Haorui Chen-10407315-COMP34412-cwk2

## 1. Named-entity recognition

a) to use NLTK classifier for NER, function nltk.ne\_chunk() is used on POS tag result of tokenized texts. Function tree2conlltags() is used to flatten tree-structured result of ne\_chunk() and expose explicitly the type of each NE. for each file, final NER result is stored under corpus1\_tagged/nltk/ .(needs to be created before running the program)

b) to use Stanford recognizer, it should first be configured with location of actual tagger file and encoding of data. method tag() under the recognizer takes raw tokens as input and token-type pairs as output. for each file, final NER result is stored under corpus1\_tagged/stanford/ .(needs to be created before running the program)

c) Stanford recognizer seems to be better in getting the bounders of named entities right as it can correctly keep word 'of' in name of NEs and detecting NEs with long names. (2-1, 2-2) NLTK recognizer have a high recall as it can detect a lot more results than Stanford one, like ‘Union’, ‘Republic’, ‘Constitution’, etc., while a bit lower precision, as it also gives some NEs that are not organizations, like 'RELATION OF', 'Unpleasant', etc.

For evaluation of full/partial matching, my standard is that matched NEs shall have the same location: in the same file and referenced by the same index. So, I add 'something' to both list of NEs detected by NLTK and Stanford recognizer, this may be an actual NE (or a part of it) or null element, depending on the label of token. In this way I can give a direct view of pair of NEs detected at the same location-although one in the pair may actually be null. For the given corpus there are 227 exact matches and 36 partial matches.



2-1 Stanford NER of recognizing ‘of’



2-2 Stanford NER of recognizing ‘of’

## 2. Sentiment analysis of movie reviews

a)

Patterns used to collect more adjectives:

1. adjectives connected by 'and', ',' and '&' are considered to be of the same polarity.

2. adjectives connected by 'but' are considered to be of the opposite polarity.

3. polarity of negated adjective (with 'no', 'not', "isn't", "wasn't", "weren't", 'never', 'seldom', 'barely', 'rarely', 'hardly', 'overly', 'excessively', 'too' in front of it) will be flipped.

4. when multiple occurrences of an adjective detected, its polarity will be positive if more of its occurrences are positive; its polarity will be negative if more of its occurrences are negative. (conflicts resolved)

A dictionary is used to record polarity of adjectives. The collection process goes for multiple rounds through the corpus. For each round, adjectives newly detected in previous rounds (for 1st round, the given ones) are used as seeds, so the process will stop when no new adjectives are detected, thus no adjectives can be used to update the polarity dictionary of adjectives. When an adjective is detected in the corpus if it has appeared in the dictionary, then its polarity increments/decrements according to its polarity according to rules above; else its polarity is initialized to 1/-1 according to the rules. Final polarity of adjectives can be determined by checking dictionary, those with polarity > 0 are considered to be positive and those with negative polarity are negative.

144 in 256 new positives and 20 in 34 new negatives are correctly classified, total accuracy is 56.55%.

A few errors occurred due to the failure of POS-tagging: some adverbs and nouns are mis-tagged as adjectives. Some negative adjectives are classified as positives and vice versa, perhaps originally they are used to describe the successful creation of horrible atmosphere or delivery of negative feelings, so the movie review those words are in may be positive towards the movie, but those words are negative on their own. Some examples in rt-polarity.pos can indicate this.

1. you have to pay attention to follow all the stories , but they're each interesting . the movie is well shot and very tragic , and one to ponder after the credits roll .
2. the last scenes of the film are anguished , bitter and truthful . mr . koshashvili is a director to watch .

Yet such a circumstance is not yet covered in those rules.

Neutral words also appear in both categories due to similar reason. Adjectives being assigned with opposite categories may also happen due to size limitation of corpus, as information is still not enough to do more accurate classification; or due to classification rule being not detailed enough.

b)

For the baseline, data cleaning on MPQA lexicon should be done first for convenience of data gathering. line 5549 and 5550: delete 'm' between 'stemmed1=n' and 'priorpolarity=negative'; line 3749: change 'polarity=negative priorpolarity=weakneg' to 'priorpolarity=negative'. A map between polarity in words to polarity in numbers is built first. In detail, 'positive' to 1, 'negative' to -1, 'both' and 'neutral' to 0. Polarity of each sentence depends on value after increments and decrements due to occurrence of tagged words in MPQR. Accuracy on positive examples is 60.4%, on negative examples is 42.4%, overall is 51.4%.

To build the machine learning classifier, I first collected all words appeared in corpus to give formation of feature vector. Then for each example, simple word count is performed to give the feature vector of it. Logistic regression is used for classification.

To expand the feature set, I considered polarity and strength of words from MPQR lexicon as well as if words are negated. Word count is still used for base of these expansions. Although we are told that using occurrence rather than frequency to build feature vector shall give a better performance, in practice they shall be almost the same as examples are too short to support a word occurs multiple times in it.

Taking into consideration of whether words come from MPQR lexicon, different weights are assigned to different words. Word not in lexicon are with lowest ones; those in lexicon and with 'type=weaksubj' have higher weight and words with 'type=strongsubj' weights highest.

A manually designed list includes words meaning that the word after it is negated. Any word in corpus with its previous word in this list will have its weight negated.

Tests are done on whether word origin and negation are considered and 5-fold cross validation is done on each of them. Basic machine learning classifier gives an obvious result than baseline with accuracy raised by around 16%. Further expanding feature, however, bring almost no improvements.

Machine learning approach gives a better approach than only using words from MPQA lexicon because BOW carries more info and thus better represents a sentence. An example sentence has around 21 words, but on average only 2.77 words in each example come from lexicon. Negation almost have no use as number of occlusions is too low compared to the size of corpus and consequently make little difference to feature vectors. The same reason applies to no usage of word origin. Only 3275 out of 20300 words in feature vector come from MPQA lexicon, and for each example this number can only be smaller. It means that change on feature vector is also limited.