

Appendix

A1. Range of Hyperparameter Values for Random Forest

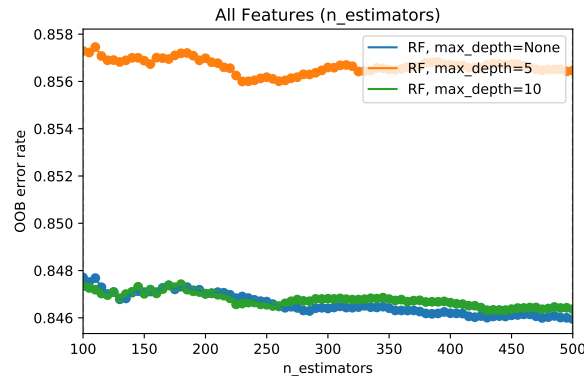


Figure 1: Range of `n_estimators` across various `max_depth` levels in a model that includes all features except word embeddings. On one train split, we illustrate the range of `n_estimators` considered between 100 and 500 across different `max_depth` parameters. We observe the OOB error rate levels off.

A2. Sensitivity of Doc2Vec to Parameter Ranges

In this section, we verify that the predictive performance of the `doc2vec` features do not vary in a meaningful way depending on the parameters used such as `window` or `min_count`. In Figure 2, we observe the MSE results are comparable across parameter settings. We also observe that unigram features are comparable to `doc2vec`. We use the vector embedding features in the main paper due to the reduced feature space which helps with the model run-time for Random Forest (100/200 features for embeddings versus over 20k features for TF-IDF features).

Next, we qualitatively check if the word embeddings identify similar words that make sense:

```
>>> model.most_similar('sick')
[(u'tired', 0.6481747031211853), (u'upset', 0.6177597641944885),
 (u'trashed', 0.5969120264053345), (u'bored', 0.578025221824646),
 (u'hungry', 0.5762773156166077), (u'pregnant', 0.5597766041755676),
 (u'robbed', 0.5588027238845825), (u'headache', 0.5584173798561096),
 (u'drunk', 0.5545065999031067),
 (u'hungover', 0.5460582971572876)]
```

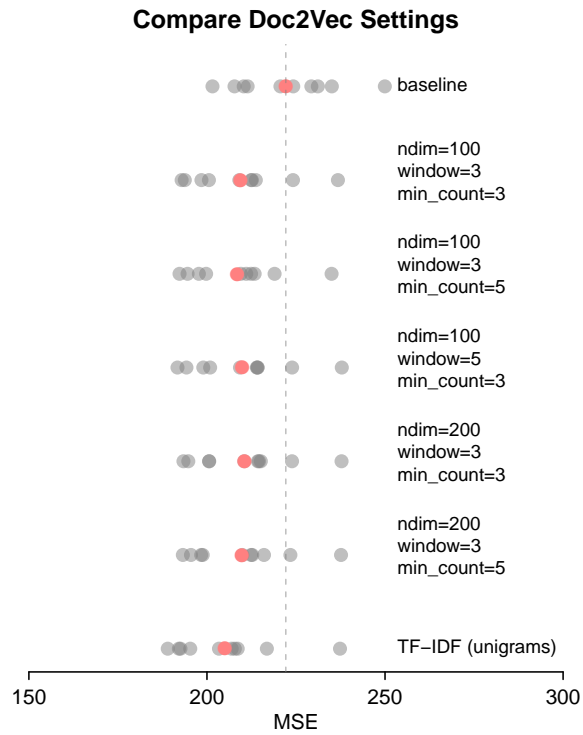


Figure 2: We compare the various parameter setting when fitting the doc2vec model and note results are not substantively affected by different settings like dimension. we also evaluate unigram features (via TF-IDF) and see comparable performance to the word embedding features.