Neural Network, Text classification, Knowledge Representation

Hamed Ahmed
Sugarbayar Enkhbayar
University of Warsaw, Faculty of Economics
h.hamedahmed@student.uw.edu.pl
University of Warsaw, Faculty of Economics
s.enkhbayar@student.uw.edu.pl

Project Goal:

- Use Classification and Neural Networks
- Try different methods like TF-IDF, words2vec, FastText
- Compare these methods using evaluation metrics
- Build a knowledge representation graph to understand research papers

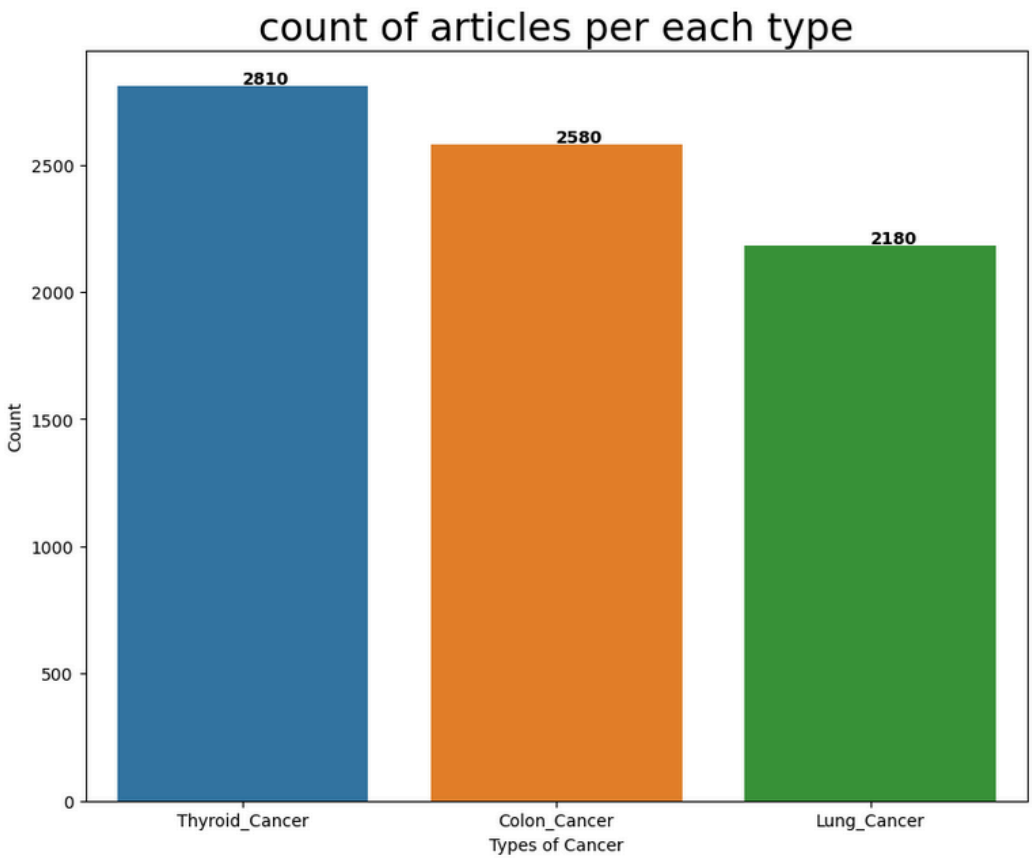
Assumptions:

- Belief that Neural Networks will perform better
- Expectation of better results using FastText for classification
- Have a basic understanding from the knowledge graph

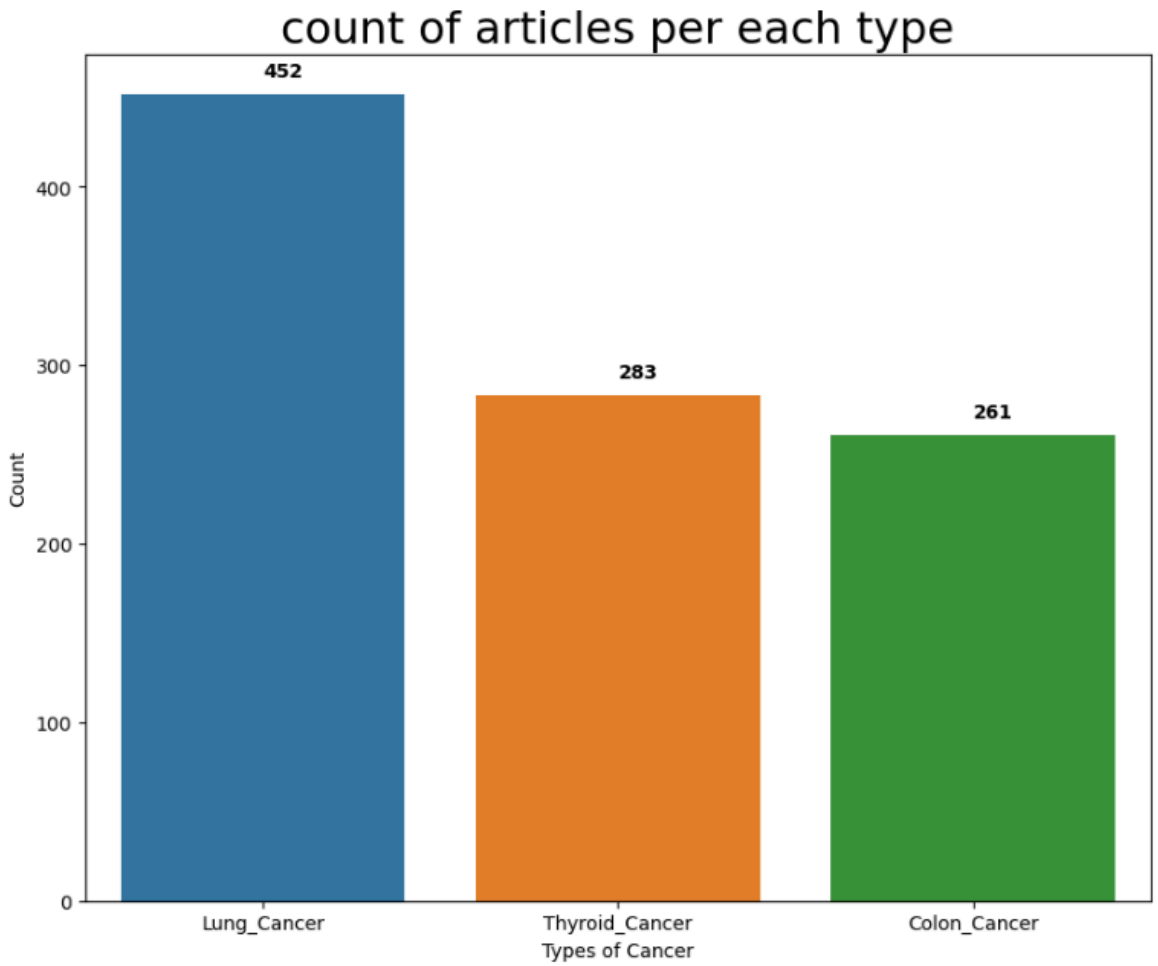
Data:

Source	Kaggle
Link	Medical dataset
Main tools	Neural Network Text classification Knowledge Representation
Number of observation	7570
Variables	[label, research paper text]
Number of observation (unique)	996

- **Dataset before cleaning the redundant values**



- **Dataset before cleaning the redundant values**



- **Data Cleaning**
  - Data were cleaned by doing the following procedures
    - Text to lower
    - Removing special characters
    - Convert words to tokens
    - Applying lemmatizer
    - Removing stop words
    - And applying words cloud hence i removed words that appear so many times like [patient, cancer,.. Etc]
    - Removed redundant data

- **Word Cloud**
  - I assume that some of the text research were checking patients in May, and probably they asked about if they are eating enough **protein** also different **methods** were applied in the research and they showed the **effect** of the medicine they patient took.

Word cloud for research paper

- **Model and evaluation:**
  - Class 0 corresponds to **Lung Cancer**.
  - Class 1 corresponds to **Thyroid Cancer**.
  - Class 2 corresponds to **Colon Cancer**.

- **XGBoost with TF-IDF**

---

	precision	recall	f1-score	support
0	0.80	0.85	0.82	55
1	0.95	1.00	0.97	89
2	0.85	0.71	0.78	56
accuracy			0.88	200
macro avg	0.86	0.86	0.86	200
weighted avg	0.88	0.88	0.88	200

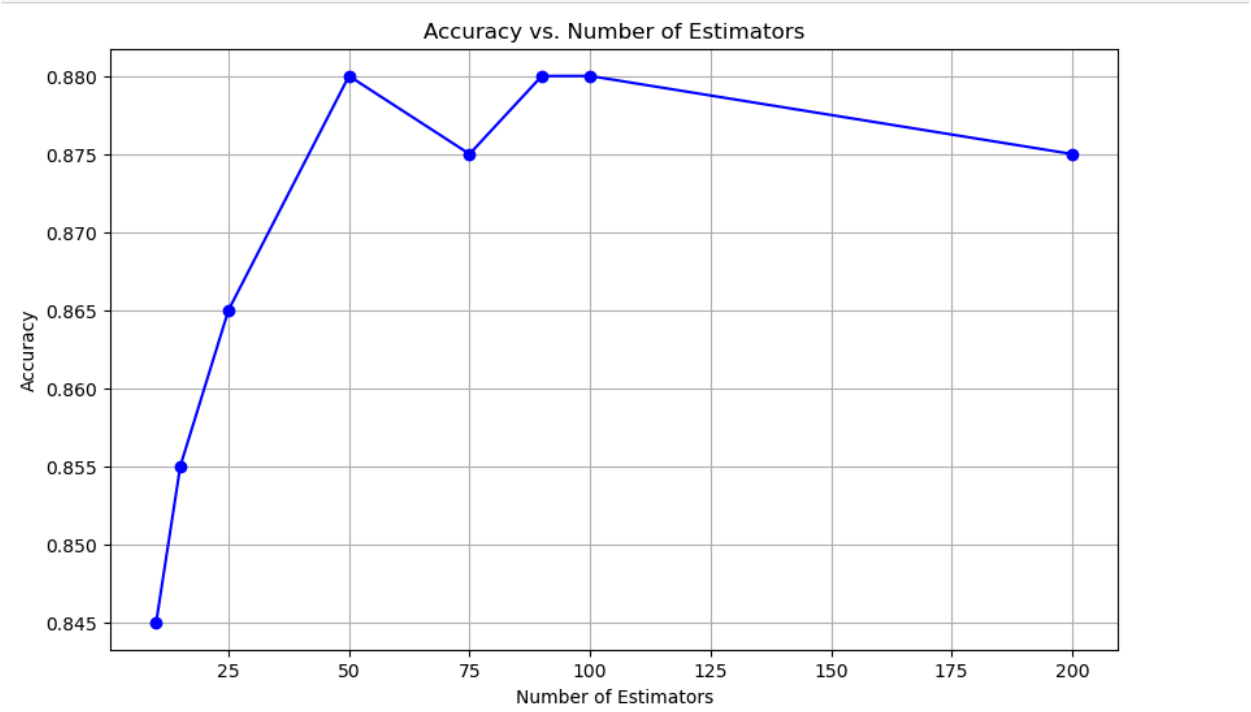
Accuracy Score for TF-IDF Features: 0.88

- **Precision** measures the accuracy of positive predictions. For example, class 1 has the highest precision (0.95), meaning it's very accurate when it predicts class 1.
- **Recall** indicates the ability to find all relevant instances. Class 1 has perfect recall (1.00), meaning it successfully identifies all true class 1 instances.
- **F1-Score** is a balance between precision and recall. A high F1-score, like 0.97 for class 1, suggests both good precision and recall.
- **Support** is the number of true occurrences of each class. Here, class 1 is the most frequent in the dataset with 89 instances.
- **Accuracy** (0.88) shows the proportion of correctly predicted instances out of total predictions, indicating good overall performance.
- **Macro Average** computes the average metrics without considering support, showing balanced performance across classes.

- **Weighted Average** takes support into account, giving more weight to larger classes. This also indicates good performance across classes considering their frequency.

The model performs best on Thyroid Cancer and has good overall accuracy and balanced performance across classes, as indicated by the macro and weighted averages.

- **Applying more different number of estimators to check model performance**

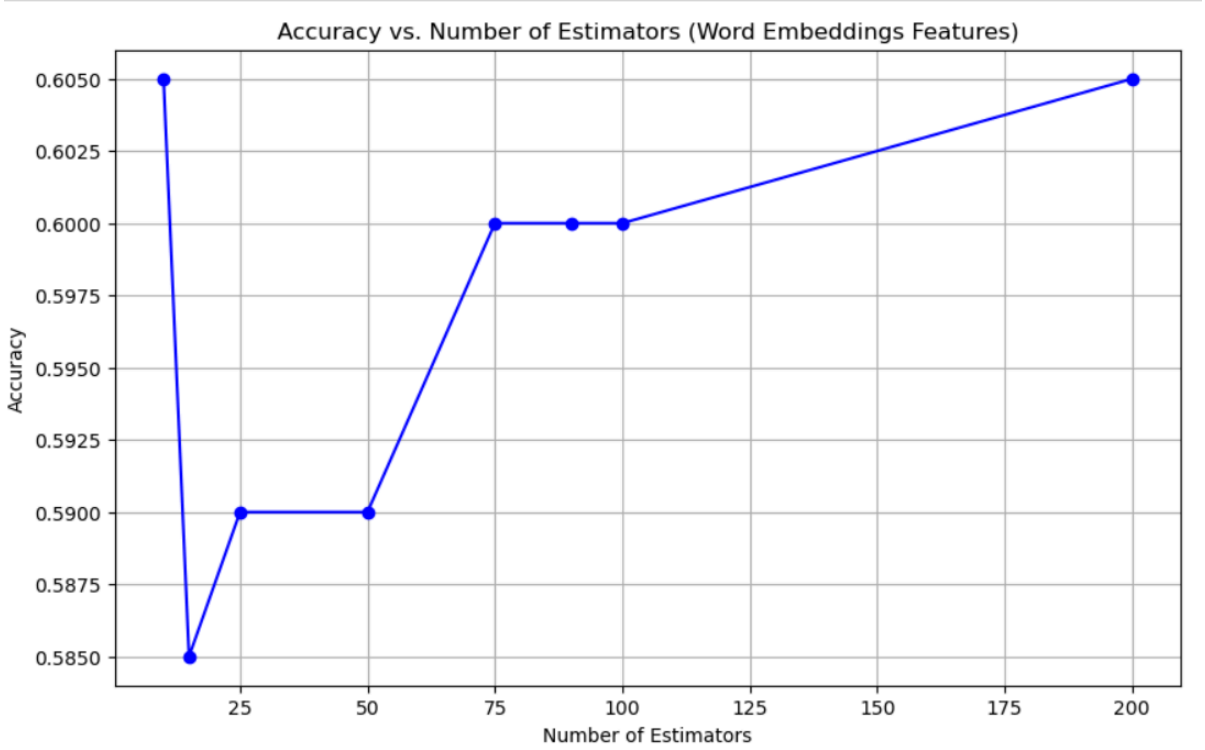


The graph indicates that model accuracy peaks with around 50 to 75 estimators and then stabilizes, suggesting that additional estimators beyond this range do not substantially improve performance.

- **XGBoost with words2vec**

	precision	recall	f1-score	support
0	0.41	0.36	0.38	55
1	0.81	0.91	0.86	89
2	0.39	0.36	0.37	56
accuracy			0.60	200
macro avg	0.54	0.54	0.54	200
weighted avg	0.58	0.60	0.59	200

The model has moderate overall accuracy at 60%, performing best in identifying class 1 with high precision and recall, while it struggles with classes 0 and 2, as indicated by their lower respective scores. The macro and weighted averages suggest there is room for improvement, especially in balancing performance across all classes.

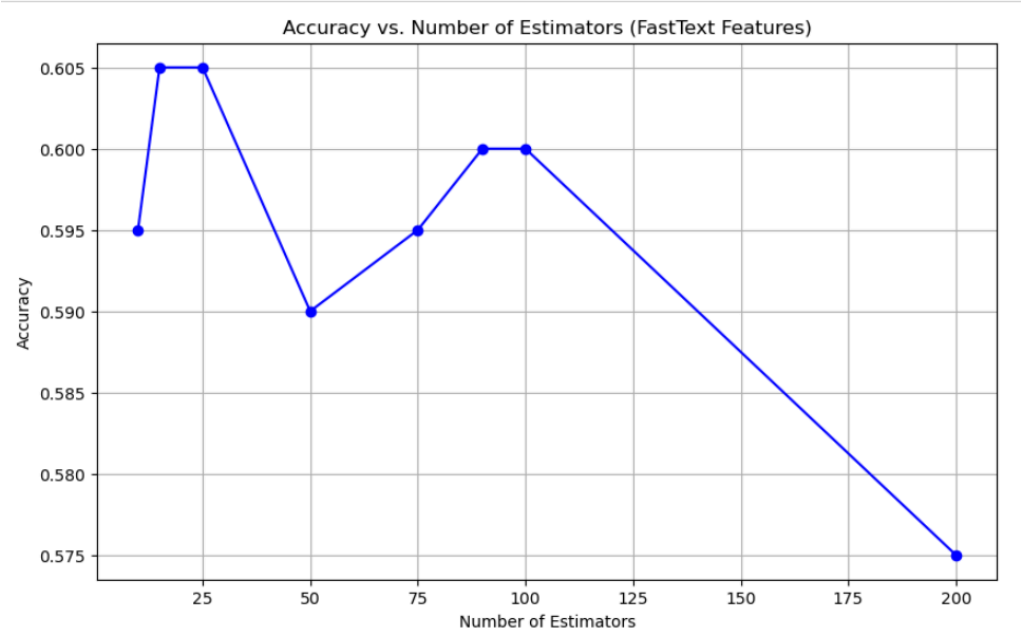


The graph initially fluctuates but generally improves as the number of estimators increases, with the highest accuracy observed at 200 estimators.

- **XGBoost with FastText**

	precision	recall	f1-score	support
0	0.34	0.25	0.29	55
1	0.78	0.89	0.83	89
2	0.38	0.39	0.39	56
accuracy			0.57	200
macro avg	0.50	0.51	0.50	200
weighted avg	0.55	0.57	0.56	200

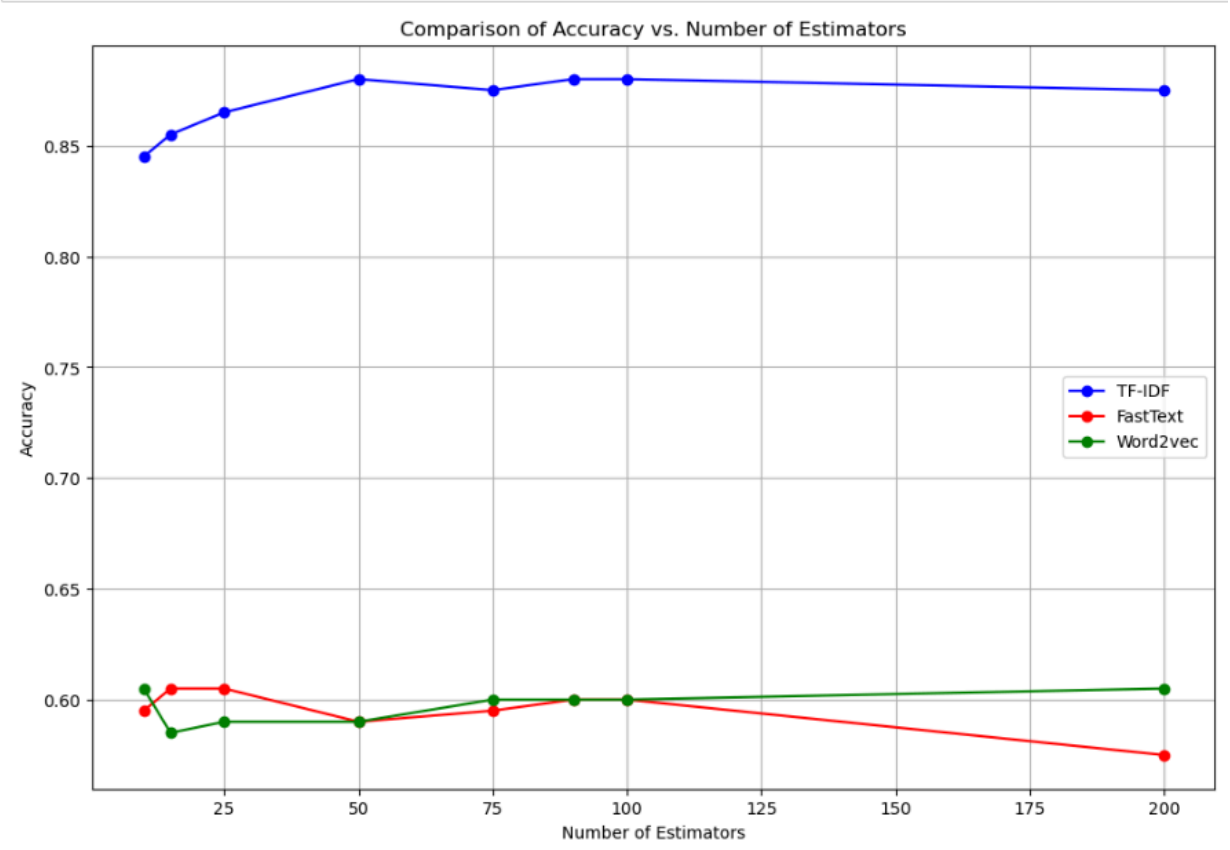
The model shows relatively low precision and recall for classes 0 and 2, a better performance for class 1, and an overall accuracy of 57%, indicating a need for improvement, especially for classes 0 and 2**.**



The graph suggests that the accuracy of a model with FastText features peaks at 25 estimators, decreases, then plateaus, and finally decreases significantly as the number of estimators reaches 200.

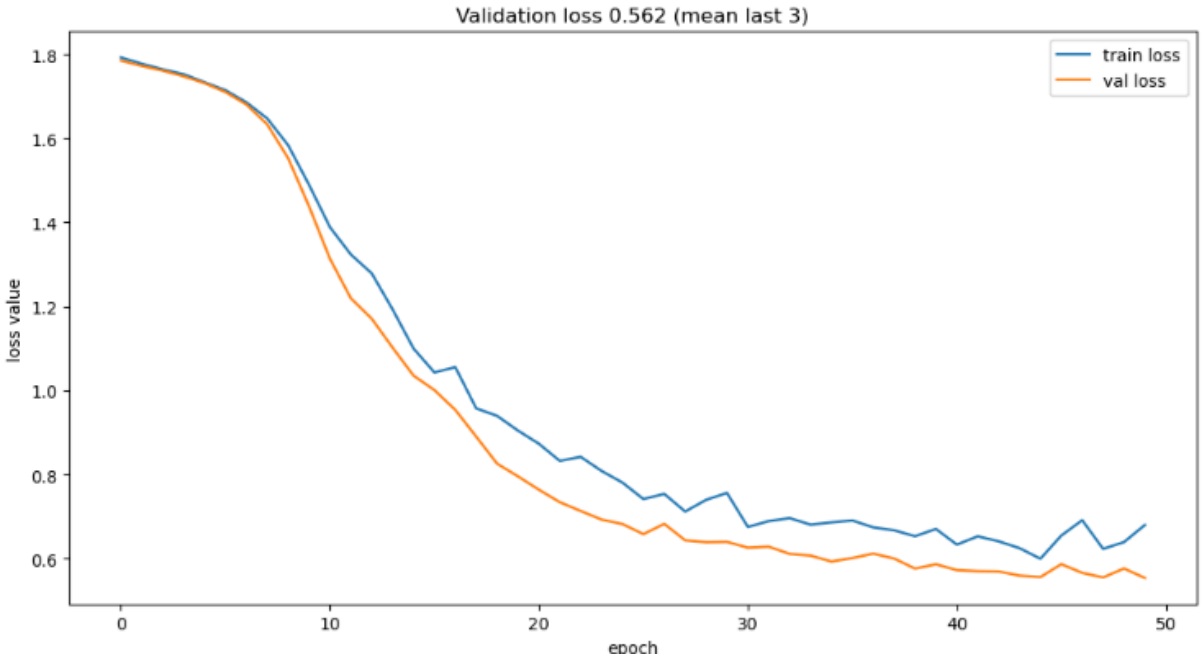
- **Comparison between the 3 methods**

The comparison graph indicates that models using TF-IDF features consistently outperform those using FastText or Word2vec features across all numbers of estimators, with TF-IDF maintaining high accuracy irrespective of the number of estimators used.



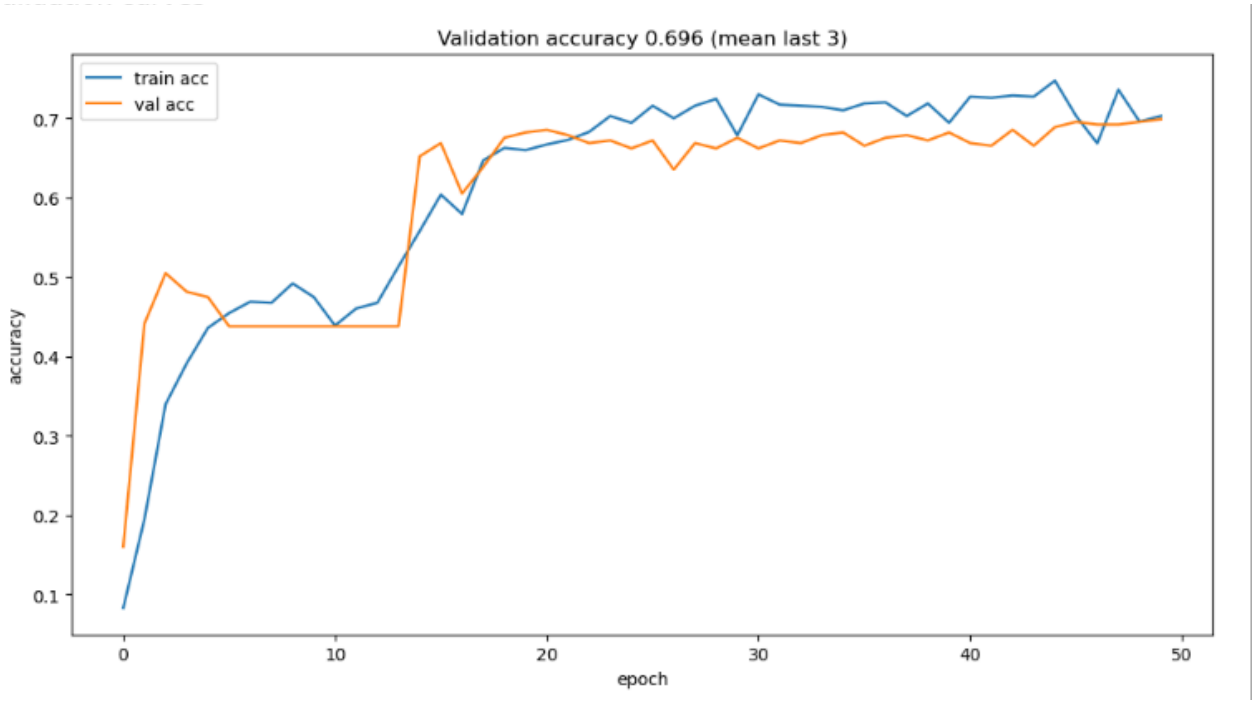
Neural Network

loss: 0.6800 - accuracy: 0.7030 - val_loss: 0.5537 - val_accuracy: 0.6990 - lr: 1.0000e-04



The graph shows the training and validation loss of a neural network over epochs.

- Both losses decrease over time, indicating learning and model improvement.
- However, as the epochs increase, the validation loss shows some volatility and does not decrease as smoothly as the training loss, which may suggest the beginning of overfitting or that the model is not generalizing as well to new data as it is to the training data.
- The mean validation loss of the last three epochs is noted as 0.562, which is likely a measure used to assess the model's performance stability toward the end of training.



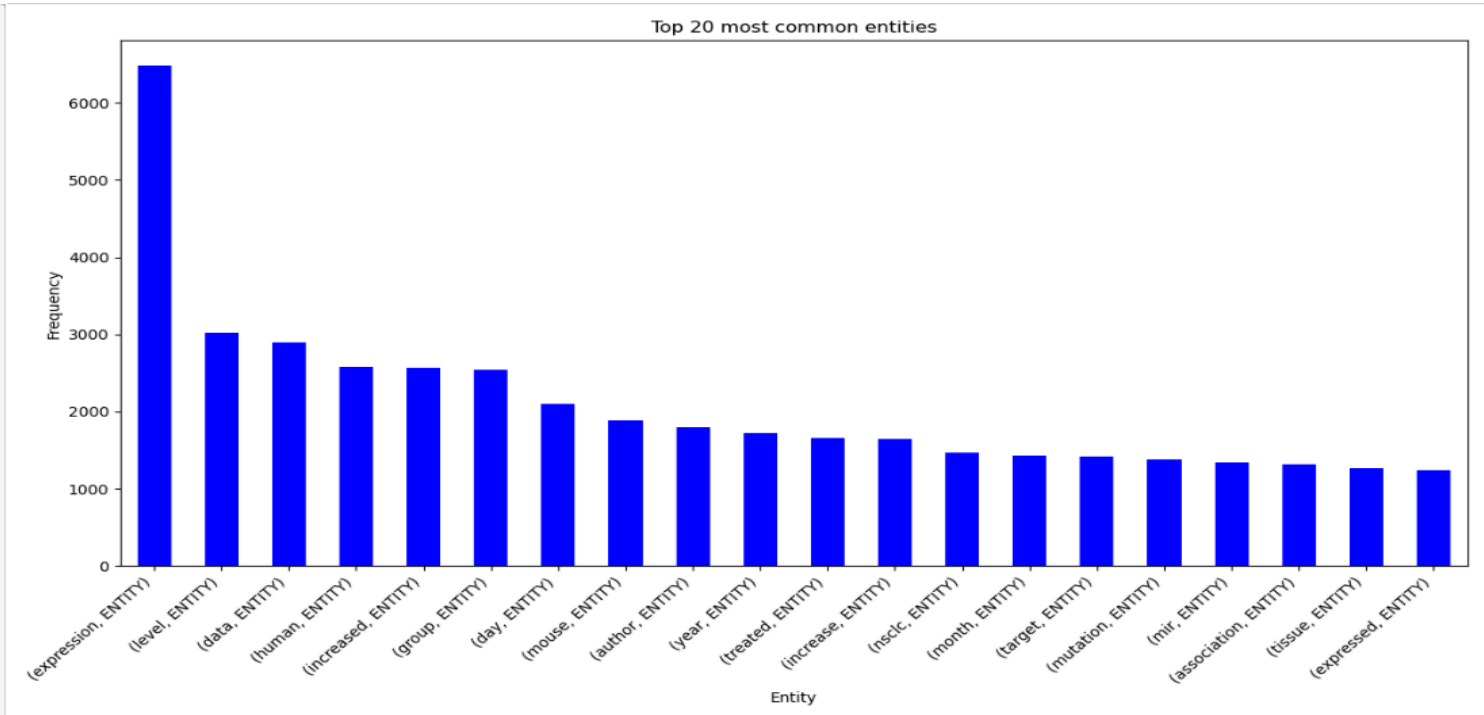
The graph depicts training and validation accuracy of a neural network over 50 epochs.

- Both accuracies improve with more epochs, but validation accuracy plateaus and shows some fluctuation, indicating the model may be starting to overfit.
- The mean validation accuracy of the last three epochs is 0.696, which is a common practice to evaluate the model's performance on unseen data.

Knowledge Graph

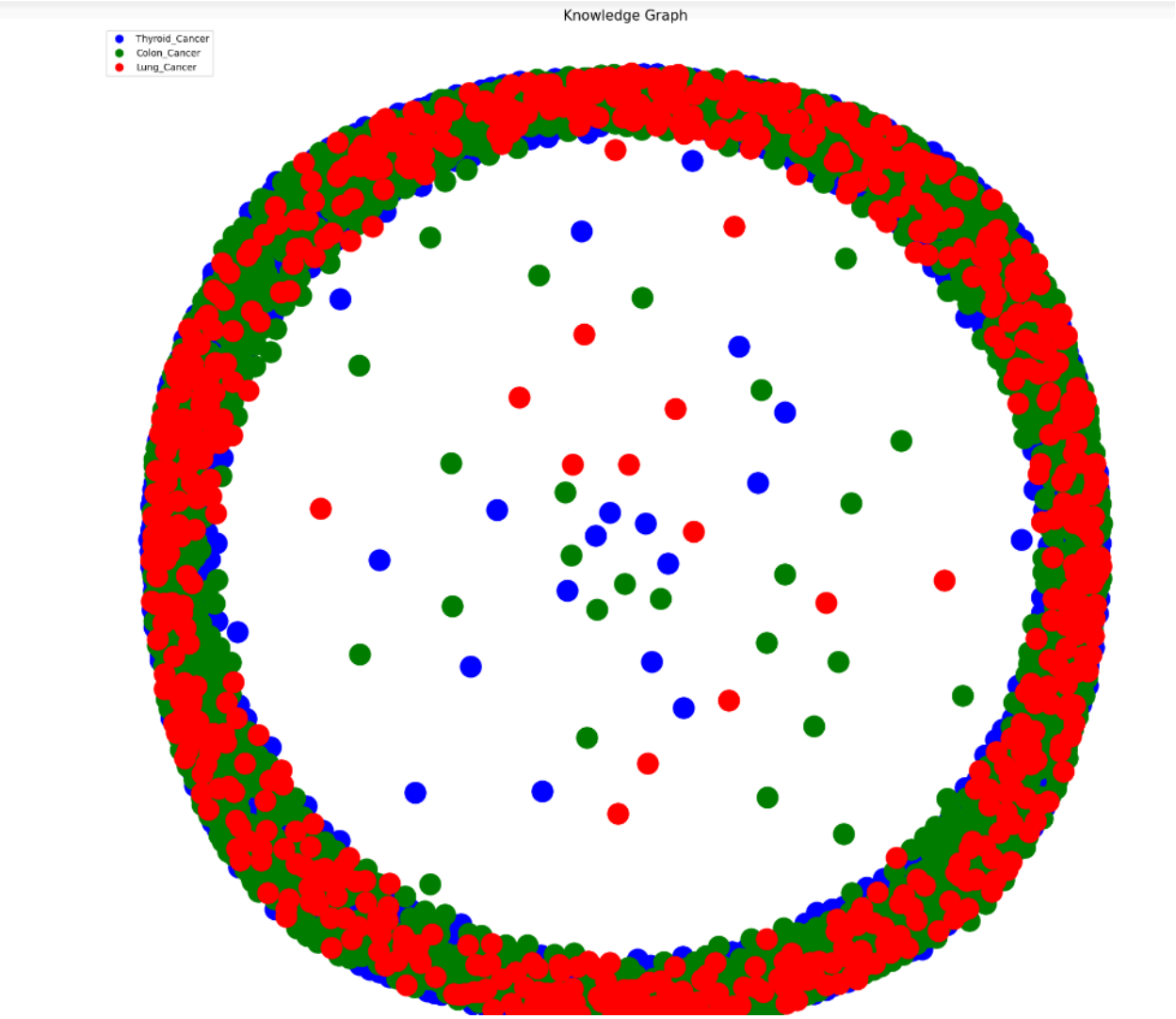
- Converting cleaned_text into entities using Spacy and extracting the relationship between them.

	label	cleaned_text	entities	relationships
0	Thyroid_Cancer	thyroid surgery child single institution osama...	[(thyroid surgery, ENTITY), (institution, ENTI...	[(aseerib, ahmed, alhumaida), (osama, ali, sau...
1	Thyroid_Cancer	adopted strategy wa used prior year four exclu...	[(year, ENTITY), (return four disjoint citatio...	[(term, made, set), (query, qwosfiltered, sear...
2	Thyroid_Cancer	coronary arterybypass grafting thrombosis brin...	[(coronary arterybypass grafting, ENTITY), (th...	[(report, admitted, bypass), (report, admitted...
3	Thyroid_Cancer	solitary plasmacytoma sp skull uncommon clinic...	[(solitary plasmacytoma, ENTITY), (clinical en...	[(report, describes, mass), (report, describes...
4	Thyroid_Cancer	aimed investigate serum matrix metalloproteina...	[(investigate, ENTITY), (serum matrix metallop...	[(metalloproteinase, mmplevels, compared), (mm...



The bar chart displays the top 20 most common entities in a dataset, with the term "expression" being the most frequent,
However we applied lemmatization but still it's not that effective as I can see [expressions] and [expressed].

Showing Knowledge graph :



The graph

- visualizes a network of research entities related to different types of cancer.
- Lung Cancer entities exhibiting stronger interconnectivity compared to Thyroid and Colon Cancer entities.
- The presence of isolated entities indicates areas with less research or weaker relationships.
- The complexity of medical terminology indeed requires specialized libraries or tools to accurately identify and map the

relationships between different entities within such a knowledge graph.

Summary:

- Using XGBoost with TF-IDF showed better performance rather than other methods even neural networks didn’t perform well most probably because the data weren’t enough to train the neural networks.
- For Knowledge graph it shows that for some researches the algorithms were able to identify entities and it’s relationship but as terms for medical words are hard to identity it requires special libraries for that
- Also in the code you can find that I applied models with removing redundant data first but the result wasn't real at all.

Appendix:

1- Used dataset inside unique-ds

2-also you can find a copy of the code inside this file: [MedicalTextClassification.ipynb]