CODE : JUPYTER NOTEBOOK

TO RUN : JUPYTER NOTEBOOK WITH KERNEL PYTHON3

Overall idea

Use basic NLP knowledge and techniques to clean the data, get corresponding vectors using word2vec and classify it into multiple classes

Cleaning data

The data was read from the tsv files and were then made to go through the following process:

1. Construct arrays of following fields:
    a. Main sentence
    b. Justification sentence
    c. Meta fields like topic, etc.
2. Remove stop words (using nltk)

3. Convert to lower case
4. Stem the words using english Snowball Stemmer (i.e. running becomes run)

Results were tried before and after doing these steps to see they do work well.

Sentence2Vec

Now individual sentence2vec arrays were generated using each tokenized and filtered fields of main sentence, justification sentence and meta fields(5 such fields) using word2vec. This was done using

1. Word2vec using gensim(Other word2vecs were tried like google news but it resultant accuracy got worse)
2. Sentence2Vec using fse

Now we have 7(sentence, justification sentence, 5 metas) sentence2vec vectors which are now concatted and we get a 700 dimensional vectore for each of train and test set

Classification

Scikit-learn has been used for classification task. The binary classification task has been handled using multi-class classification problem itself. The classes have been alloted integers as follows:

1. 'False'
2. 'Pants-fire'
3. 'Barely-true'
4. 'Half-true'
5. ‘Mostly-true'
6. 'True'

Now after being classified simply the predictions were divided by 3(consider them 0 indexed) and floor was taken to get binary class predictions.

The following classifiers were tried upon:

a. Nearest Neighbors
b. Linear SVM
c. RBF SVM
d. Gaussian Process
e. Decision Tree

f. Random Forest
g. Neural Net
h. AdaBoost
i. Naive Bayes
j. QDA

AdaBoost gave the best results followed by SVM with RBF kernel, but to save training time SVM was used as results were very similar.

LSTM was also tried upon and code is already given but due to lack of time results could not be obtained

Results

- - - -X

Test-set accuracy:-

Multiclass: 20.91%

Two-class: 56.35%

References

- - - - -X

Libraries used

Scikit-learn

Nltk

Gensim

Numpy

Keras

Resources

https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

https://aclweb.org/anthology/W18-5513.