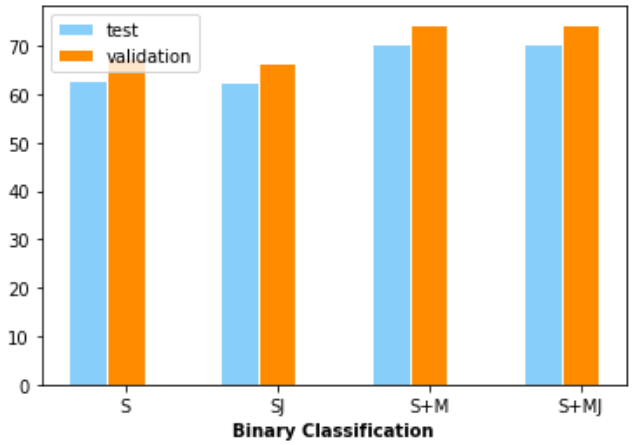
Fake News Detection

- *Shreyash Arya, IIIT-Delhi, India*

Note: All the experiments are done on LIAR version 2 dataset.

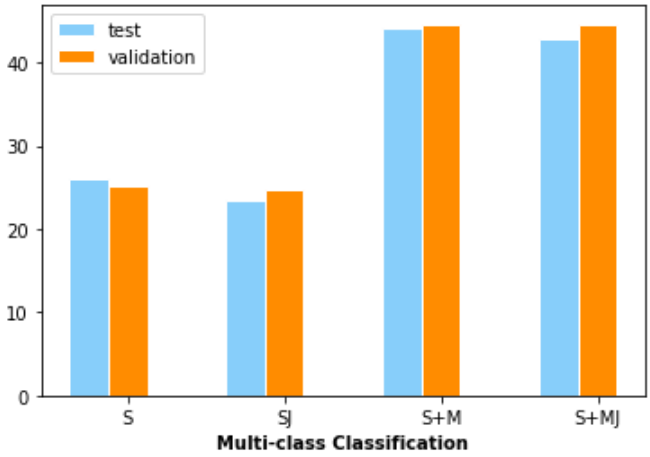
The highest accuracy you achieved

1. Binary classification task (true,false)



- S Condition:
  - Test set:
    - *Doc2Vec = 62.98%*
  - Validation set:
    - *Doc2Vec = 67.60%*
- SJ Condition:
  - Test set:
    - *Logistic Regression = 62.35%*
  - Validation set:
    - *Logistic Regression = 66.51%*
- S+M Condition:
  - Test set:
    - *MLP Classifier = 70.48%*
  - Validation set:
    - *MLP Classifier = 74.22%*
- S+MJ Condition:
  - Test set:
    - *MLP Classifier = 70.24%*
  - Validation set:
    - *MLP Classifier = 74.45%*

2. Six-way classification task (pants on fire, false, mostly false, half-true, mostly true, true)



- S Condition:
  - Test set:
    - *Doc2Vec = 25.73%*
  - Validation set:
    - *Ensemble = 25.23%*
- SJ Condition:
  - Test set:
    - *Logistic Regression = 23.44%*
  - Validation set:
    - *Ensemble= 24.76%*
- S+M Condition:
  - Test set:
    - *MLP Classifier = 44.04%*
  - Validation set:
    - *MLP Classifier = 44.62%*
- S+MJ Condition:
  - Test set:
    - *MLP Classifier = 42.93%*
  - Validation set:
    - *MLP Classifier = 44.54%*

Results table:

All the values mentioned are the % accuracies; ‘Test’ is the test set and ‘Val’ is the validation set.

Binary-class classification

	S	SJ	S+M	S+MJ
SVM	Test: 56.59 Val: 61.60	Test: 58.08 Val: 61.05	Test: 60.93 Val: 63.47	Test: 58.32 Val: 62.30
Logistic Regression	Test: 59.11 Val: 63.47	Test: **62.35** Val: **66.51**	Test: 62.35 Val: 67.52	Test: 62.35 Val: 67.75
MLP	Test: 55.80 Val: 57.94	Test: 55.32 Val: 60.12	Test: **70.48** Val: **74.22**	Test: **70.24** Val: **74.45**
Ensemble	Test: 59.03 Val: 64.56	Test: 60.77 Val: 65.96	Test: 65.03 Val: 69.78	Test: 67.82 Val: 68.53
Doc2Vec	Test: **62.98** Val: **67.60**	Test: 60.93 Val: 34.57	Test: 61.87 Val: 67.75	Test: 61.40 Val: 66.74

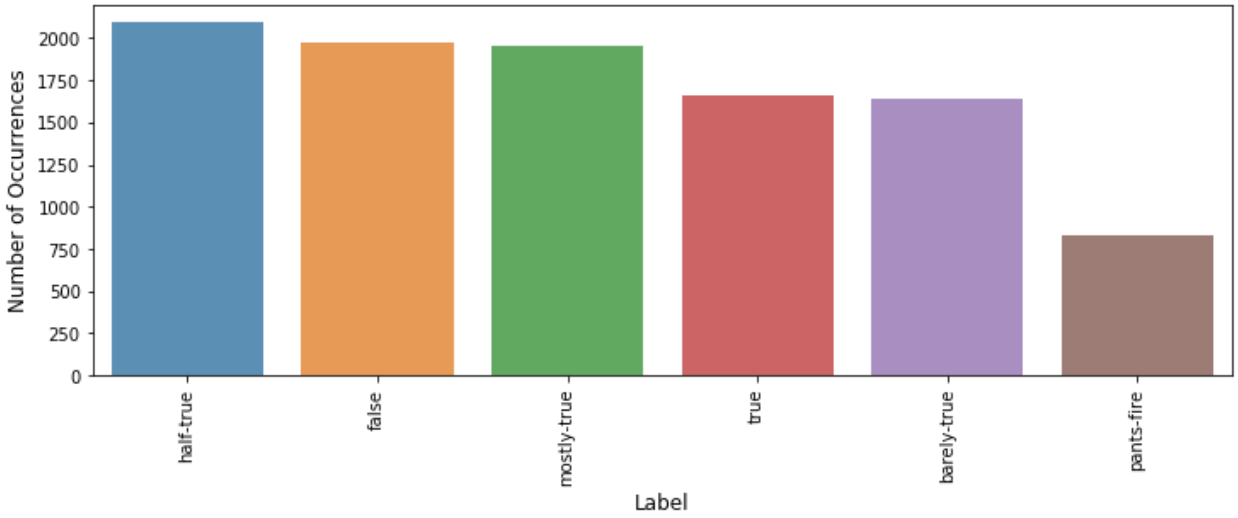
Multi-class classification

	S	SJ	S+M	S+MJ
SVM	Test: 23.04 Val: 23.98	Test: 21.78 Val: 21.33	Test: 24.23 Val: 26.55	Test: 25.01 Val: 29.12
Logistic Regression	Test: 23.36 Val: 23.83	Test: **23.44** Val: 24.22	Test: 29.75 Val: 32.08	Test: 30.30 Val: 33.25
MLP	Test: 21.94 Val: 23.05	Test: 23.31 Val: 21.41	Test: **44.04** Val: **44.62**	Test: **42.93** Val: **44.54**
Ensemble	Test: 23.44 Val: **25.23**	Test: 22.88 Val: **24.76**	Test:33.30 Val: 35.28	Test: 29.59 Val: 37.14
Doc2Vec	Test: **25.73** Val: 23.52	Test: 22.41 Val: 22.42	Test: 26.44 Val: 27.33	Test: 25.49 Val: 28.27

Explanation of how you achieved the result

Data Preparation: I preprocessed the statements and the justifications using my own created preprocessing pipeline including tweet-preprocessor (which removes all the URLs, hashtags, mentions, emojis), different regex for removal of punctuations and whitespaces, removal of stopwords, lower-casing and lemmatization.

Class Distribution:



The data is fairly balanced except the ‘pants-fire’ class which we’ll see later in the F1 scores and which affects the accuracy.

Model Preparation: I used the data (version 2) for preparing the different model conditions (S, SJ, S+M, S+MJ) following the steps mentioned in the paper.

- S model data is prepared using the unigrams of the statements.
- SJ model data is prepared using the unigrams of the statements and the justifications.
- S+M model data includes the unigrams of the statements, sentiments of the statements (positive and negative) calculated through the ‘SentiStrength’, emotion of the statements using the ‘Emolex’ lexicon and the metadata including the history score (the number of claims each speaker makes for every truth-label).
- S+MJ includes all the attributes of the S+M data model plus the unigrams of the justifications.

My basic strategy was to first replicate the results mentioned in the paper and then try to improve upon it using different methods. Hence, I started with the machine learning models Linear SVM and Logistic regression and calculated the accuracies on different models. The Deep Learning and Machine Learning had the same results as mentioned in the paper so I tried to save time and use some other models apart from the mentioned ones in the paper such as Multi-Layer Perceptron (MLP), Ensemble Learning and the Doc2Vec DBOW method.

I was able to achieve the same (and better in most cases) for the S and the S+M data models using the Logistic regression but the SJ and S+MJ had lesser accuracies than the mentioned in the paper if I used the Linear SVM and the Logistic Regression with the default parameters. The 'justification' feature was not able to improve as much as mentioned in the paper. Though, using the Multi-Layer Perceptron with a single hidden layer of 30 nodes, better accuracies were achieved in all the cases (except the SJ which can be because of the small dataset fed to the neural network and it cannot converge well).

The ensemble of SVM, Naive-Bayes, Random Forest and Logistic Regression also worked well as it has the second-highest accuracies in most of the cases. Hard voting was is used which chooses the majority voting score.

Distributed Bag of Words (DBOW) is used in Doc2Vec which is similar to Skip-gram model in word2vec which tries to predict the probability of distribution of source context words given a randomly sampled target word. It uses neural networks for the probability calculations and Logistic Regression is used for label prediction. Similar to MLP, it gave good accuracies on both S and SJ datasets. S+M and S+MJ datasets contain metadata which is numerical and hence to incorporate it in Doc2Vec setting, it can be converted to text. For example, for a positive sentiment with score = 2, we can add '_pos_sentiment_' word to the text two times. Here, we can see that it doesn't perform well as these metadata features will be less in number as compared to other words in the combined 'statement' and 'justification' unigrams.

The combination of train and validation set after tuning the hyper-parameters using Grid-Search didn't make much difference to the ML models and were also time-consuming. Hence, I used the default parameters which gave nearly the same accuracies for the basic ML models and better accuracies for the advanced methods.

Different ideas you tried out

Machine Learning Models - Support Vector Machines (SVM; Linear kernel), Logistic Regression, Random Forests, Gaussian Naive Bayes, Multinomial Naive Bayes.
Neural Networks - Multi-Layer Perceptron (MLP)
Ensemble Learning - Combining the SVM, Logistic Regression, Random Forest and Gaussian Naive Bayes.
Doc2Vec similarity.

You must cite all libraries and code from online resources that you use

https://radimrehurek.com/gensim/
https://numpy.org/
https://spacy.io/

https://github.com/bmabey/pyLDAvis
https://pypi.org/project/tweet-preprocessor/
https://www.nltk.org/
https://pandas.pydata.org/
https://github.com/tqdm/tqdm
https://scikit-learn.org/
https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4
https://www.google.com/search?q=tfidfvectorizer&oq=tfidf&aqs=chrome.1.69i57j0j69i60l2j0l2.3537j0j7&sourceid=chrome&ie=UTF-8
https://pypi.org/project/sentistrength/
https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
https://github.com/Tariq60/LIAR-PLUS
https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.hstack.html
https://www.youtube.com/watch?v=ZkAFJwi-G98
http://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/
https://python-graph-gallery.com/11-grouped-barplot/