dataset/Method	Accuracy
TRAIN 2016 – LAPT TF.IDF Method 1	0.7634
TEST 2016 – LAPT TF.IDF Method 1	0.7316

Table 8: Slot 3(polarity) generated by adding adjectival, clausal and open clausal complements to the feature set.

feature set, the performance improved by 2.9% and 2.6% on the training and test datasets respectively (see Table 8).

One of the major reasons why the polarity method did not perform better is that we adapted a method that was designed for identifying two categories (positive and negative) to the ABSA task, which has three categories (positive, negative and neutral). This is evident when we break down the results by polarity category: the F-measure in the Restaurant domain for Positive, Negative and Neutral categories is 0.8815, 0.6439 and 0.1194 respectively. For the Laptop it is 0.8004, 0.5727 and 0 respectively. Further work is needed on better identification of neutral cases.

4 Conclusion

In this paper, we described our system for aspect-based sentiment analysis used for aspect category identification, extraction of opinion target expression and polarity identification. Our polarity identification method only leverages available consumer reviews with the associated overall review ratings assigned by the consumer. It does not require any sentiment lexicons or sentiment annotations in the texts of the reviews. The polarity identification and OTE extraction showed promising results among other systems having performed within one percent of the mean scores of all participating systems in the restaurant and laptop domains. To advance our system, we identified weaknesses of the aspect category identification and hopeful next steps to improve the results by refined treatment of NULL target sentences.

References

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 219–230. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

K Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information Processing & Management*, 36(6):809–840.

Olga Vechtomova and Stephen E Robertson. 2012. A domain-independent approach to finding related entities. *Information Processing & Management*, 48(4):654–670.

Olga Vechtomova, Kaheer Suleman, and Jack Thomas. 2014. An information retrieval-based approach to determining contextual opinion polarity of words. In *Advances in Information Retrieval*, pages 553–559. Springer.

Olga Vechtomova. 2014. A method for automatic extraction of multiword units representing business aspects from user reviews. *Journal of the Association for Information Science and Technology*, 65(7):1463–1477.