Presentation
Advanced Information Retrieval [706.705] - Jan 2023

A true news recommender an TF-IDF variation and Topic Modelling

Group 16

Responsibilities: Research, Idea, implementation, Running the project (GPU/CPU)

- Csongor Hegedüs
- Mario Lukas Mauberger
- Roman Lukas Prunč
- Suraj Yathinatti

Repository link: https://github.com/mariomauberger/fakeNewsClassification/

Aim

Based on 1 news article

Suggest <u>5 additional</u> articles

Covering the same or the most similar <u>topic</u>

Which are <u>true</u> and not fake news

Table of Contents

- Dataset
- Exploratory Data Analysis
- Data Cleaning
- Baseline Model
- Class relevant TF-IDF
- Topic Modelling with BERTopic
- BERT-Plus Model
- Evaluation
- Conclusion

Dataset

- Dataset:
  WELFake - Dataset
  https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification/versions/35?resource=download

- **Title**, **Text**, and **Label** are the three main columns.

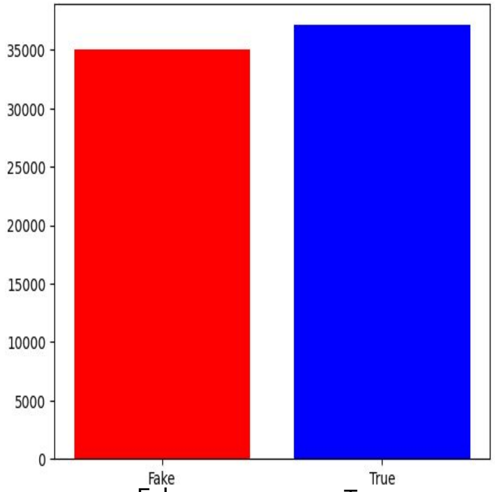
- **72134** distinct values

- **Label** can be either 0 or 1- indicating if the news is fake or true.

Exploratory Data Analysis

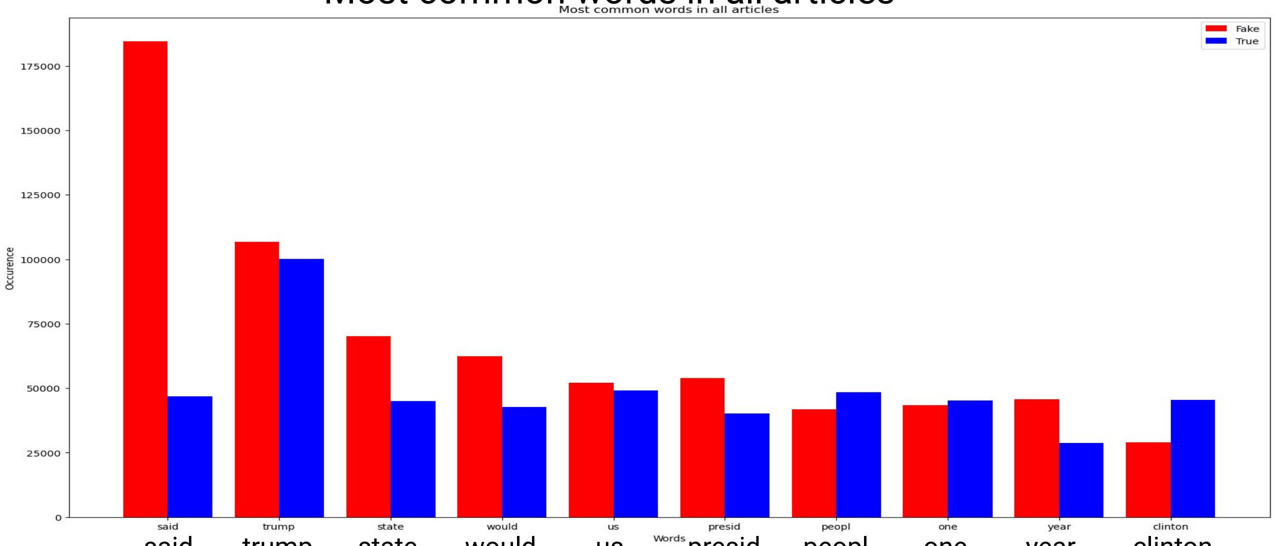
- Balance between false and real news - slightly more tn.
- Word count after preprocessing steps (Punctuation and stop word removal, stemming)

```
Unnamed: 0    72134
title         62348
text          62719
label             2
dtype: int64
```



Fake True

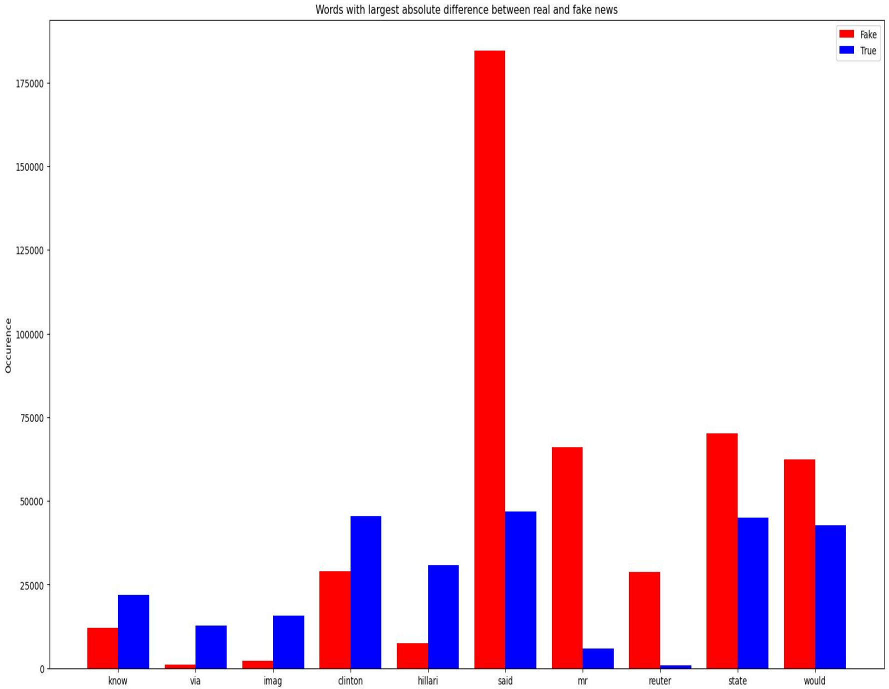
Most common words in all articles



said trump state would us presid peopl one year clinton

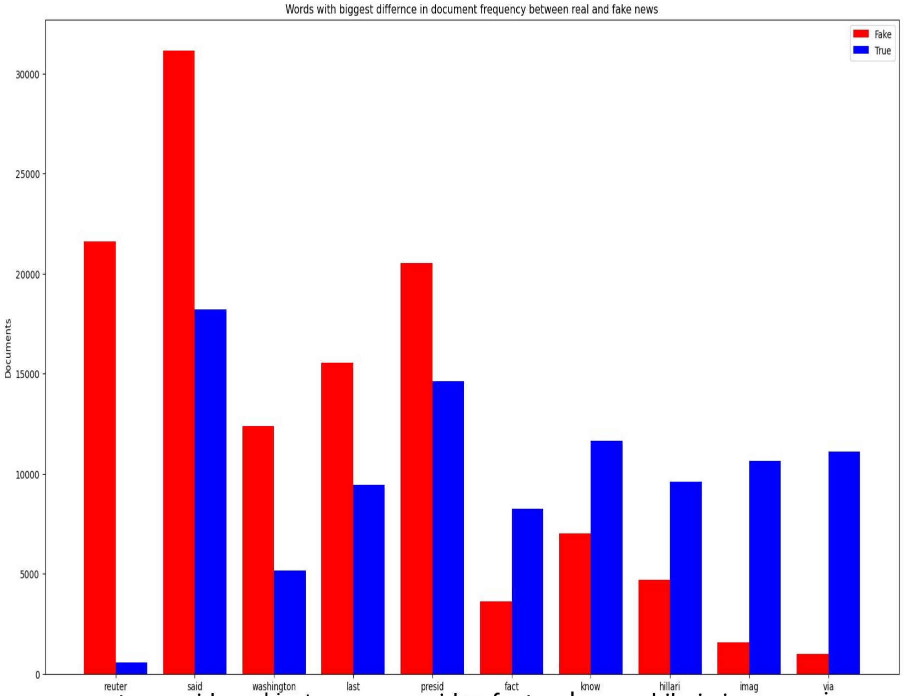
Words with the largest absolute difference between true and false news

Words with largest absolute frequency between tn and fn



know via imag clinton hilari said mr reuter state would

Words with biggest df difference between tn and fn



reuter said washington last presid fact know hilari imag via

Splitting the data

- **Training-test data split: 80%-20%**
- **Training data: 57707 documents**
- **Test Data: 14427 documents**

Baseline Model

- Simple **TF-IDF**

- **Naive Bayes Classifier**

- Fake news **detection accuracy - 87%**

- Used for article r**ecommendation - MAP@k = 51%**

CLASS RELEVANT TF–IDF

- Identifying **overrepresented words** in one news category

- Based on training data

- **IDF** for each term in a document - but only **considering opposite class**

    E.g. 1 tn document regarded against all fn documents

- For terms in **more than 10 documents** in respective category

- Considered if **difference** between tn IDF and fn IDF **0.5 or larger**

- Then **weighted** with simple **TF**

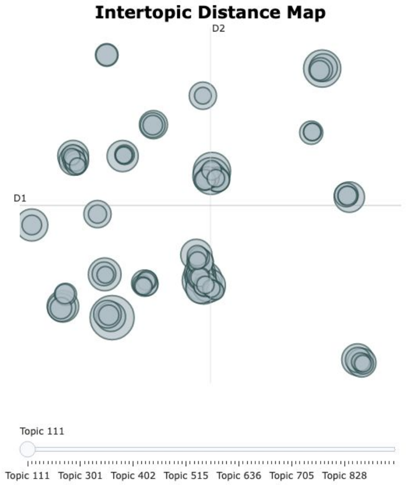
Topic Modelling with BERTopic

- Identified **911 topics - 18343 documents** in “garbage topic”

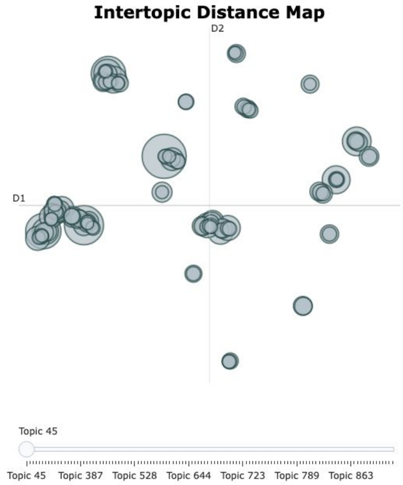
- **91 topics only in fn** (2173 training documents)

- **116 topics only in tn** (2510 training documents)

Fake news



True news



Note: Due to time and computational limitations, we had to eliminate the initial implementation that used sentiment scores to distinguish between fake news and true news within the topics.

BERT-Plus Model

- **Learning language**
  - Contextualized meaning

- **Predicting** if articles are TRUE/FAKE

```
class BertPlusModel(torch.nn.Module):
    def __init__(self, bert_model):
        super(BertPlusModel, self).__init__()
        self.bert_model = bert_model
        self.dropout = torch.nn.Dropout(p=0.2)
        self.linear = torch.nn.Linear(769, 1)
```

Model Layers

- **BERT** (Bidirectional Encoder Representation of Transformers) (bert-base-uncased)
  - MLM (Masked language modeling)
  - NSP (Next sentence prediction)

- **Linear Layer** incorporates the **Topics**

BERT-Plus model performance

	Training loss	**Test accuracy**
Without garbage-category: (57k training data) (14k test data)	Epoch 1: 0.0064 Epoch 2: 0.0014 Epoch 3: 0.0008 Epoch 4: 0.0005 Epoch 5: 0.0003	96.36%
Including garbage-category: (39k training data) (8k test data)	Epoch 1: 0.0091 Epoch 2: 0.0020 Epoch 3: 0.0010 Epoch 4: 0.0005 Epoch 5: 0.0004	97.08%

Recommendations

- **Query**
  - Random article
- **Output**
  - Top 5 articles
    - same (or most similar) topic(s)
    - TRUE news.

- Within one topic
  - Evaluate whether articles **true or false**
  - Rank tn based on **tf-idf**
  - Return in that order - until 5 recommended
- If l**ess than 5 articles** found - add new ones from **next topic**

Evaluation/Results

	Fake news detection	Mean average precision @k
Baseline model	87.00%	51.00%
BERT-Plus model	97.08%	92.70%

MAP@k tested for the BERT-Plus model on 1000 articles from the test data.
Larger datasets crashed our kernels.

Conclusion

- **Improved fake news detection**
    - ~10%

- **Improved recommendations (map@k)**
    - ~41%

- **Good dataset for training our model**
    - 95 % accuracy with 20k train data
    - 97 % accuracy with 57k train data
    - Garbage category did not have a big influence on the accuracy

- **Good results but computationally expensive**

And our struggles 😅