UMDuluth-CS8761-12: A Novel Machine Learning Approach for Aspect Based Sentiment Analysis

Akshay Reddy Koppula, Ranga Reddy Pallelra, Ravikanth Repaka, Venkata Subhash Movva
Department of Computer Science
University of Minnesota Duluth
320 Heller Hall
1114 Kirby Drive
Duluth, MN 55812-2496, USA
{koppu001, palle015, repak003, movva002}@d.umn.edu

Abstract

This paper provides a detailed description of the approach of our system for the Aspect-Based Sentiment Analysis task of SemEval-2015. The task is to identify the Aspect Category (Entity and Attribute pair), Opinion Target and Sentiment of the reviews. For the In-domain subtask that is provided with the training data, the system is developed using a supervised technique Support Vector Machine and for the Out-of-domain subtask for which the training data is not provided, it is implemented based on the sentiment score of the vocabulary. For In-domain subtask, our system is developed specifically for restaurant data.

1 Introduction

With the increase in usage of internet, most of the users record their experiences of a particular product or item in the form of online reviews. Users might express their opinion about many different aspects of an item in a review.

While most of existing systems try to extract the overall polarity of a sentence, Semeval 2015 conducted a task on Aspect-Based Sentiment Analysis and the requirement was to extract entities (e.g., Food, Price, Service for Restaurant data), attributes(e.g., Quality, Style) for each sentence and then to determine the polarity for each entity-attribute pair.

The fajitas were delicious, but expensive.

In the above example, there are two opinions.The first opinion has FOOD#QUALITY as the entity-attribute pair with positive polarity and second has FOOD#PRICES with negative polarity. The target for both these opinions is fajitas. Since there are two opinions with two different polarities, it is useful to identify entities, attributes and targets for each sentence.

Our system tries a new approach of trying to split the sentence to find out more than one opinion in a sentence. Initially, all the unnecessary words are removed and then sentences are split in a way such that each split sentence has an opinion. These split sentences are given to a classifier for identifying entities and attributes. Later, these entities are used to extract opinion targets. Polarity is found using a classifier and voting mechanisms.

The rest of the paper is structured as follows: Section 2 presents the description of SemEval-Task Aspect-Based Sentiment Analysis. Section 3 presents the description of our system. Section 4 discusses the results of our system and analyze them. Section 5 presents a conclusion to the paper.

2 SemEval Task Description

The SemEval Task is divided into two subtasks.

2.1 Subtask 1

Following are the slots in the Subtask 1

2.1.1 Slot 1 - Aspect Category (Entity and Attribute)

It specifies the category of the domain to which the review refers. Aspect Category contains the Entity#Attribute pair of the review.

Entity is the aspect of the domain for which an opinion is expressed in the given review. Attribute is

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 742–747,
Denver, Colorado, June 4-5, 2015.

the quality or feature the review refers to and this is dependent on the Entity.

Great for a romantic evening, but over-priced. {Entity#Attribute} –>{Ambience#General, Restaurant#Prices}

2.1.2 Slot 2 - Opinion Target Expression

Opinion target is the target word in the review on which an opinion is expressed.

The Shrimp was awesome, but over-priced. {Entity#Attribute, Target} –>{Food#Quality, "Shrimp"}, {Food#Prices, "Shrimp"}

2.1.3 Slot 3 - Sentiment Polarity

Every Entity#Attribute pair obtained from sentence should be assigned a polarity of either positive, negative, or neutral depending on the sentiment expressed by the user.

2.2 Subtask 2

The task is to find out the polarity for each entity, attribute pair of the review which will be provided in the test data. No training data is provided for this task.

Further details of the task description are provided in (SemEval, 2015).

3 System Description

This system has been developed specifically for Restaurant data for subtask 1 and it is constrained for subtask1, unconstrained for subtask2.

The different stages in which the system proceeds are described in respective subsections. Most of them use an SVM classifier for predictions. This classifier is described extensively in subsection 3.9.

3.1 Subjectivity Classification

There are two types of sentences: Subjective and Objective. Subjective sentences are based on personal opinions. Objective sentences are factual and observable. Linear SVM classifier is used to categorize the subjective and objective sentences.

Training: Training sentences that have opinions are given a constant value and that do not have opinions are given another constant value. Using this binary classification model, a Linear SVM classifier is trained using unigram Bag of words feature for the given training dataset.

Testing: The trained Linear SVM classifier is used in predicting the test sentences with subjective information.

Only these predicted subjective test sentences are considered for further processing.

3.2 Clean the Sentence

The main functionality of this module is to remove unnecessary words and modify the sentence in a way that helps in splitting of the sentence in next stage. Specifically, clean the sentence to remove the articles (a, an, and the) and append ',' before 'but', 'at', and 'with' words. This addition of ',' will help to split the sentence in the next processing stage. A ',' is prepended to 'at' if it is preceded by an adjective and to 'with' if any adjective exists in any of the three previous words. These rules are extracted by observing the training data.

The food is great and they have a good selection of wines at reasonable prices.

In the above example, 'at' will be prepended with ',' and 'a' will be removed.

3.3 Split the Sentence

Each sentence may contatin multiple opinions and we believe that divison of sentence into subsentences will help in making these predictions better. Observations from the training data led to the understanding that ',' and 'and' are used frequently to express multiple opinions in one sentence and hence these tokens are used to divide the sentence. Some words like 'at', 'but', 'with' are also being used to express multiple opinions and as ',' has been appended in the previous stage this helps in splitting these sentences also properly.

Below are some examples on this splitting

The food is great and they have a good selection of wines, at reasonable prices.

Split sentences: 1) The food is great 2) they have a good selection of wines 3) wines at reasonable prices

Thalia is a beautiful restaurant, with beautiful people serving you, but the food doesn't quite match up

Split sentences: 1) Thalia is a beautiful restaurant 2) with beautiful people serving you 3) but the food doesn't quite match up

If a split sentence has an adjective but does not have a noun, then the noun(s) in the previous split sentence will be appended to current split sentence.

Similarly, if the split sentence has a noun but does not have an adjective, then the adjective from the previous split sentence will be appended to current split sentence.

We love food, drinks, and atmosphere

Split sentences: 1) we love the food 2) love drinks 3) love atmosphere

In contrast, if a split sentence does not have both noun and adjective then append this split sentence to the previous split sentence.

3.4 Identify Entities

In this section, we use the output from the split sentences. Since there can be multiple split sentences and entities, each split sentence has to be matched with it's corresponding entity. For Example:

Pizza is delicious, ambience is bad.

This example has two different entities: Food, Ambience. After splitting the sentences, assigning an entity to its respective part of sentence is important:

Pizza is delicious- Food
ambience is bad - Ambience

To assign each split sentence with it's respective entity, Wordnet is used. Find the similarities between the words in each split sentence and each entity using wordnet. For each entity, assign a split sentence to which the most similar word for that entity belongs to.

After each split sentence has been assigned to it's respective entity, the words from that split sentence whose parts of speech are among nouns, verbs, adjectives or adverbs are extracted and given as input to SVM. Use the linear SVM model as described in subsection 3.9 to predict the entity.

3.5 Identify Attributes

All the nouns, verbs, adjectives, adverbs for each particular attribute are extracted from the training data. Each attribute along with their respective extracted words are given as input to the SVM Classifier. Use the linear SVM model as described in subsection 3.9 to predict the entity.

Apart from this process some predefined lexicons from the training data are extracted manually. For example, if there are words like money or price in the sentence then it is likely that the sentence is talking about the attribute price. Words like these will, in almost all of the cases, belong to attribute 'price', these were extracted manually from training data as only a few of them were present. Upon the encounter of such words in the test data, the attribute associated with them is assigned. If none of these predefined words are encountered, then SVM classifier is used as described above.

3.6 Extract Opinion Targets

In order to extract opinion targets, The following procedure is applied for finding targets where the entities extracted in previous section are among 'Food', 'Restaurant', 'Drinks', 'Location'.

Training: Targets are found out based on Entities and most of them are nouns with a few being adjectives. Each entity has some nouns that will not be the targets. For example, a noun such as 'food' will not be the target for the 'restaurant' entity. In the training data, for each entity, identify all the nouns, adjectives that are not targets. Also, identify the target words for each entity. All these extracted words are used for finding the targets in a test sentence.

Testing: If a given test sentence has one of target words extracted in training, return that target. If not, remove all the non-targets in the sentence that were extracted from training. After this removal, if there are no more nouns in the sentence, then return the target as NULL. If more nouns exist in the sentence, then return the largest substring of the consecutive nouns and adjectives. If the entity is restaurant then return the proper noun as the target if it is preceded with 'at' or 'to'.

For Entities (Ambience and Service): For sentences that has Ambience and Service as the identified entities, a different approach is employed to extract opinion targets: A vocabulary of targets is constructed from the training data and is given as input to a classifier along with the corresponding sentences and their labels. This classifier is described in subsection 3.9

3.7 Sentiment Polarity

From the given sentence, all noun(s), adjective(s), adverb(s), and verb(s) are extracted and given as input to the classifier to predict the polarity as either positive, negative or neutral. Usually classifiers can have multiple parameters. So, using the grid search method from Scikit Learn package, different parameters such as unigrams, bigrams and trigrams are tested and it was observed that trigrams resulted in better performance of the classifier. Hence trigrams are used whenever needed.

Two different techniques are tried for the classification of the given training data:

1. All unique tri-grams in the training sentences are identified and TF-IDF values are calculated for these trigrams. Count Vectorizer and TF-IDF transformer from 'Scikit Learn' package are used to extract the BoW features from the sentences.

2. Categorical Probability Proportion Difference (CPPD) (CPPD, 2012)

When compared to CPPD, BoW features resulted in higher accuracy. But, CPPD model might work good for other domains. To predict the polarity for test sentences, voting (Brill et al., 2001) among classifiers is used. The classifiers used in the voting procedure are Naive Bayes, Linear SVC, and Logistic Regression.

By experimentation it is observed that Naive Bayes has a good "negative recall" when compared to voting. This experiment was helpful in deciding the polarity of a sentence. If Naive Bayes predicts negative, then the polarity for that sentence is assigned as negative, else it is assigned as the value predicted from voting.

3.8 Out-of-Domain

In the out-of-domain subtask, no training data or knowledge about the domain would be provided or used to predict the polarity of the given test sentence.

The steps taken in this task are:

1. Splitting of the test sentence into sub sentences is done based on the number of opinions it has. From the split sentences, words with parts of speech tag as noun, verb, adjective, or adverb are extracted.

2. Polarity is predicted using two tools Sentiwordnet and Pattern. The nearest opinion word (adjective, adverb, or verb) to the target word is identified and polarity is found out for this word and is set as the polarity for the sentence. If this word does not have polarity, then the average polarity score for the remaining opinion words in the sentence is calculated and is set as the polarity for the sentence. Apart from these two predictions, Pattern tool is also used to predict the polarity for the complete sentence.

3. Voting is applied to these three predictions and the output of this would be the final polarity for the sentence.

3.9 Linear SVM Model

The steps involved in training the Linear SVM classifier for our system are described below:

Features are extracted using unigram Bag of words (BoW), Tf-Idf, Univariate feature selection model (Scikitlearn, 2011). An optimized regularization parameter (C value) is also used.

Train the Classifier: With the help of all the above mentioned parameters, the classifier is trained for the given training dataset. Linear SVM model with BoW as features is trained using the multi-class classification method for the given training dataset.

Predict the Label: The Linear SVM classifier predicts the output label for each test sentence by using the C value identified in the Cross-validation step.

4 Results and Analysis

Our system was trained on 1314 review sentences and tested on 685 review sentences for sub task 1. Evaluations are done for slot 1, slot 2, slot 1 and slot 2, slot 3, and subtask 2. The results for each of them are provided in the tables. Each table has the scores of the best team, our system and the SemEval baseline. Table1 provides the results for slot 1 in which our system is ranked 2nd among all the constrained systems participated in the task and is ranked 3rd among all the participating systems. As our system for subtask 1 is constrained, all our scores are compared only with the best constrained system. For subtask 2, the best score among all systems is considered.

As evident from the results, extraction of opinion targets can be attributed to the failure of both slot 2 and slot 1 & slot 2. We suspect that the reason behind this could be our concentration on finding those

Team	F1-Score
Best	61.94
UMDuluth-CS8761-12	57.20
Baseline	51.32

Table 1: Slot 1.

Team	F1-Score
Best	66.91
UMDuluth-CS8761-12	50.36
Baseline	48.06

Table 2: Slot 2.

Team	Accuracy
Best	85.84
UMDuluth-CS8761-12	71.38
Baseline	71.68

Table 3: Subtask 2.

Team	F1-Score
Best	42.72
UMDuluth-CS8761-12	32.60
Baseline	34.44

Table 4: Slot 1 and Slot 2.

Team	Accuracy
Best	75.50
UMDuluth-CS8761-12	71.12
Baseline	63.55

Table 5: Slot 3.

words that are non-targets rather than on trying to find words that should be targets. If a noun is not a target in one sentence, it doesnt mean that it cannot be a target in any sentence having similar entity.

5 Conclusion

Overall, our system performed well especially in slot 1 and slot 3. Identifying the number of opinions that each sentence might express is an important step to be taken, which we have achieved by splitting the sentence so that each split sentence can express an opinion. Applying supervised machine learning techniques on these split sentences resulted in a much better predictions compared to the complete sentences.

Acknowledgments

We would like to take this opportunity to thank our professor Dr. Ted Pedersen for encouraging us to participate in this task and for his guidance and advice.

References

Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 122825–2830, 2011.

Hu, Minqing and Liu, Bing Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM 168–177,2004

Maria Pontiki, Dimitrios Galanis, Haris Papageogiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. *SemEval-2015 Task 12: Aspect Based Sentiment Analysis* In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado.

Agarwal, Basant, and Namita Mittal. 2015. *Categorical probability proportion difference (CPPD): A feature selection method for sentiment classification.* Proceedings of the 2nd Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2012), COLING. 2012.

Tan, Songbo, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. *Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis.* ECIR, LNCS 5478, pp. 337–349, 2009.

Moghaddam, Samaneh, and Martin Ester. *Opinion digger: an unsupervised opinion miner from unstructured product reviews* Proceedings of the 19th CIKM, pp. 1825–1828, Toronto, ON, 2010.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio *Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach.* ICML, 2011.

Banko, Michele, and Eric Brill *Scaling to very very large corpora for natural language disambiguation.* Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, 2001.