Opinion Dominion - Jaecee Naylor

Phase 1 Project Report

IE Task

The IE task I am working to complete involves extracting an opinion's target expression (OTE), polarity, and category. The input I receive is the data from the SemEval 2016 task 5 restaurant data set in English. My program parses this input to create Review, Sentence, and Opinion objects corresponding to information in the provided XML. The dataset splits each review into sentences and also provides the manual annotated data for the opinion(s) found in the sentence. These correspond to the label types of "target" for the OTE, "category" for the entity#attribute (E#A) pair, and "polarity" for the polarity of the opinion. My system reads each sentence on a review and generates its own predictions for these variables stored in an Opinion object. The output of my system is the associated scores when comparing the predicted values to the expected values on an opinion. My system also creates six files, a file for each opinion label that is expected and predicted. The file format for each file is:

SentenceID: <sentence_id>
<Label>: <label_value>

Example:

SentenceID: 1090587:1
Polarity: positive

Resources List

1. Dataset(s): https://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools
2. NLTK (used for stopwords, wordnet, word and sentence tokenizers, lemmatizer, sentiment analysis, and part of speech tagging): https://www.nltk.org/
3. Stanza (used to gather relations between words in a sentence): https://stanfordnlp.github.io/stanza/

Technical Description

I have constructed different ways for my system to predict each of the labels specified above. I will start by explaining how I have designed my system to extract the OTE(s) from a sentence. For the simple baseline approach to find an opinion's OTE, I start by tokenizing the associated sentence and each of the sentence's words. I label the part of speech associated with each word. If there are two nouns that occur directly after one another, I store them as one (in case there are both a part of the target). For example, "pad thai" is not a compound noun so the system as is will not recognize them as related. Next, I use Stanza to get the relations between each word in the sentence (a form of dependency parsing). Once I have the dependency tree, I aim to only look at the dependencies between nouns and adjectives. This is because usually an opinion revolves around a noun and is supported by an adjective. After all the connections between nouns and adjectives are collected, the system uses each noun (with an adjective related to it) as one opinion. In the case that the system could predict more than the expected amount of opinions, the system creates an opinion with each of the predicted OTEs put together as the last opinion's OTE.

Next, I will explain how I am able to predict the category of each opinion. Since this is more of a labeling task (as is detecting the polarity of an opinion), I am not focusing a large chunk of my efforts on designing a complicated system to do this. My baseline system simply creates a dictionary containing every OTE that is associated with each category. This is created from the manually annotated data. Then, when my system is predicating the category of an opinion it checks to see if the predicted OTE is in the collected dictionary. If it is, the opinion is assigned the associated category. Otherwise, if the predicted OTE is not found in the dictionary, the category is assigned “NULL#NULL” which indicates that a guess was not made.

Lastly, I will describe how my system finds the polarity of each opinion. The system had pulled out potential emotion words related to each opinion when it found the OTE for the opinion. The system gathers each emotion word for one OTE. On each emotion word, the system gets the most popular synset for that word from NLTK’s wordnet’s synsets. Then it gets the positive score, negative score, and object score for the word from NLTK’s sentiwordnet. It uses these scores to compute a polarity score for the opinion. The polarity will be positive if the score is positive, negative if the score is negative, and neutral if the score is zero.

Evaluation

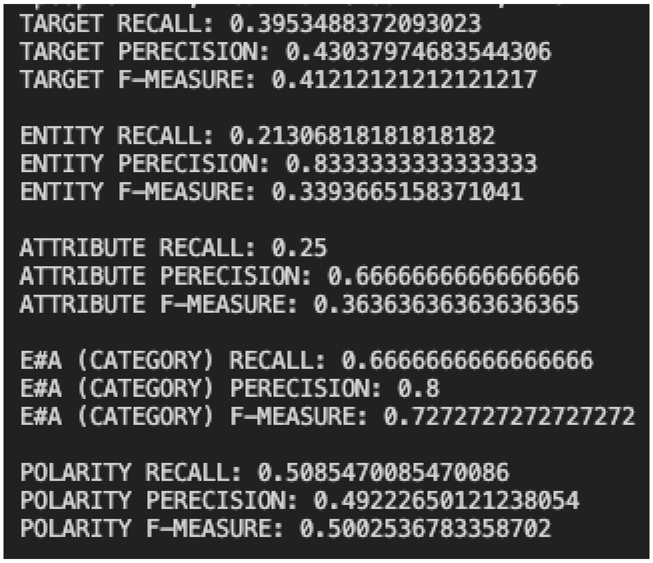
The performance of the system is evaluated using accuracy, precision, and f1-measure scores for each of the corresponding output labels. I also split up the category scores to also include pure Entity and pure Attribute scores since category output is a pair of these values. For OTE scores, I chose to calculate these scores based on word overlap between the predicted OTE and the expected OTE. This has its downsides as order does not matter and small words in common that are not important to the answer will boost the score falsely potentially.

For OTE recall, precision, and f1-measure scores are produced as follows:
Recall: The number of correct words generated by my system divided by the total number of words in the expected target answer.
Precision: The number of correct words generated by my system divided by the total number of words generated by my system for the target.
F-Measure = (2*Recall*Precision) ÷(Precision + Recall)

For Polarity, Category, Entity, and Attribute the scores are calculated as follows:
Recall: The average of all the classes recall scores that are calculated by the number of events where the system correctly identified class *i* divided by the number of instances where the expected class matched class *i*
Precision: The average of all the classes precision scores that are calculated by the number of events where the system correctly identified class *i* divided by the number of instances where the system predicted class *i*
F-Measure = (2*Recall*Precision) ÷(Precision + Recall)

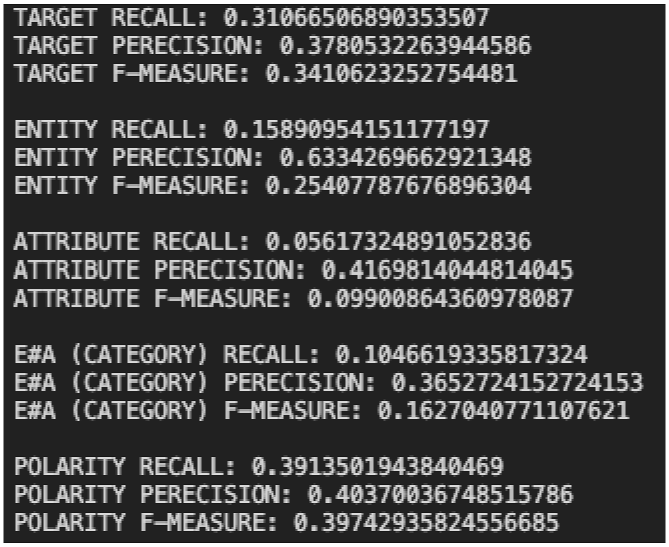
Opinion Dominion - Jaecee Naylor

Results on Trial Data:



```
TARGET RECALL: 0.3953488372093023
TARGET PERECISION: 0.43037974683544306
TARGET F-MEASURE: 0.41212121212121217

ENTITY RECALL: 0.21306818181818182
ENTITY PERECISION: 0.8333333333333333
ENTITY F-MEASURE: 0.3393665158371041

ATTRIBUTE RECALL: 0.25
ATTRIBUTE PERECISION: 0.6666666666666666
ATTRIBUTE F-MEASURE: 0.36363636363636365

E#A (CATEGORY) RECALL: 0.6666666666666666
E#A (CATEGORY) PERECISION: 0.8
E#A (CATEGORY) F-MEASURE: 0.7272727272727272

POLARITY RECALL: 0.5085470085470086
POLARITY PERECISION: 0.49222650121238054
POLARITY F-MEASURE: 0.5002536783358702
```

Results on Train Data:



```
TARGET RECALL: 0.31066506890353507
TARGET PERECISION: 0.3780532263944586
TARGET F-MEASURE: 0.3410623252754481

ENTITY RECALL: 0.15890954151177197
ENTITY PERECISION: 0.6334269662921348
ENTITY F-MEASURE: 0.25407787676896304

ATTRIBUTE RECALL: 0.05617324891052836
ATTRIBUTE PERECISION: 0.4169814044814045
ATTRIBUTE F-MEASURE: 0.09900864360978087

E#A (CATEGORY) RECALL: 0.1046619335817324
E#A (CATEGORY) PERECISION: 0.3652724152724153
E#A (CATEGORY) F-MEASURE: 0.1627040771107621

POLARITY RECALL: 0.3913501943840469
POLARITY PERECISION: 0.40370036748515786
POLARITY F-MEASURE: 0.39742935824556685
```