Summary

This report outlines the methodologies and findings from three exercises conducted in a data retrieval course, focusing on the processing and analysis of a dataset comprising 15,000 textual documents distributed equally among three categories: A, B, and C.
The first exercise involved extensive preprocessing of these texts, including cleaning and lemmatization, to prepare them for feature extraction. Techniques such as TF-IDF with BM25/Okapi, word2vec, doc2vec, and BERT or Universal Sentence Encoder were employed to generate sparse matrices and document vectors, providing a foundation for subsequent analyses.
The second exercise delved into clustering and classification. Various clustering methods like k-means, DBSCAN, and Mixture of Gaussian were applied to paired document groups to explore thematic or stylistic similarities. Additionally, classification models, including an Artificial Neural Network (ANN), along with Naive Bayes, Support Vector Machine, and Logistic Regression, were developed. These models were used through 10-fold cross-validation to ensure reliability, with an emphasis on understanding the features that significantly contribute to group distinctions.
The third exercise focused on sentiment analysis using 3 different methods, such as NLTK sentiment analysis after translation with Google Translate API, Hebert a hebrew model based on Bert and lastly CNN, a convolutional neural network trained on hebrew texts. Thrice methods gave different insights on the models themselves and on the texts.

Overall, the exercises underscored the importance of meticulous data preparation, the selection of appropriate feature extraction methods, and the application of advanced machine learning techniques in uncovering the latent structures within text data. The findings from this study not only contribute to the academic understanding of text analysis but also offer practical implications for data retrieval applications in various domains.

INTRODUCTION

In the rapidly evolving field of data retrieval, the ability to effectively process and analyze large volumes of text data is paramount. This report presents a comprehensive examination of methodologies employed in handling a substantial dataset of 15,000 documents, evenly distributed across three distinct categories: A, B, and C. The primary objective of these exercises was to explore and implement a series of text processing, feature extraction, and analytical techniques to not only clean and prepare the data but also to uncover underlying patterns and relationships within the texts.

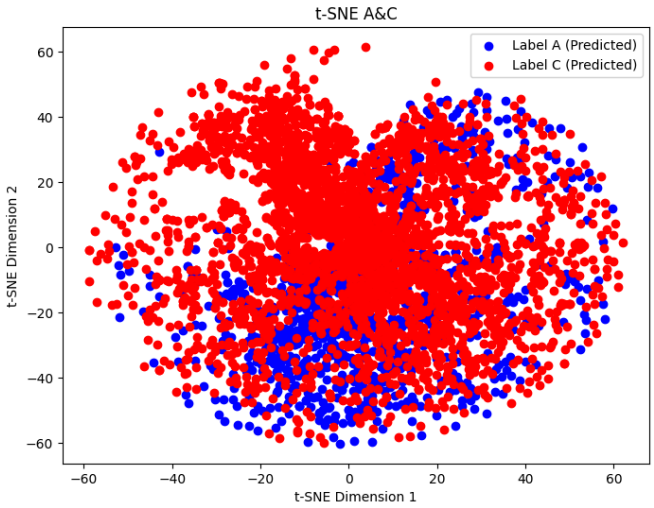
The first exercise set the foundation by focusing on the essential task of data cleaning and preprocessing. Given the diverse nature of textual data, which often includes various forms of noise such as punctuation and special characters, the process of normalizing the text to ensure that each word is treated as a distinct entity was critical. The exercise introduced us to the challenges and solutions in preparing textual data for analysis, highlighting the importance of lemmatization and stemming in reducing words to their base or root form, thereby simplifying subsequent analyses.

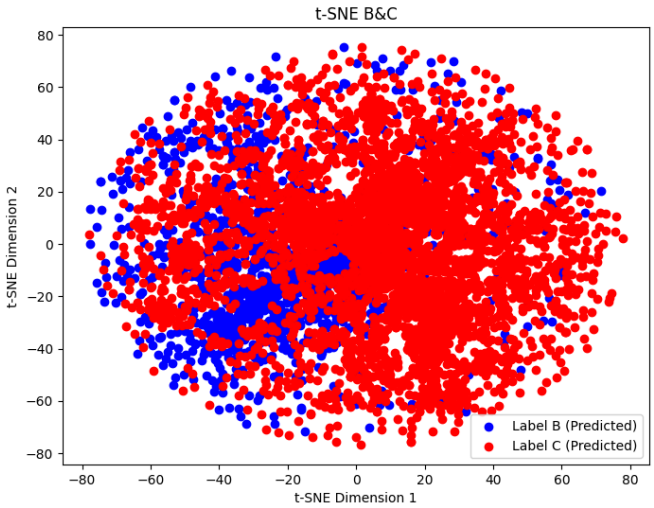
Building upon the cleaned dataset, the second exercise ventured into the realms of clustering and classification. The application of techniques such as k-means clustering, DBSCAN, and machine learning models like Artificial Neural Networks (ANN) provided practical insights into the segmentation of texts based on thematic or stylistic similarities and differences. This exercise not only reinforced the significance of feature extraction methods, including TF-IDF and word2vec, but also underscored the value of understanding the intricacies of various clustering and classification algorithms in grouping and differentiating textual documents. After processing and presenting all the data, the final task was to develop and analyze the sentiment analysis of each text separately.

Unsupervised algorithms

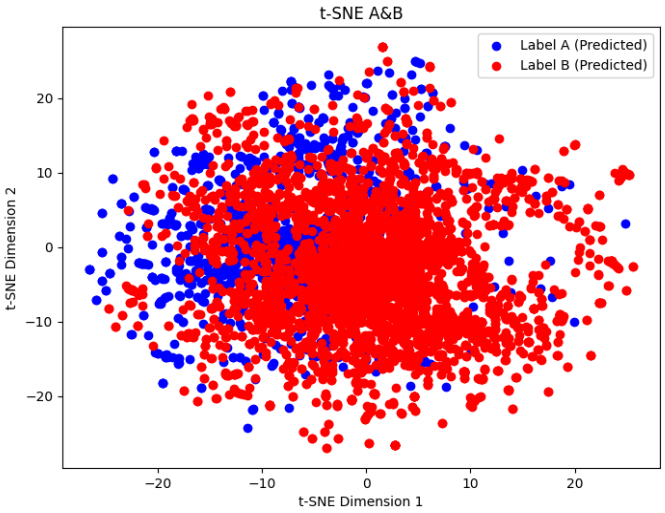
K-means:

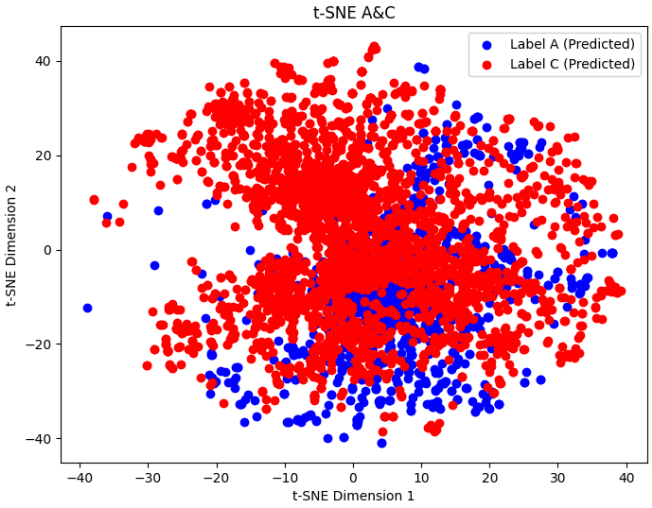
On TF-IDF Words:

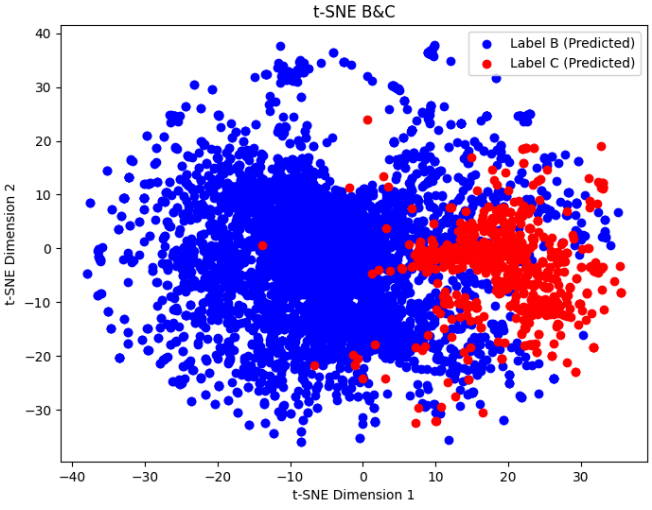




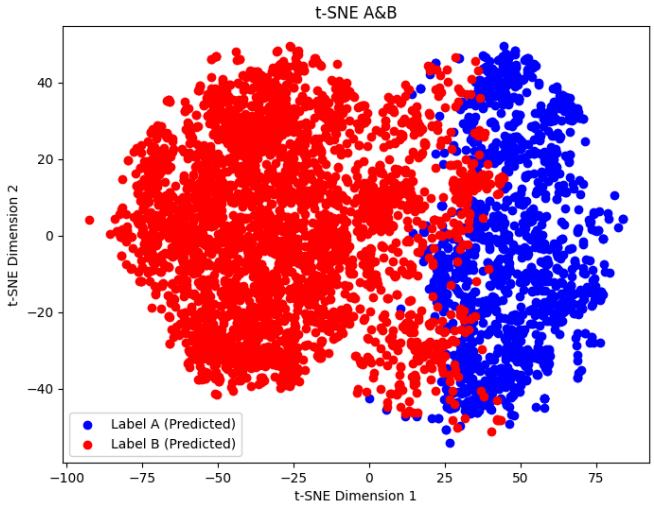
On TF-IDF Lemm:

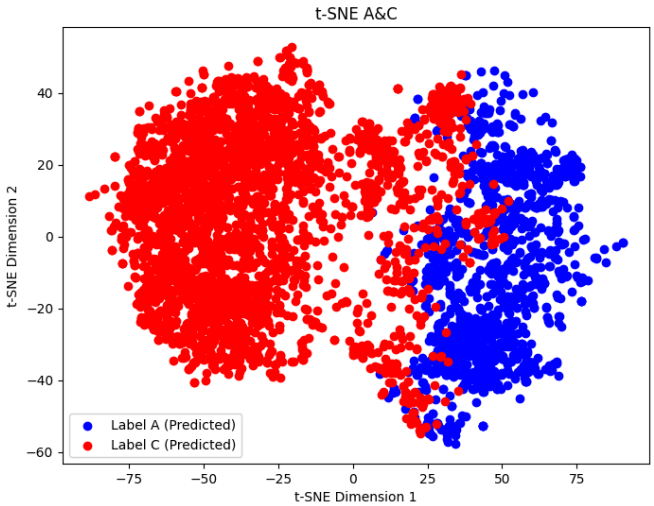


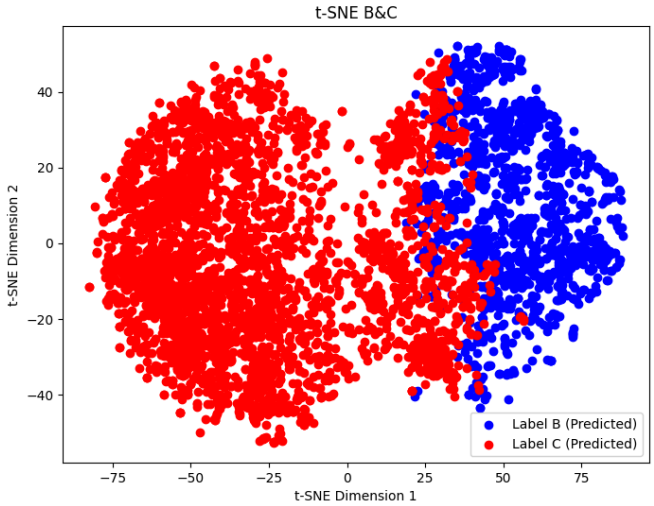




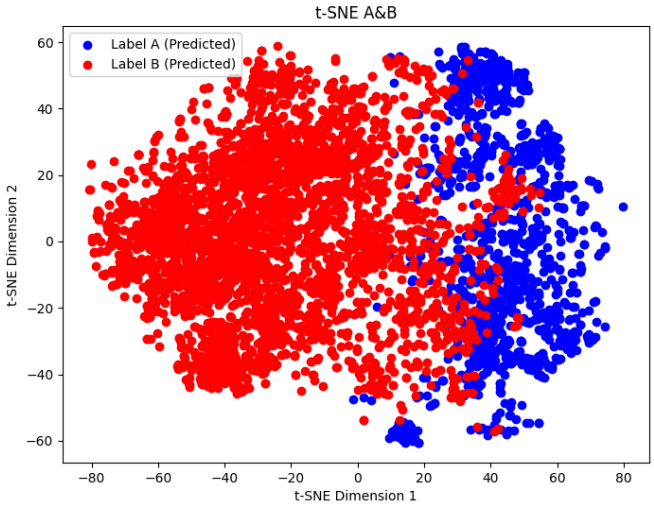
On Word2Vec Lemm(With SW):

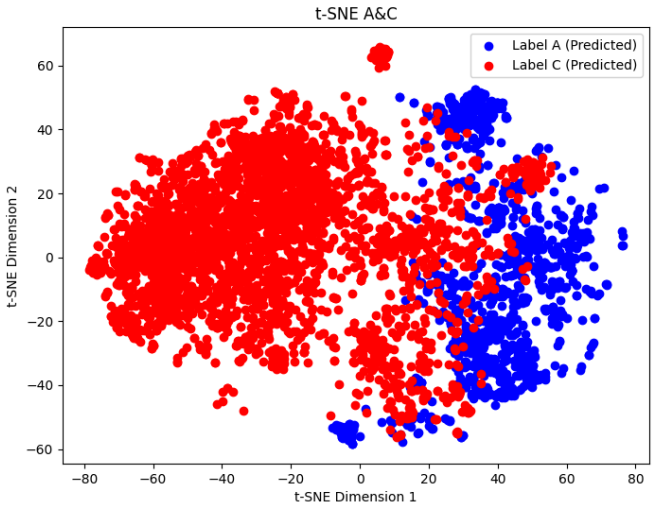


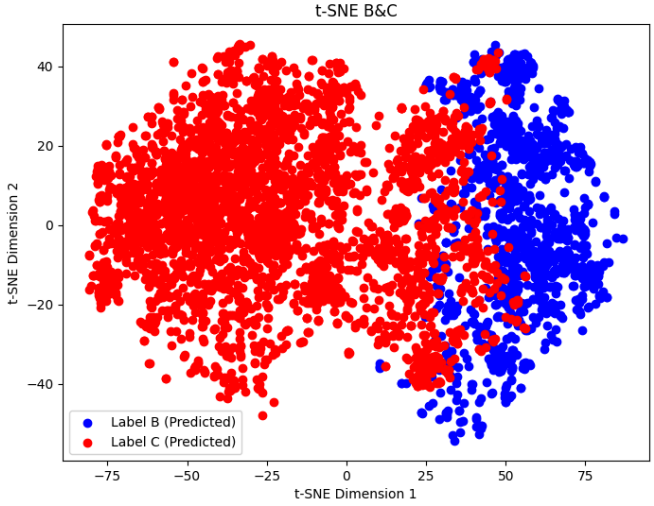




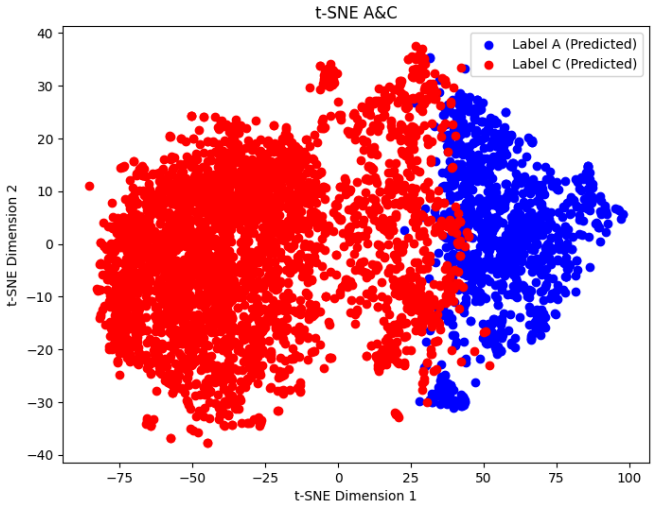
On Word2Vec Lemm(Without SW):

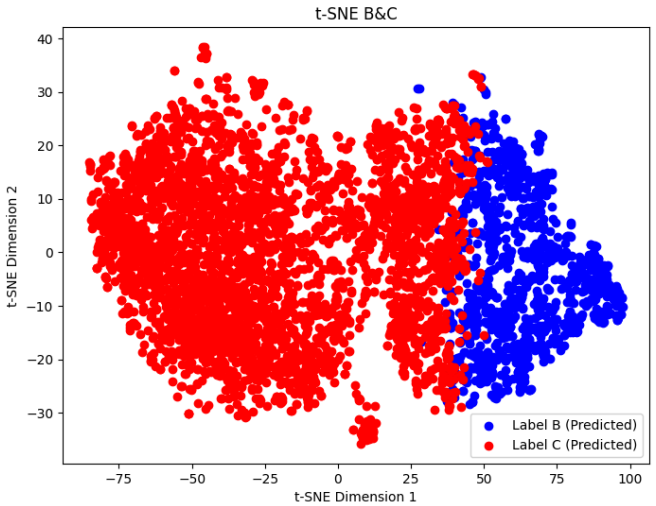


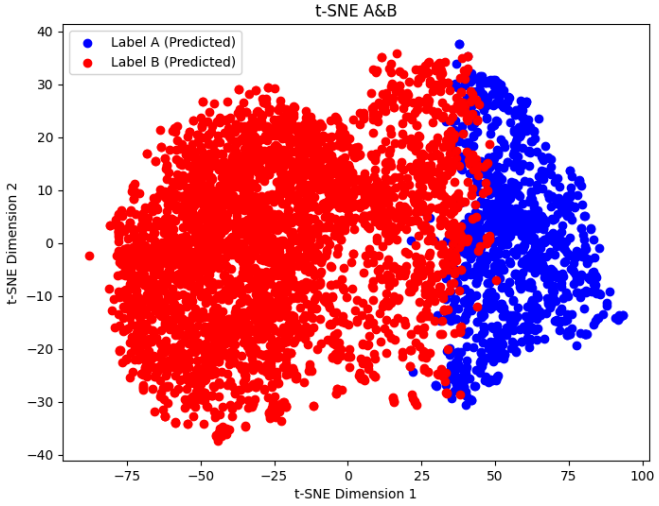




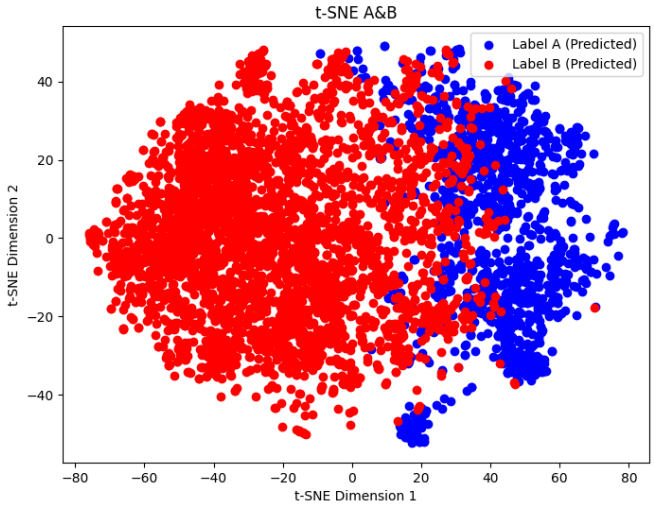
On Word2Vec Words(With SW):

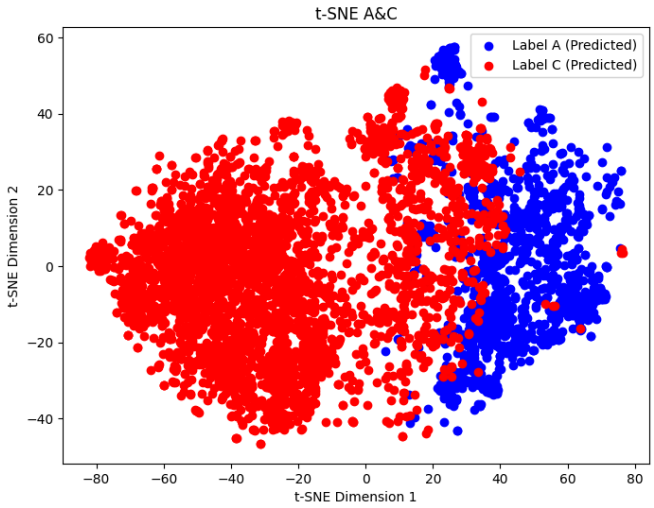


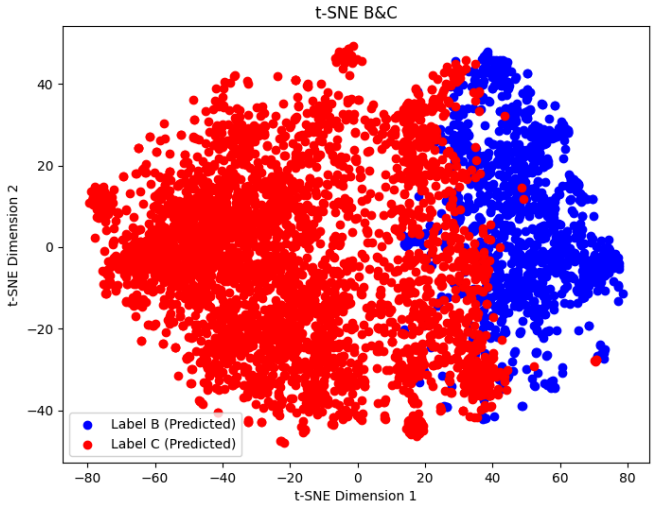




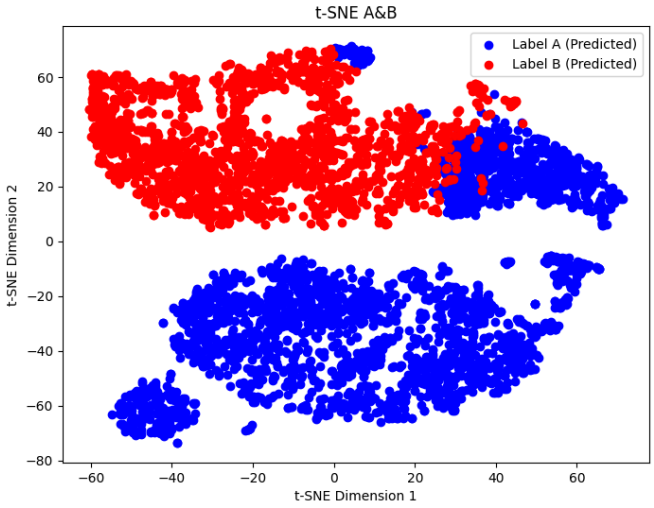
On Word2Vec Words(Without SW):

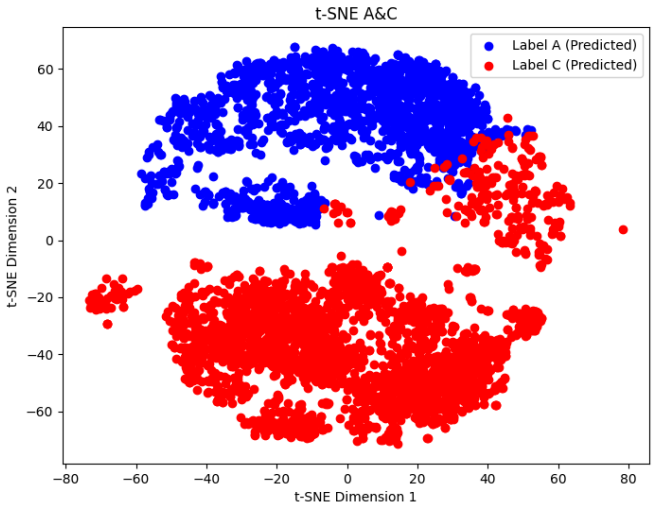


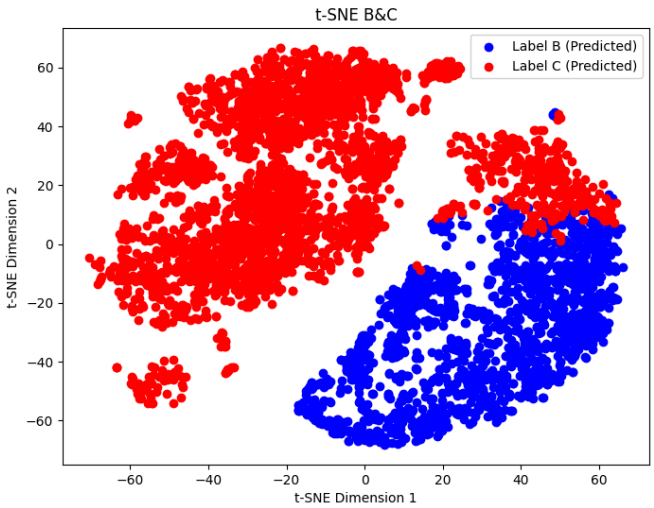




On Bert:


t-SNE A&B
Label A (Predicted)
Label B (Predicted)
t-SNE Dimension 1
t-SNE Dimension 2

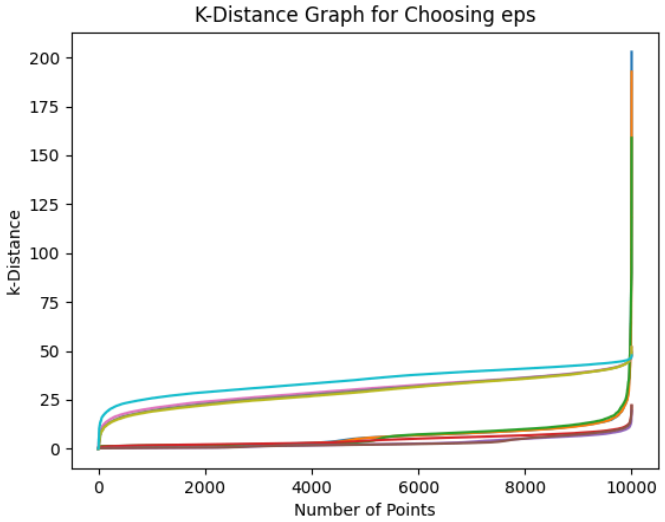

t-SNE A&C
Label A (Predicted)
Label C (Predicted)
t-SNE Dimension 1
t-SNE Dimension 2

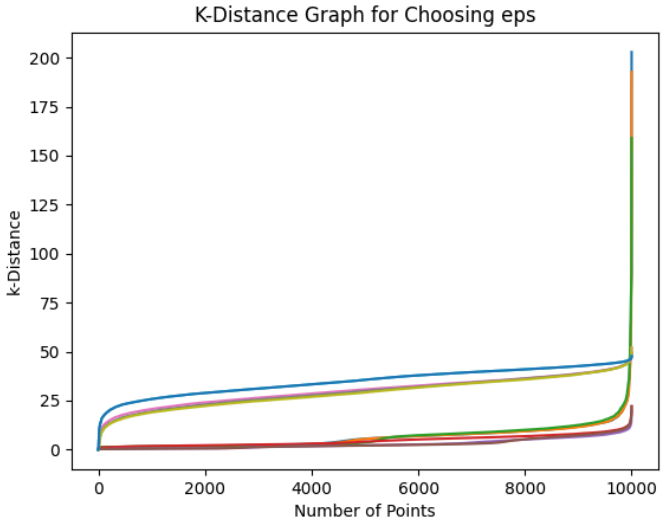

t-SNE B&C
Label B (Predicted)
Label C (Predicted)
t-SNE Dimension 1
t-SNE Dimension 2

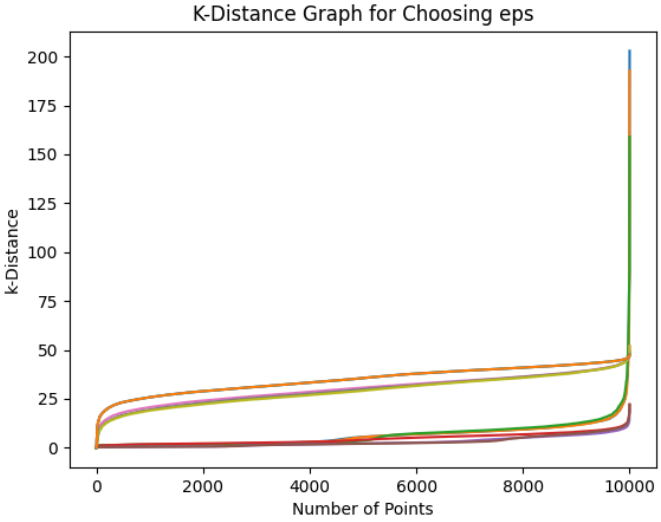
DBSCAN:

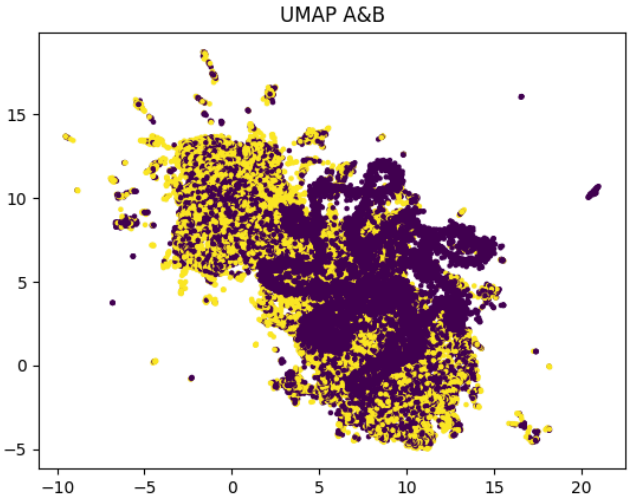
(The elbow plots are in order AB-AC-BC)

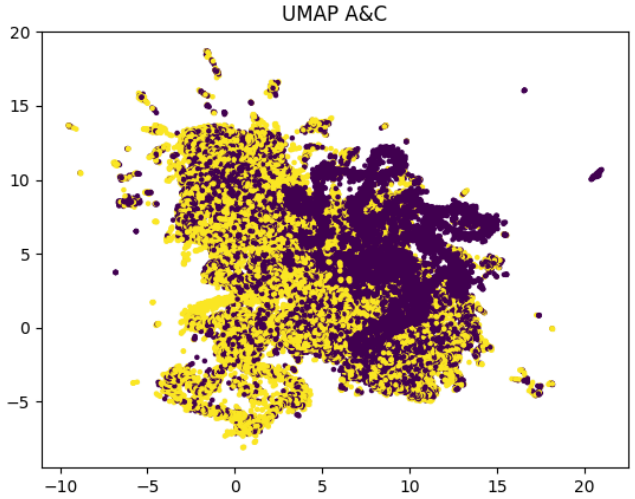
On TF-IDF Words:

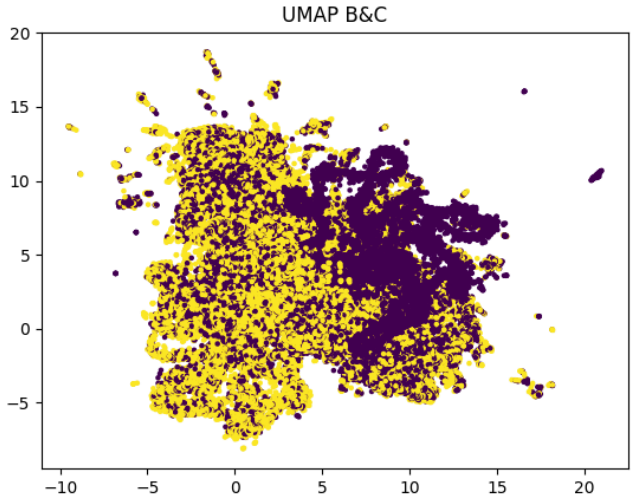




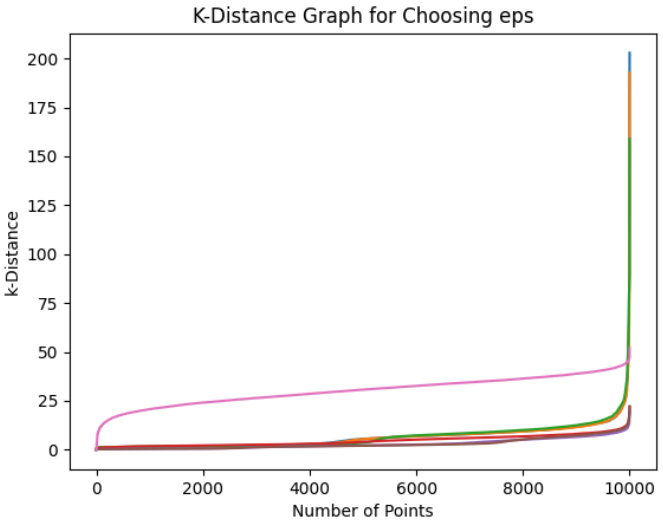

K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
25
50
75
100
125
150
175
200
0
2000
4000
6000
8000
10000

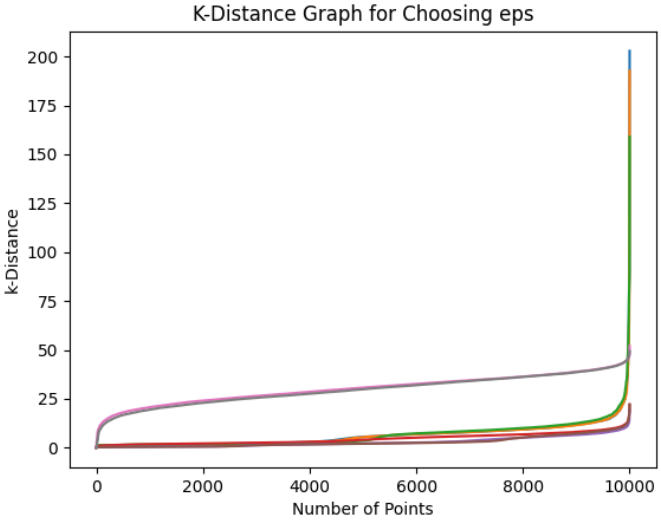

UMAP A&B
15
10
5
0
−5
−10
−5
0
5
10
15
20

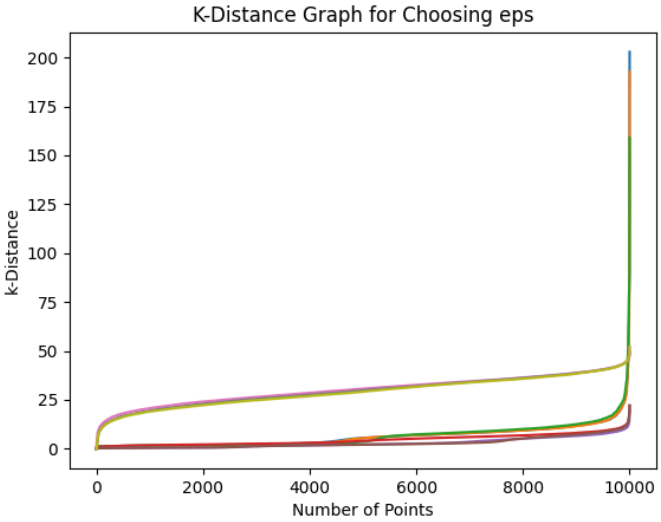


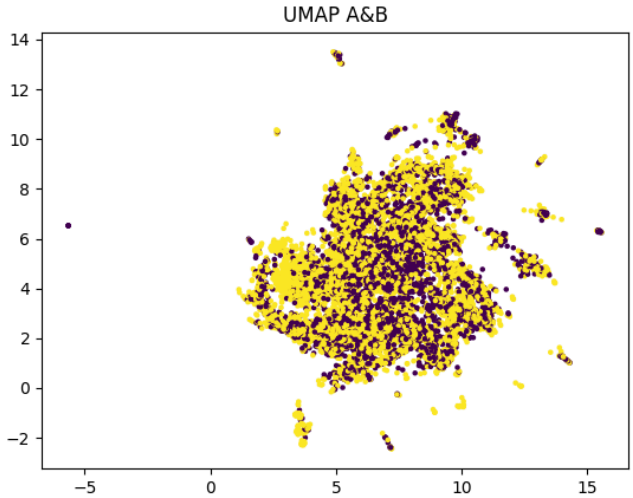


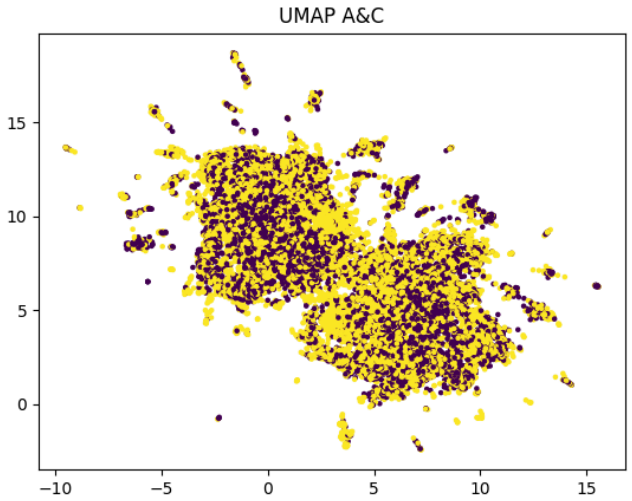
On TF-IDF Lemm:

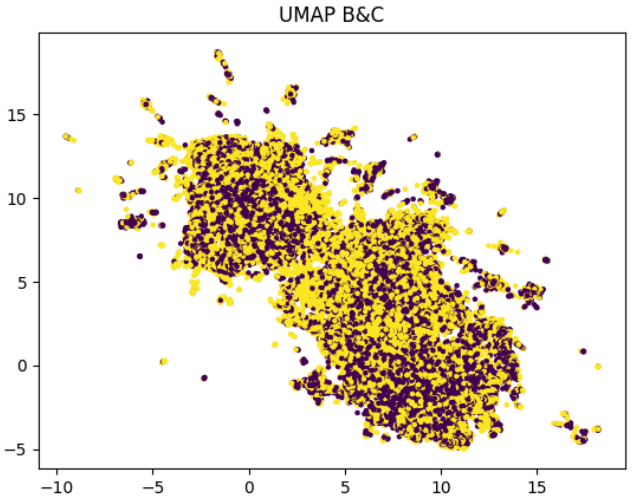



K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
25
50
75
100
125
150
175
200
0
2000
4000
6000
8000
10000

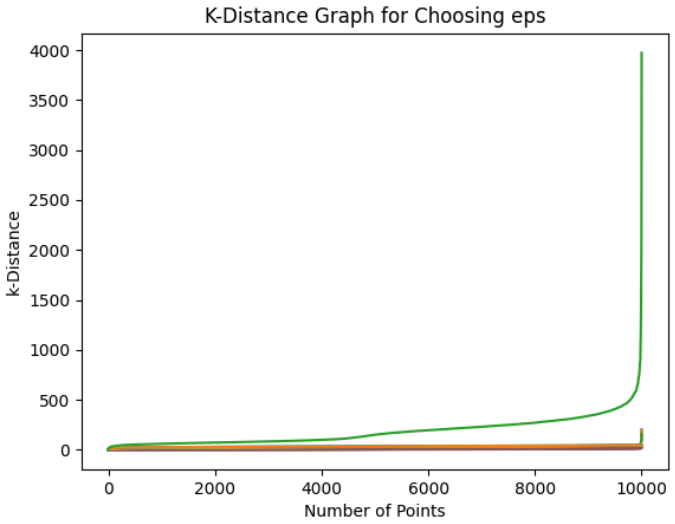

K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
25
50
75
100
125
150
175
200
0
2000
4000
6000
8000
10000

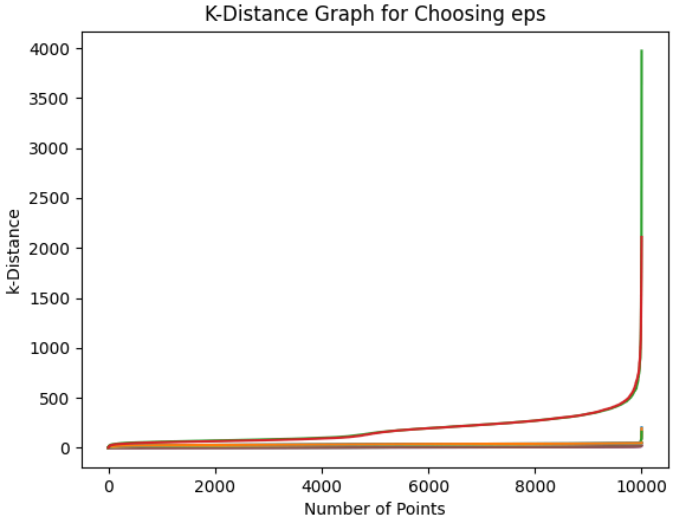

UMAP A&B
14
12
10
8
6
4
2
0
−2
−5
0
5
10
15

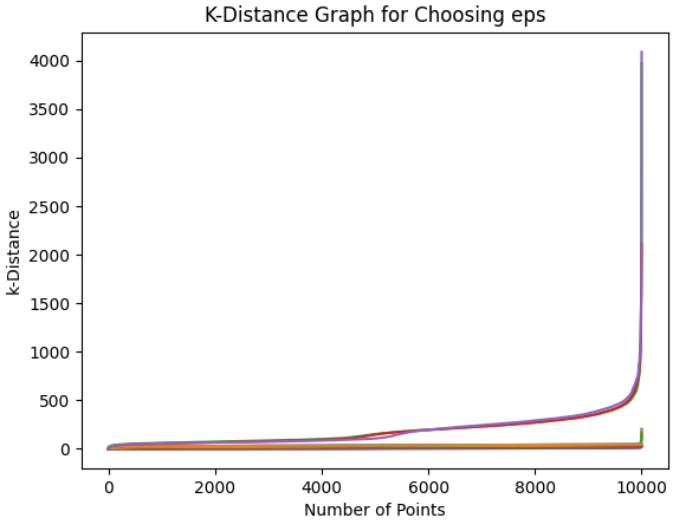


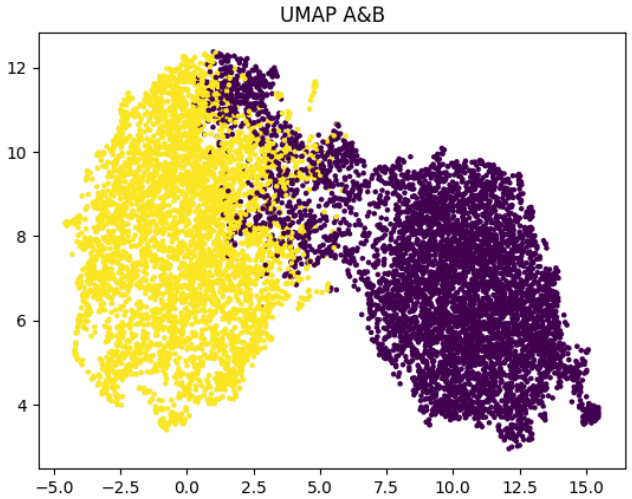


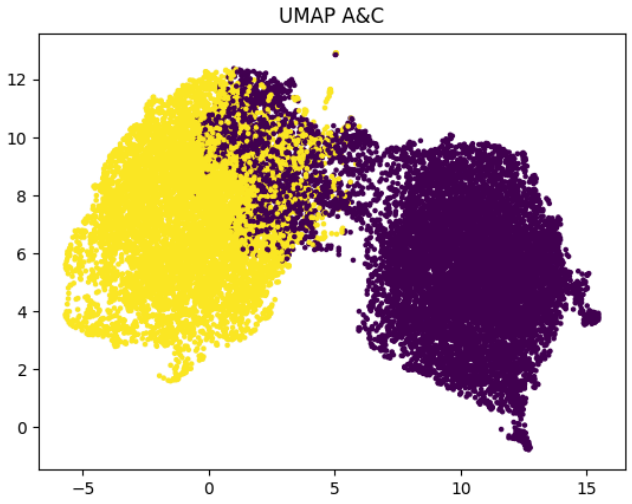
On Word2Vec Lemm(With SW):

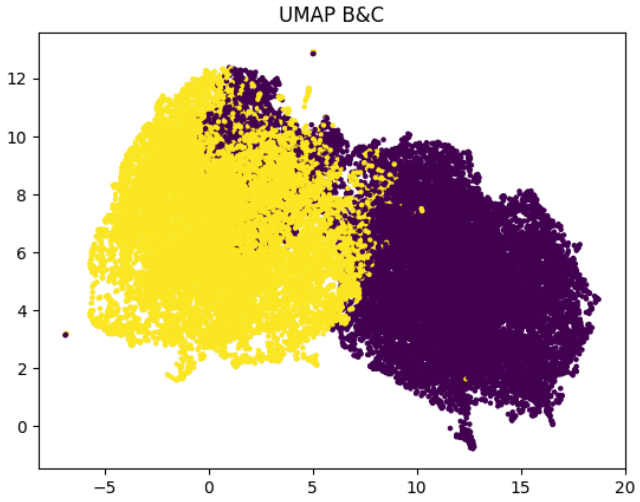



K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
500
1000
1500
2000
2500
3000
3500
4000
0
2000
4000
6000
8000
10000

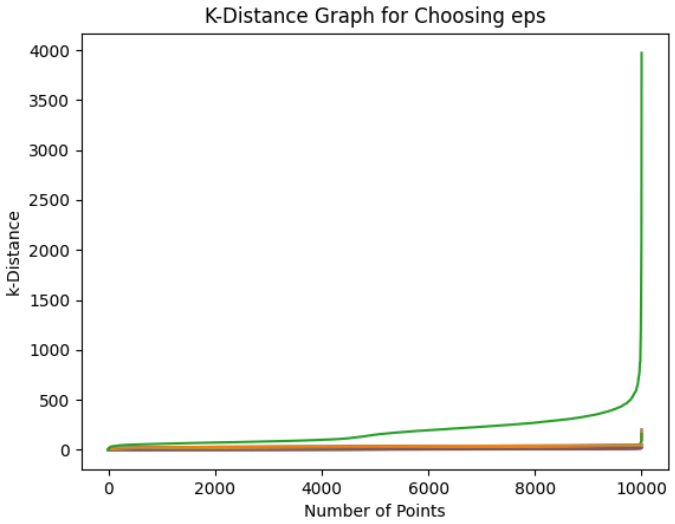

K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
500
1000
1500
2000
2500
3000
3500
4000
0
2000
4000
6000
8000
10000

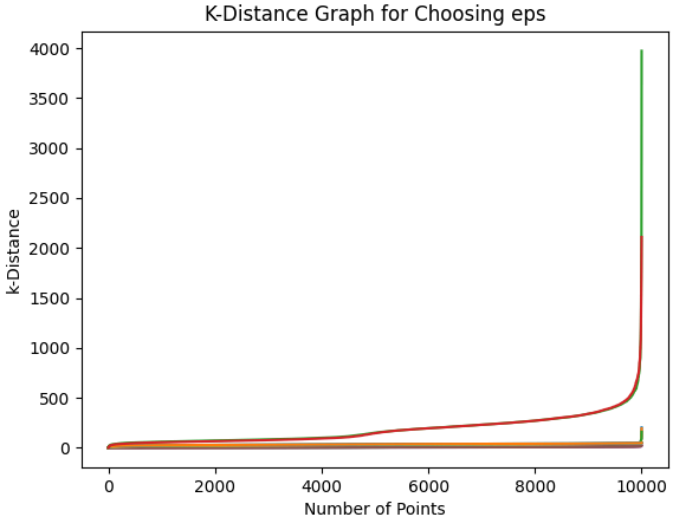

UMAP A&B
4
6
8
10
12
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

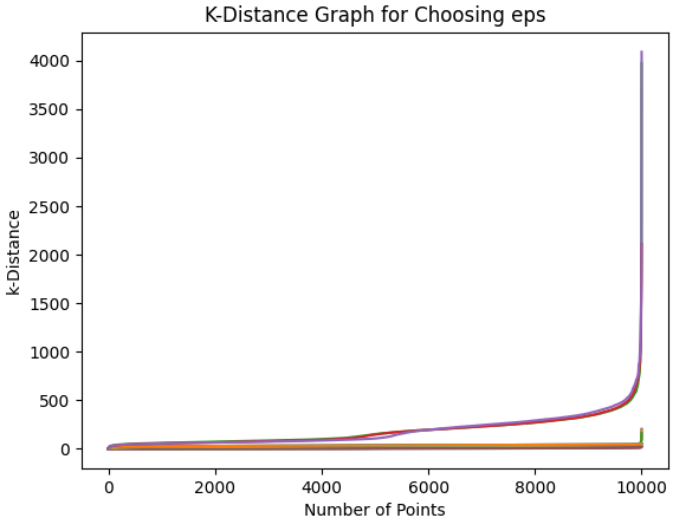


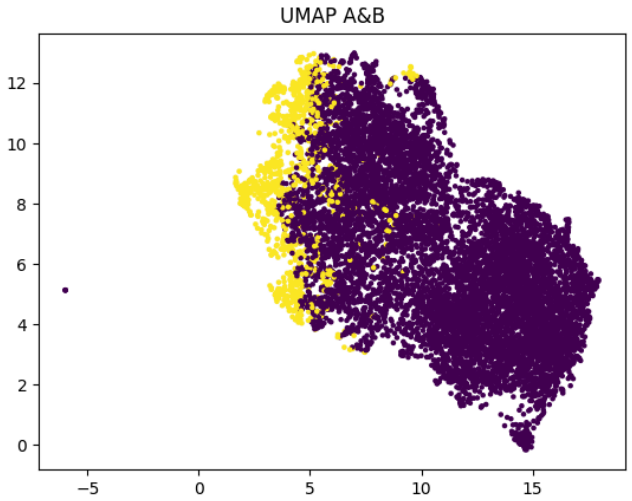


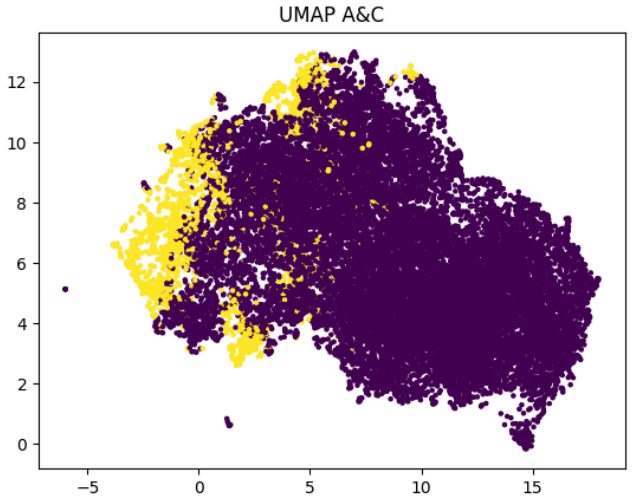
On Word2Vec Lemm(Without SW):

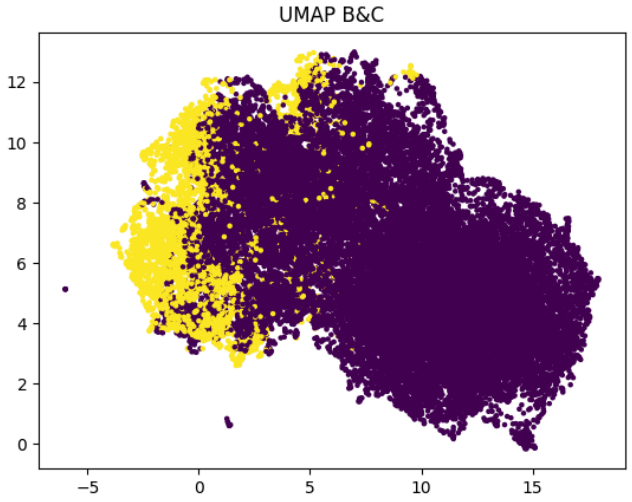



K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
500
1000
1500
2000
2500
3000
3500
4000
0
2000
4000
6000
8000
10000

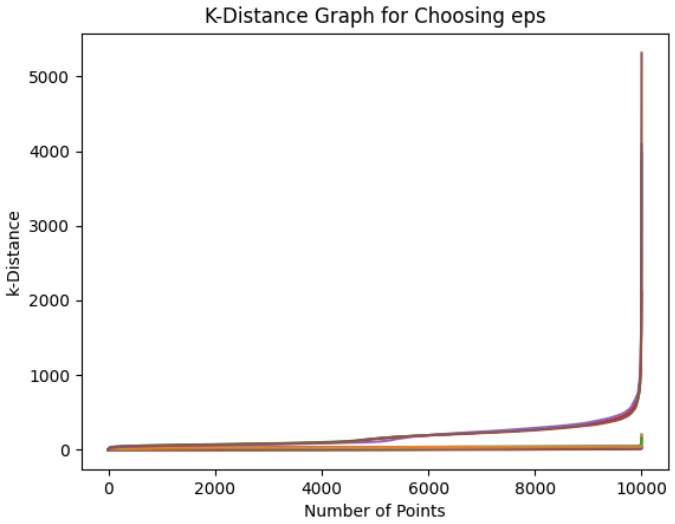

K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
500
1000
1500
2000
2500
3000
3500
4000
0
2000
4000
6000
8000
10000

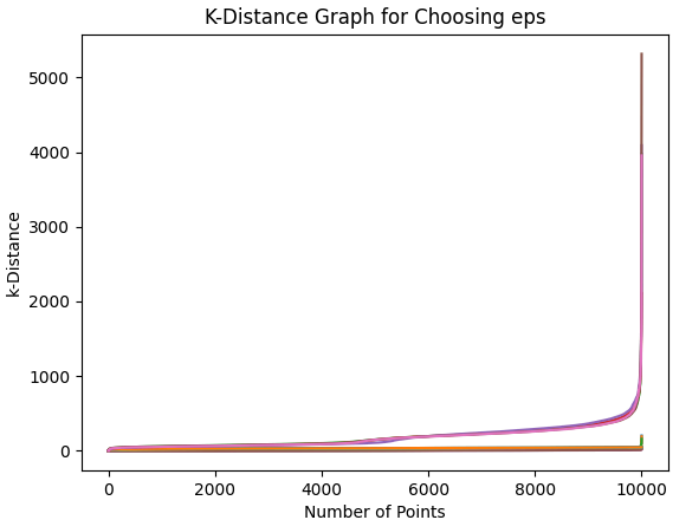

UMAP A&B
12
10
8
6
4
2
0
−5
0
5
10
15

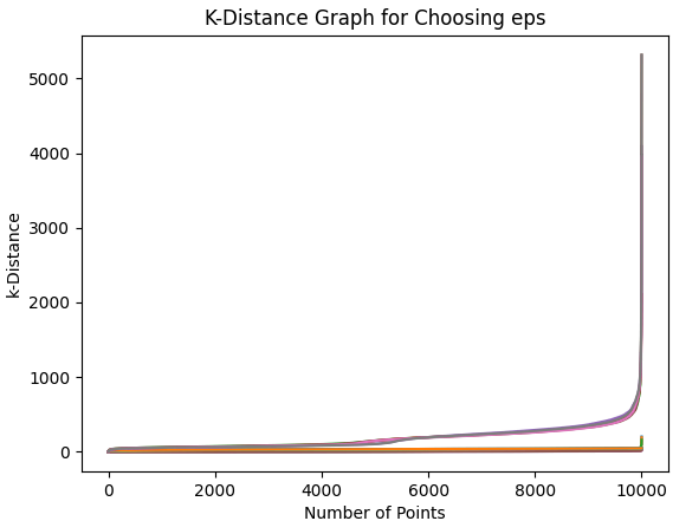

UMAP A&C
12
10
8
6
4
2
0
−5
0
5
10
15

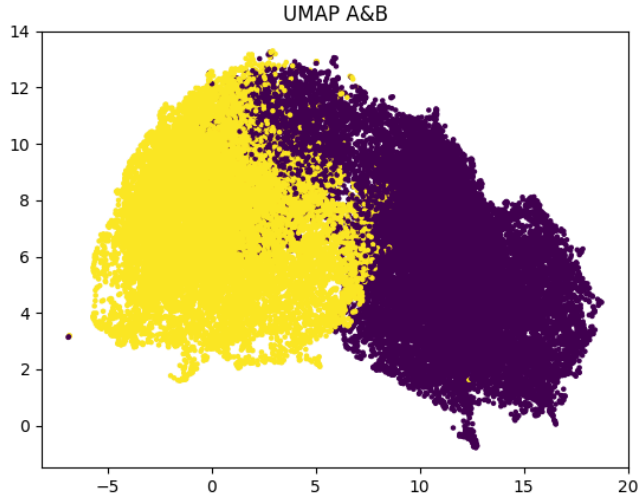

UMAP B&C
12
10
8
6
4
2
0
−5
0
5
10
15

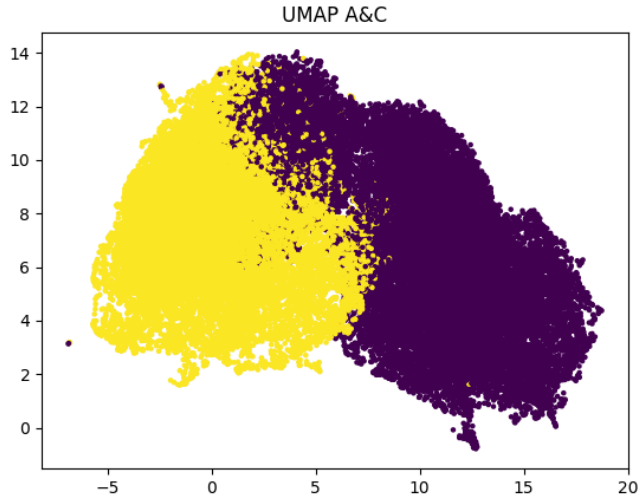
On Word2Vec Words(With SW):

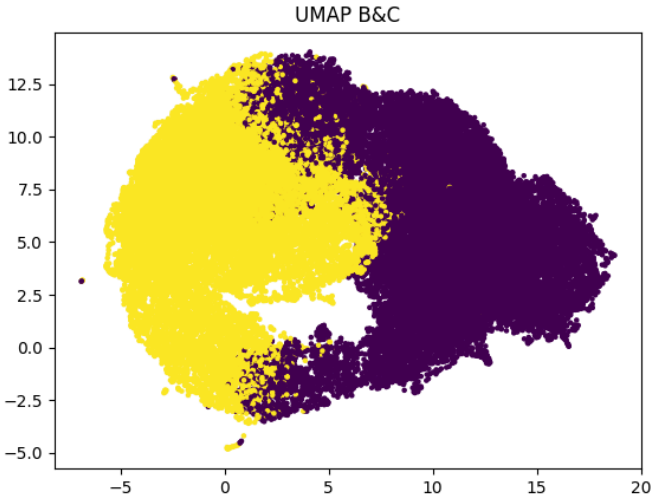




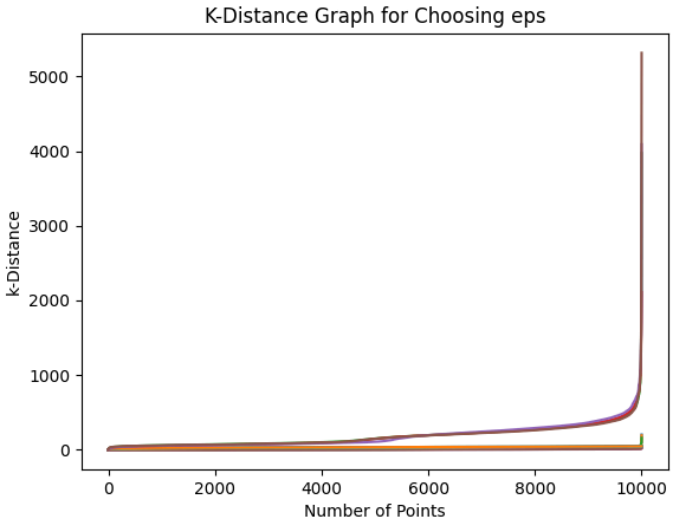


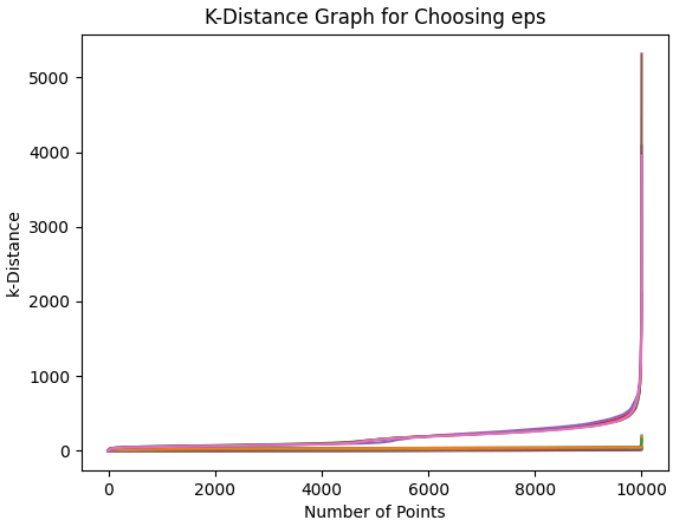

UMAP A&B
14
12
10
8
6
4
2
0
−5
0
5
10
15
20

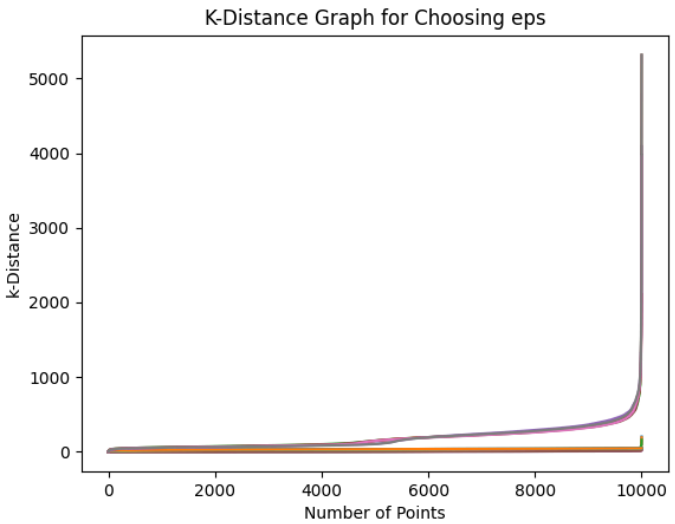

UMAP A&C
14
12
10
8
6
4
2
0
−5
0
5
10
15
20

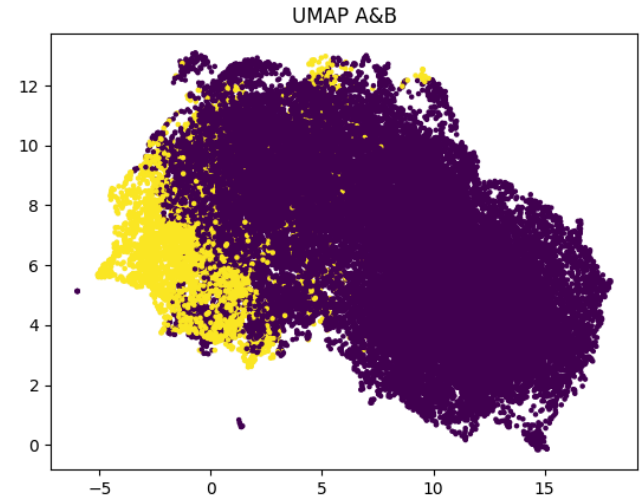

UMAP B&C
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−5
0
5
10
15
20

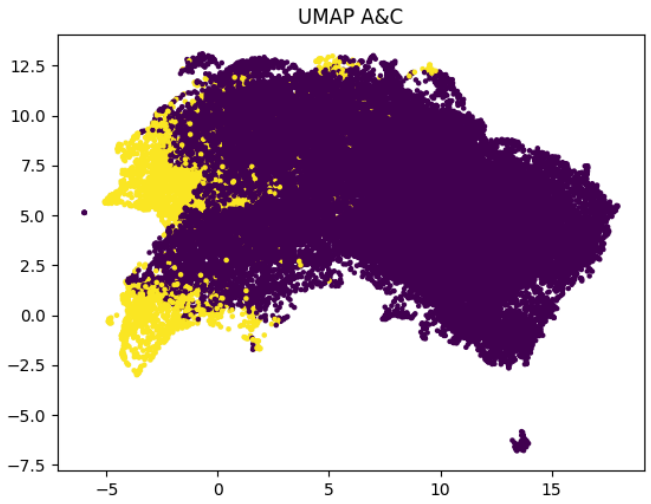
On Word2Vec Words(Without SW):

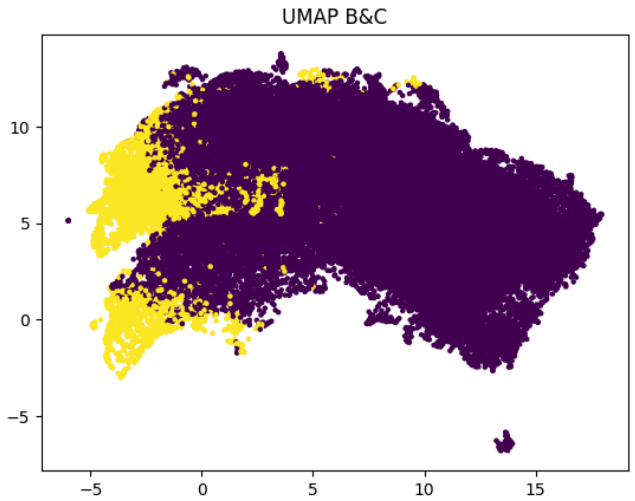




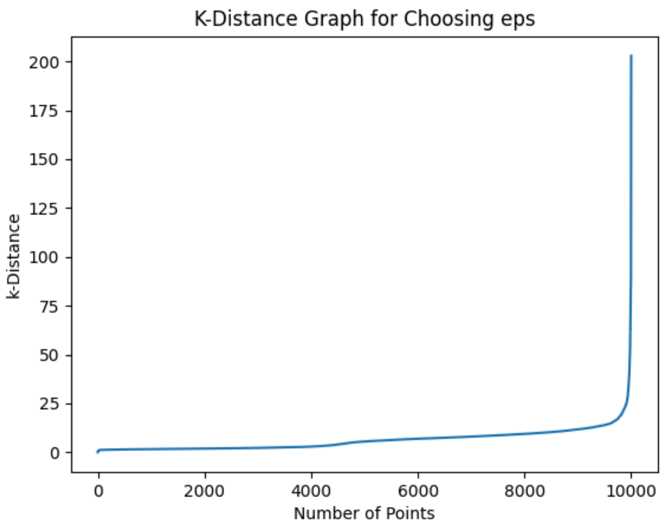


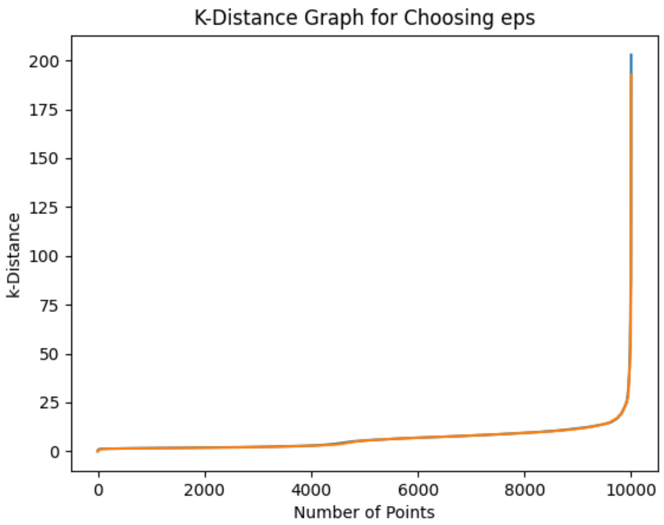

UMAP A&B
12
10
8
6
4
2
0
−5
0
5
10
15

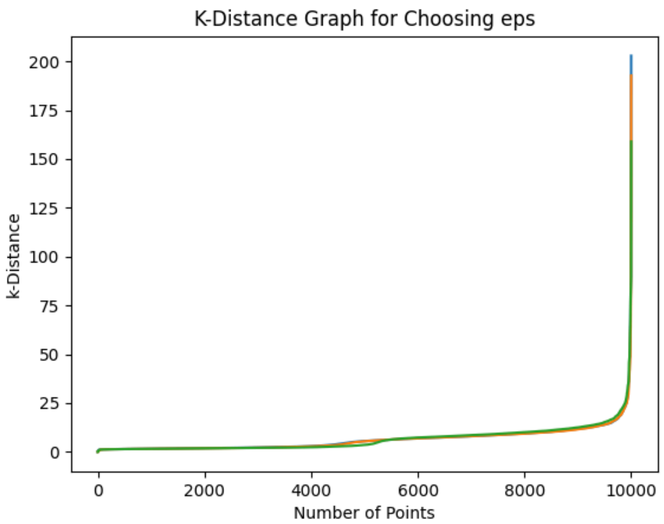

UMAP A&C
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−5
0
5
10
15

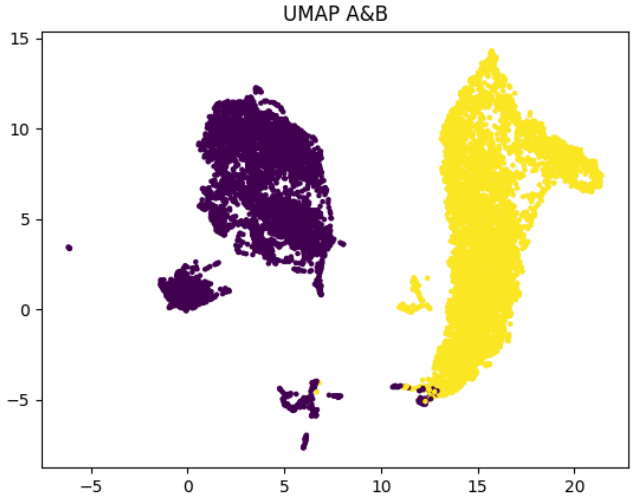

UMAP B&C
10
5
0
−5
−5
0
5
10
15

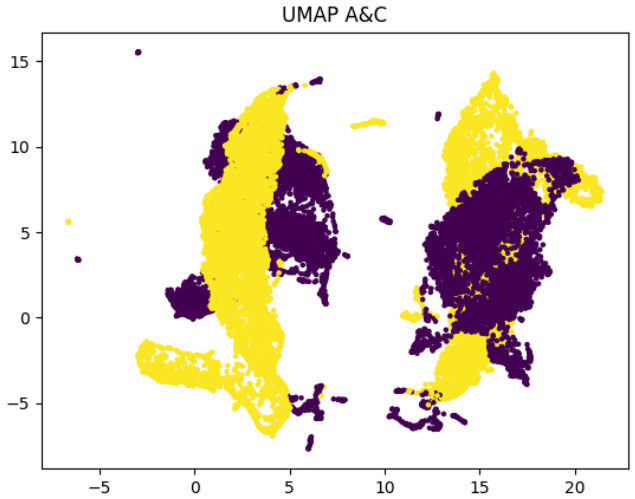
On Bert:

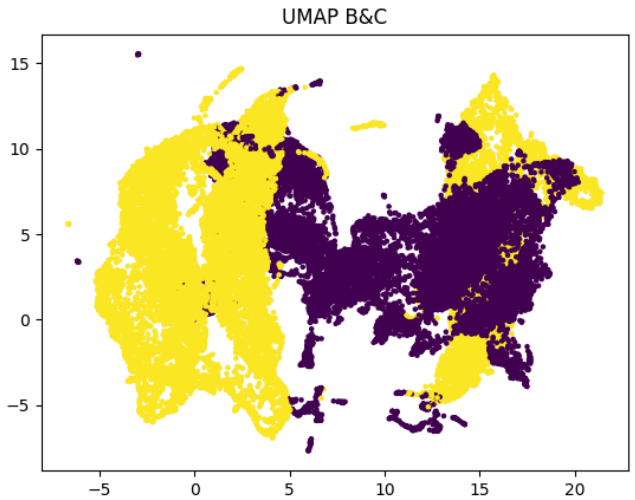

K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
25
50
75
100
125
150
175
200
0
2000
4000
6000
8000
10000


K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
25
50
75
100
125
150
175
200
0
2000
4000
6000
8000
10000


K-Distance Graph for Choosing eps
k-Distance
Number of Points
0
25
50
75
100
125
150
175
200
0
2000
4000
6000
8000
10000

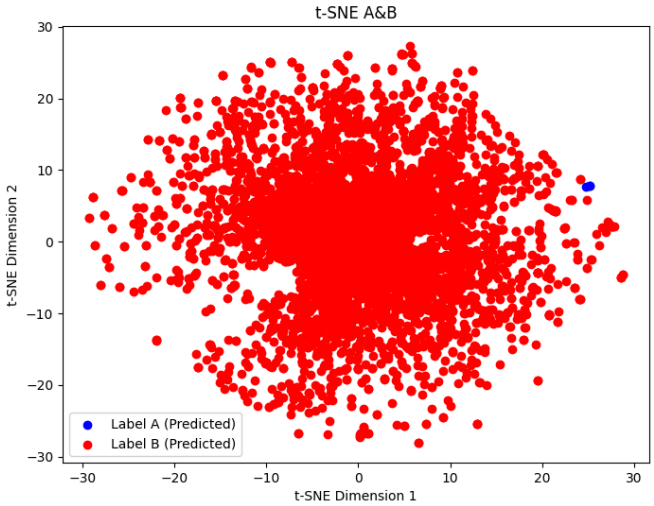

UMAP A&B
15
10
5
0
−5
−5
0
5
10
15
20

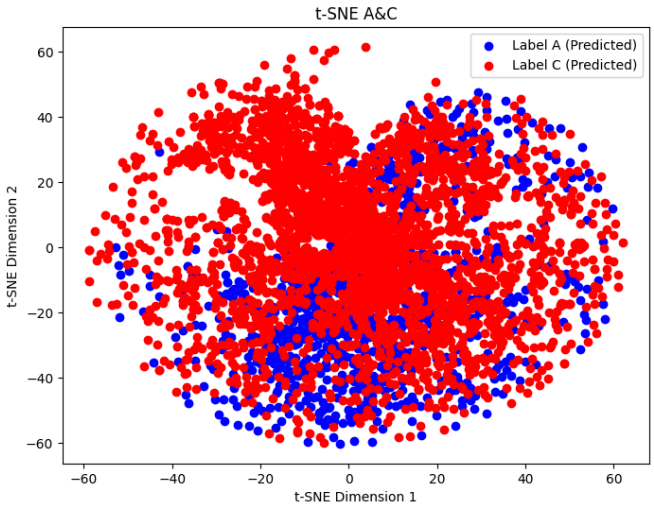

UMAP A&C
15
10
5
0
−5
−5
0
5
10
15
20

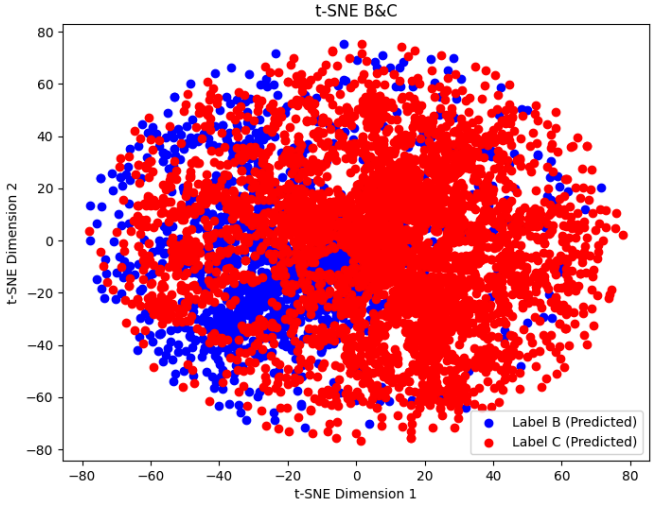

UMAP B&C
15
10
5
0
−5
−5
0
5
10
15
20

GMM:

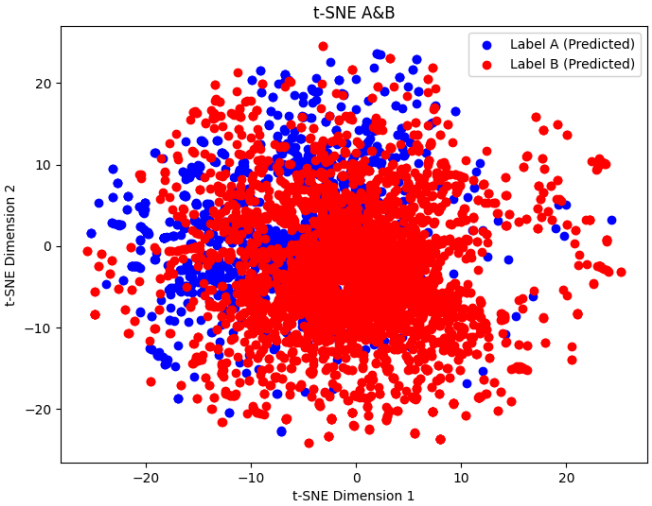
On TF-IDF Words:

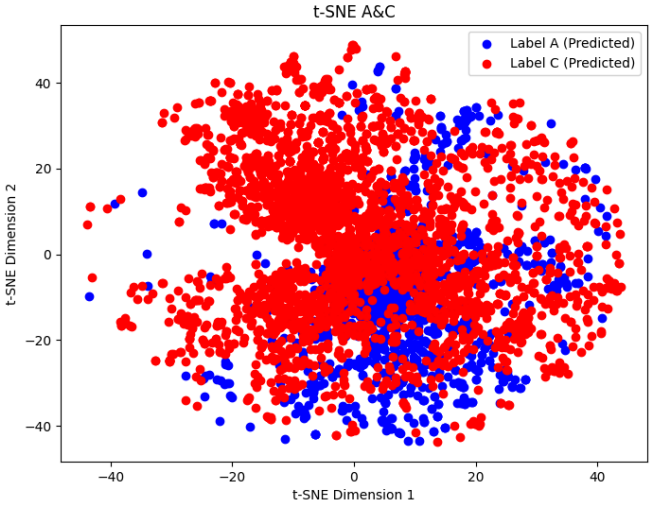


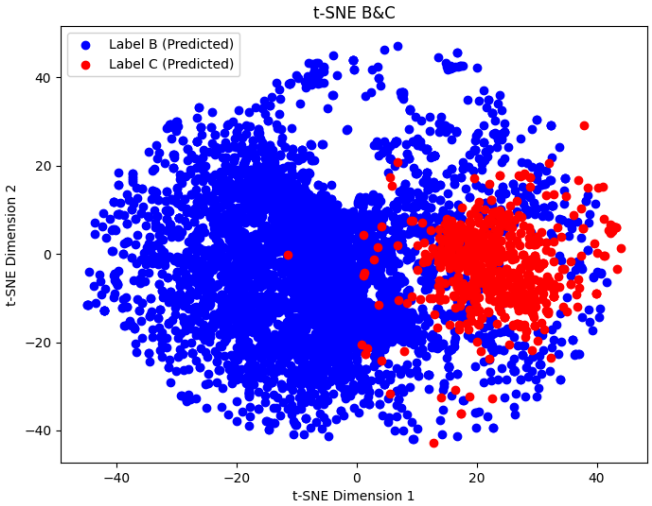




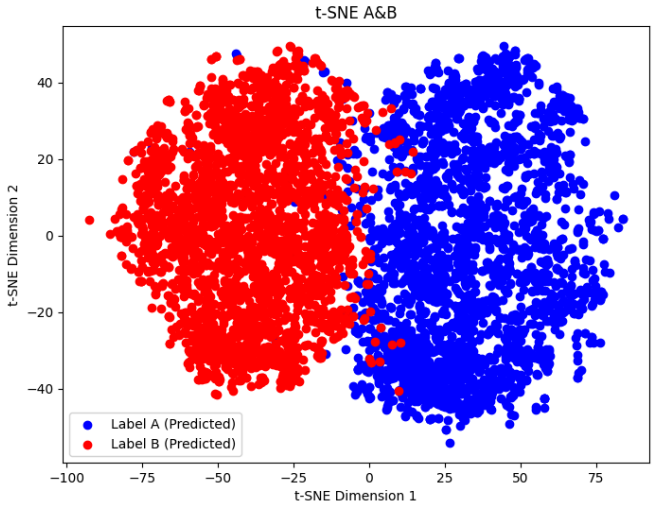
On TF-IDF Lemm:

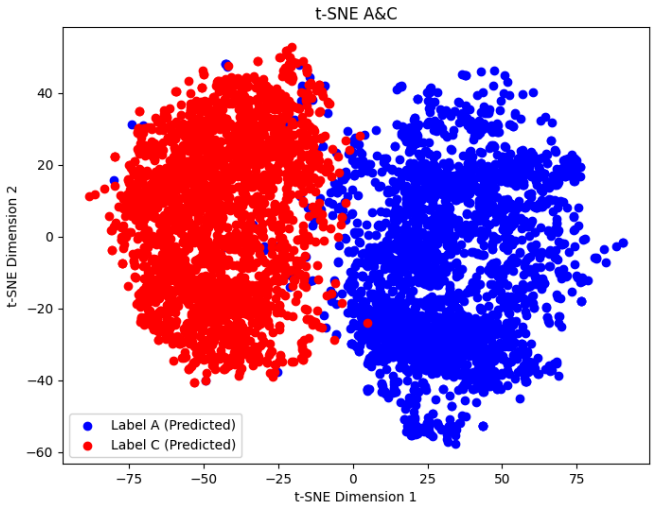


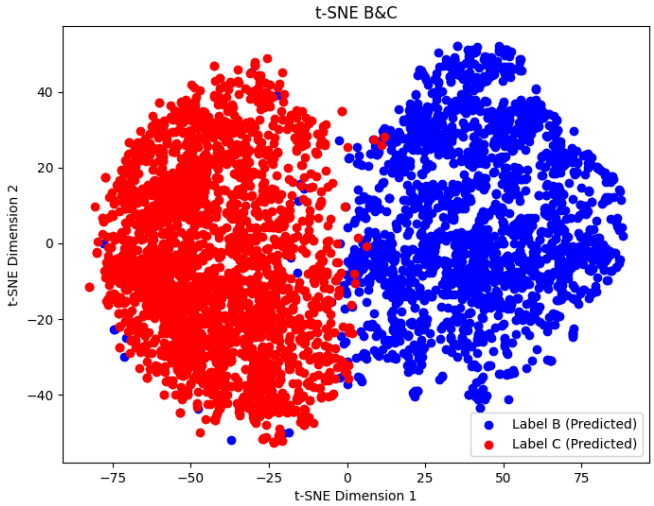




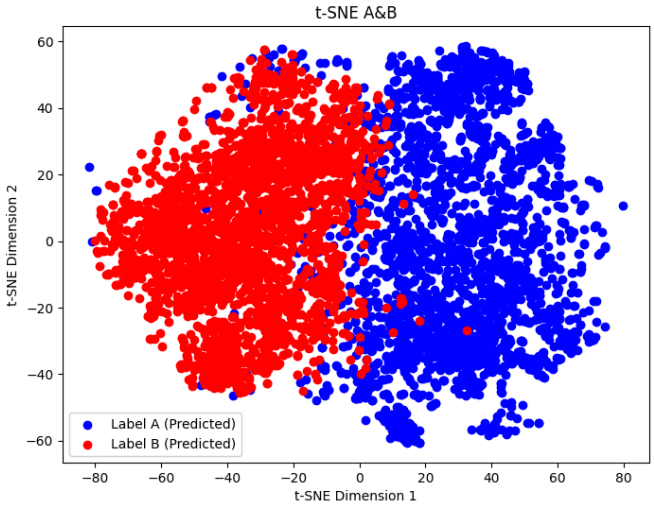
On Word2Vec Lemm(With SW):

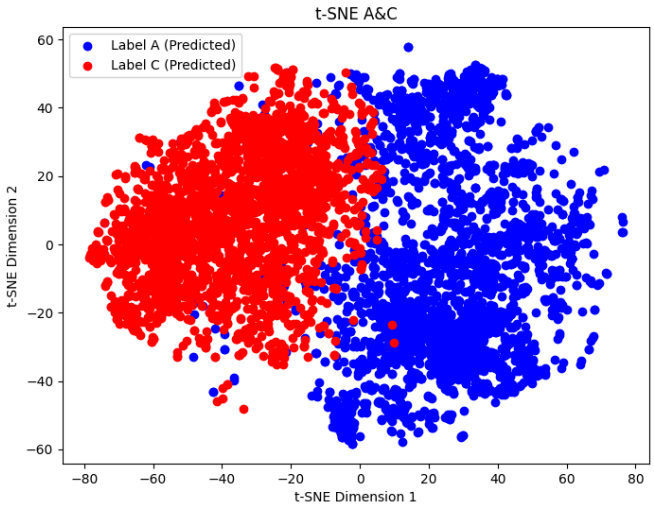


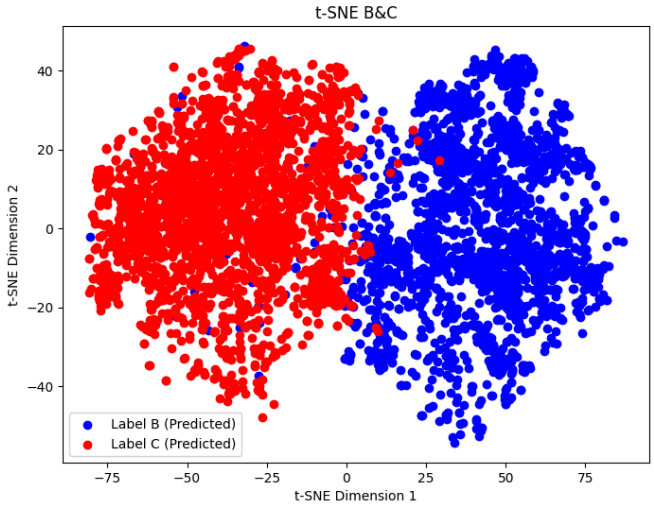




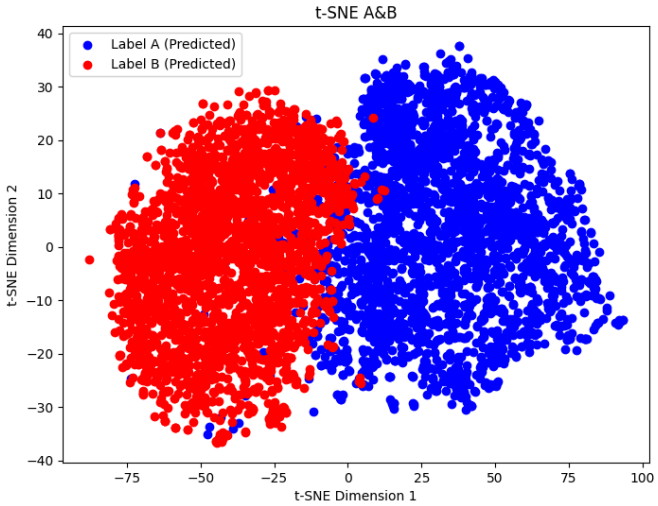
On Word2Vec Lemm(Without SW):

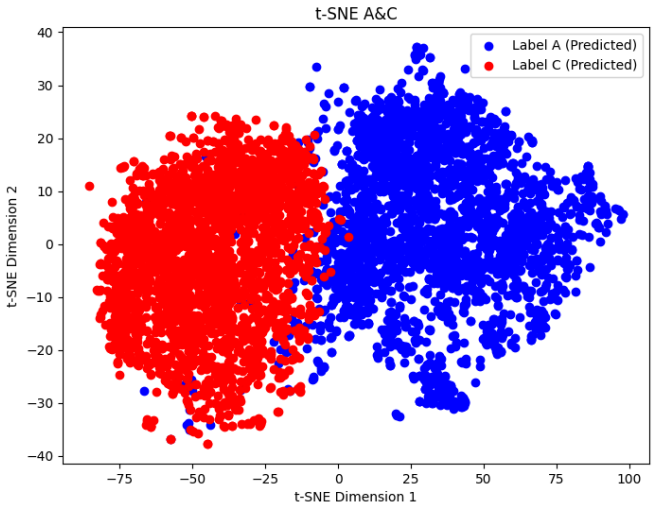


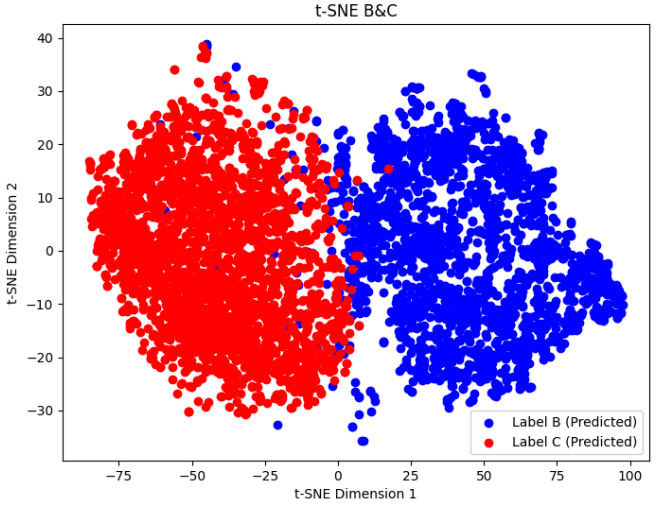




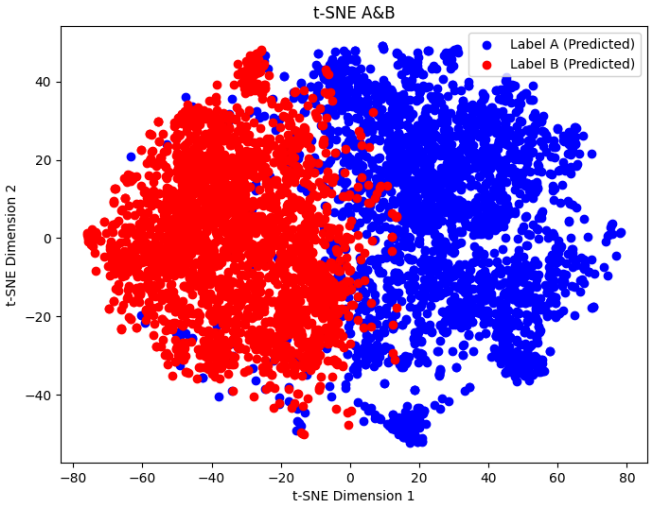
On Word2Vec Words(With SW):

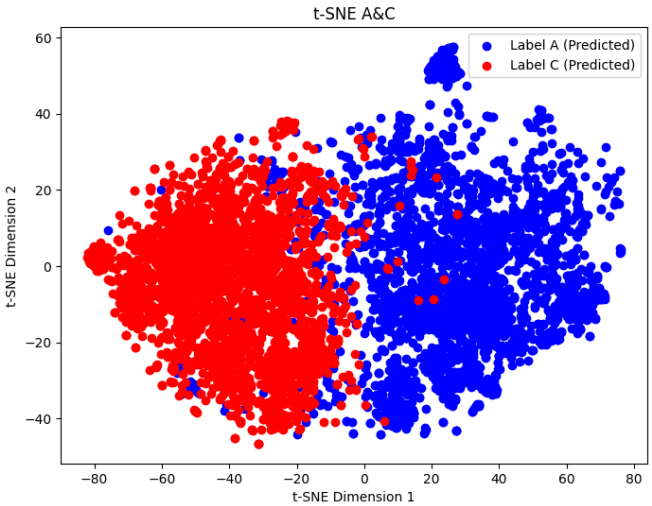


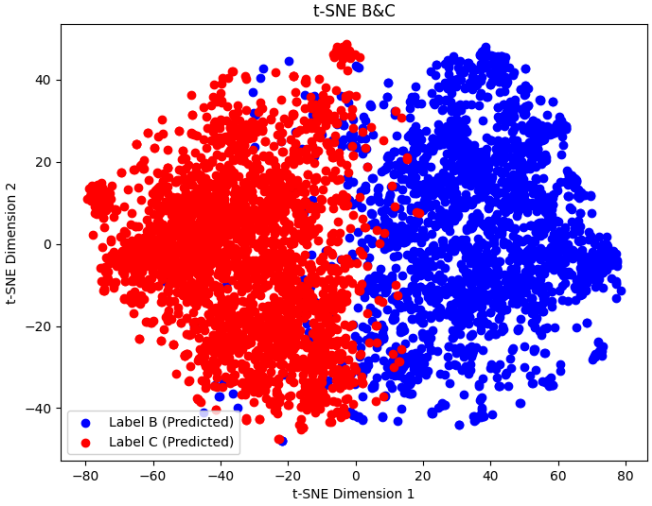


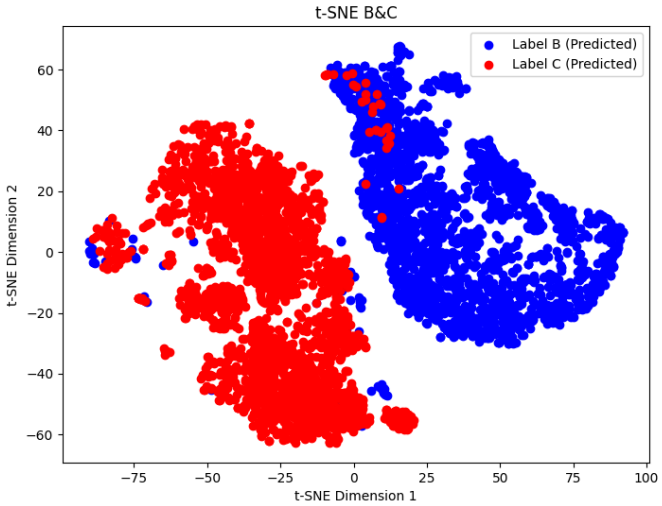


On Word2Vec Words(Without SW):

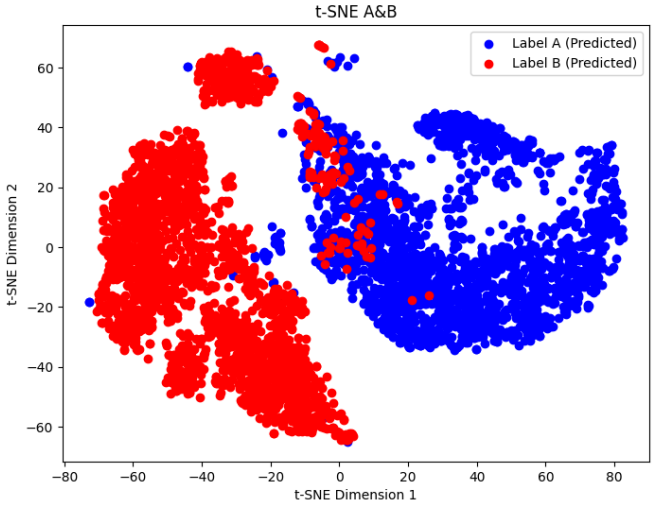


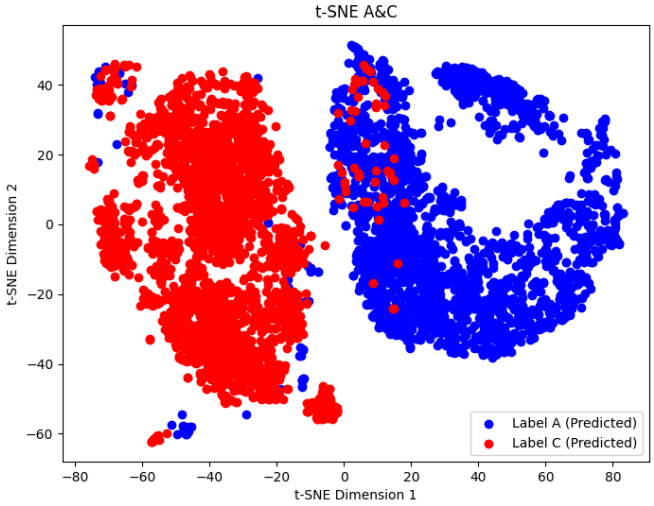





t-SNE B&C
Label B (Predicted)
Label C (Predicted)
t-SNE Dimension 2
t-SNE Dimension 1

On Bert:

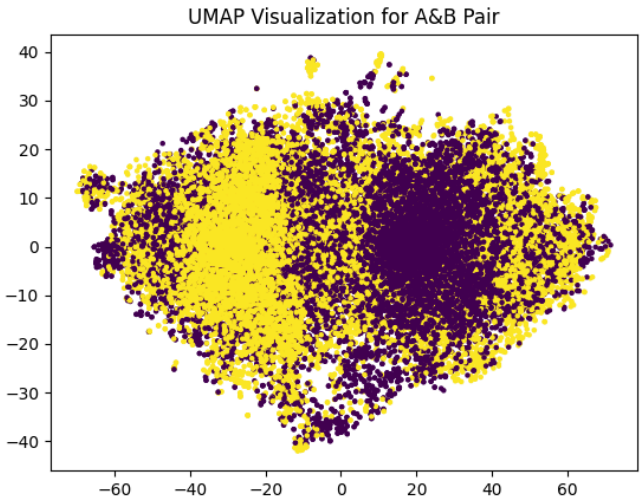

t-SNE A&B
Label A (Predicted)
Label B (Predicted)
t-SNE Dimension 2
t-SNE Dimension 1

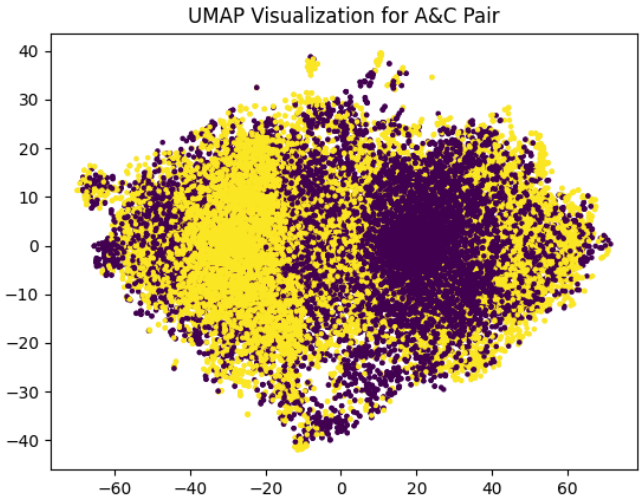

t-SNE A&C
Label A (Predicted)
Label C (Predicted)
t-SNE Dimension 2
t-SNE Dimension 1

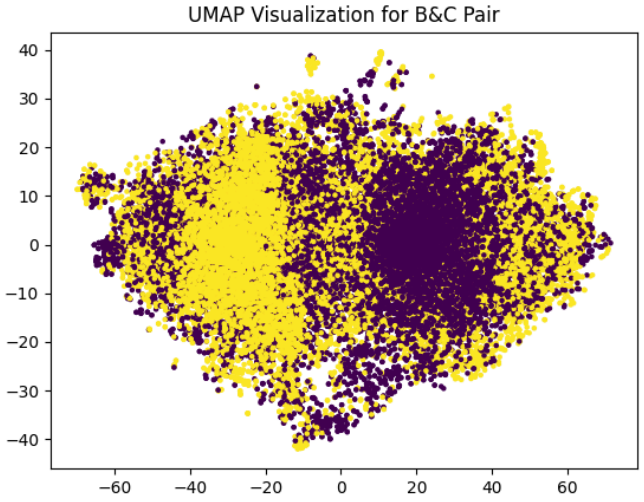
Supervised algorithms

NB:

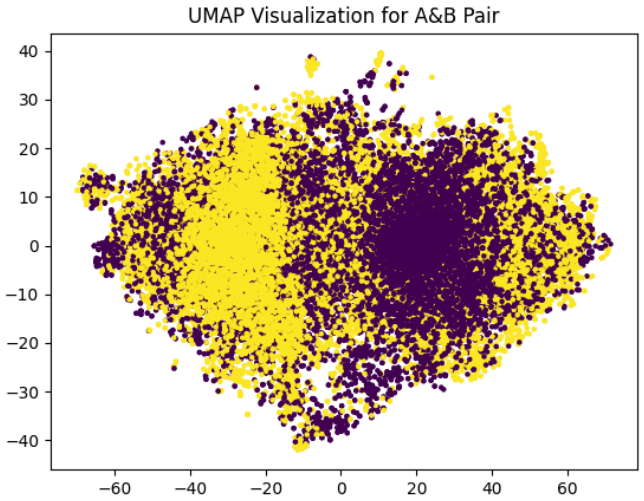
On TF-IDF Words:

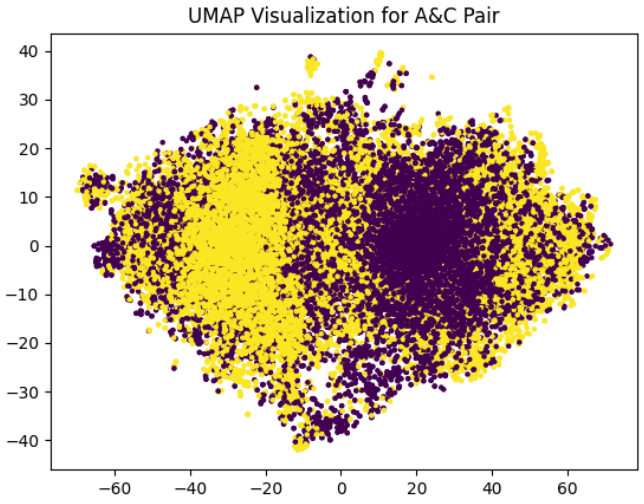


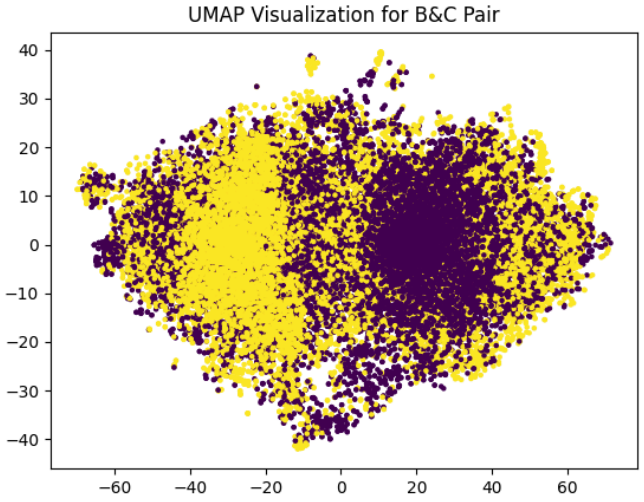




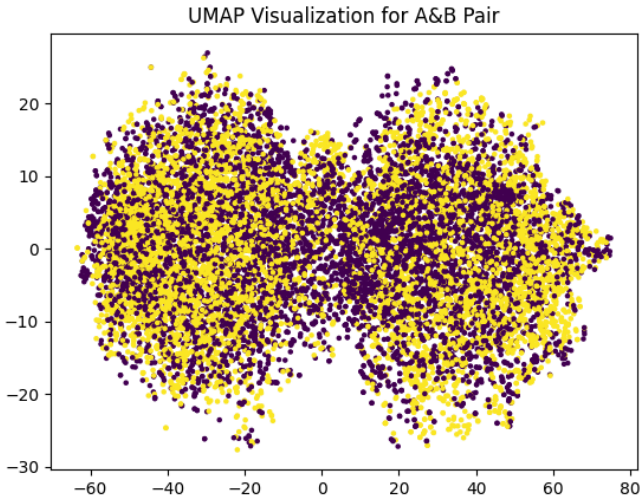
On TF-IDF Lemm:

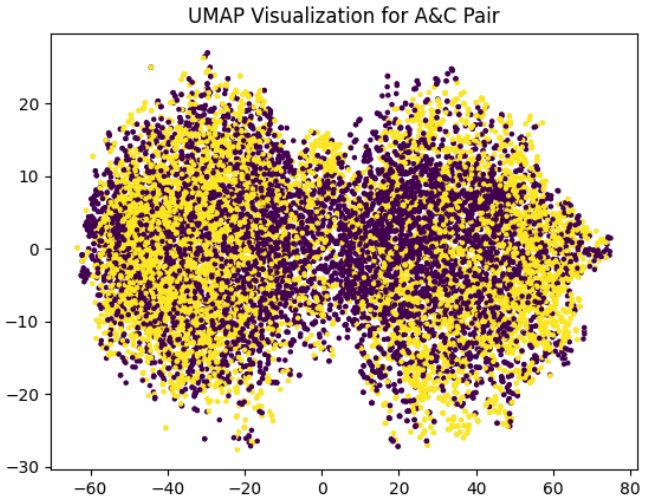


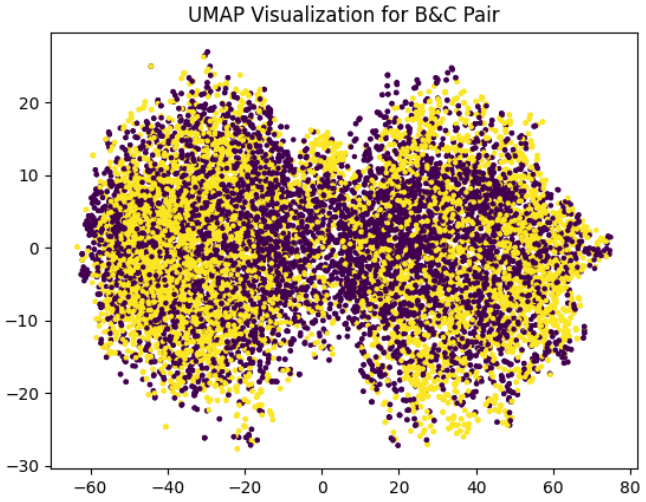




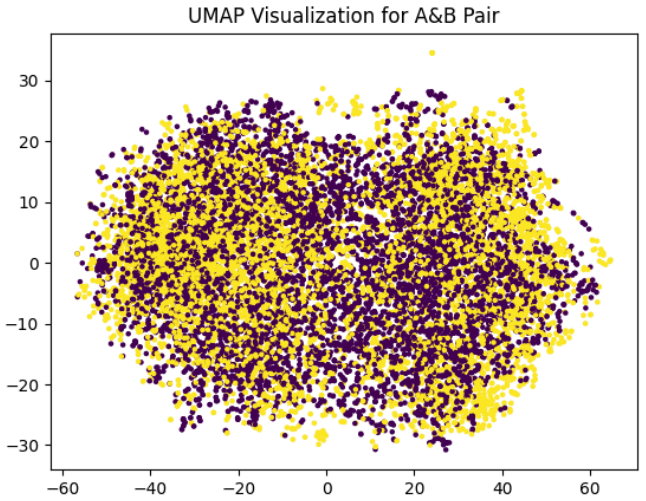
On Word2Vec Lemm(With SW):

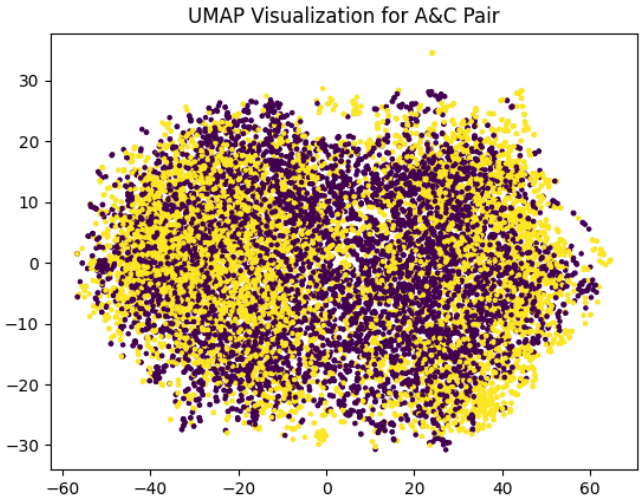


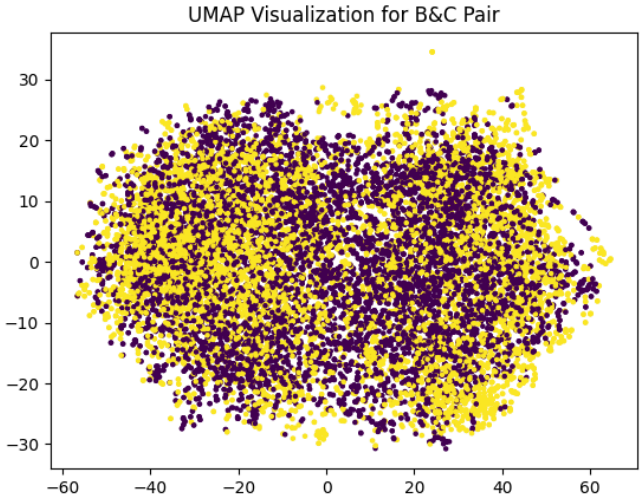




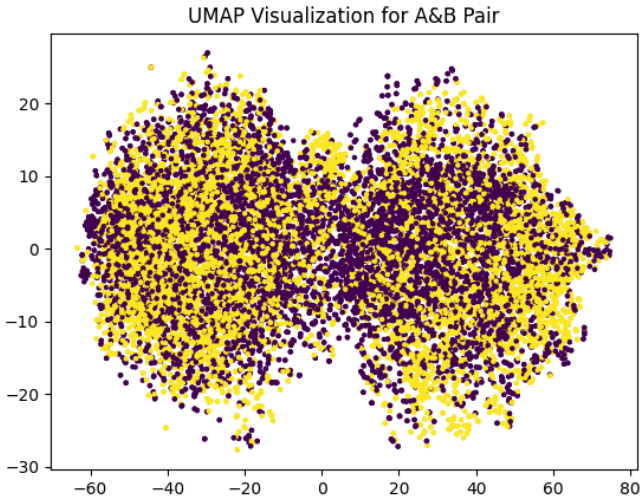
On Word2Vec Lemm(Without SW):

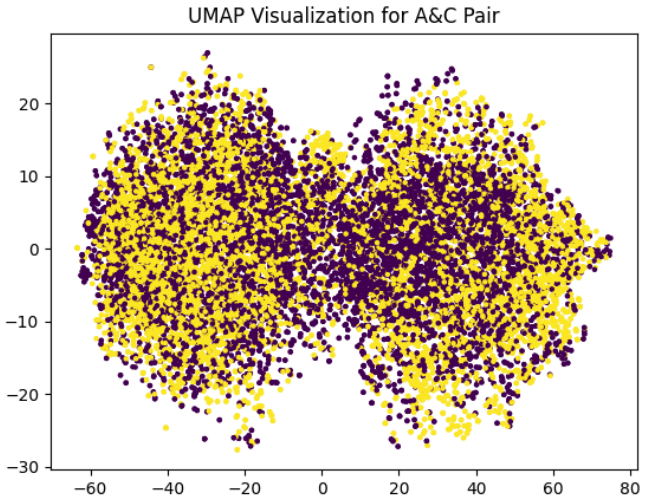


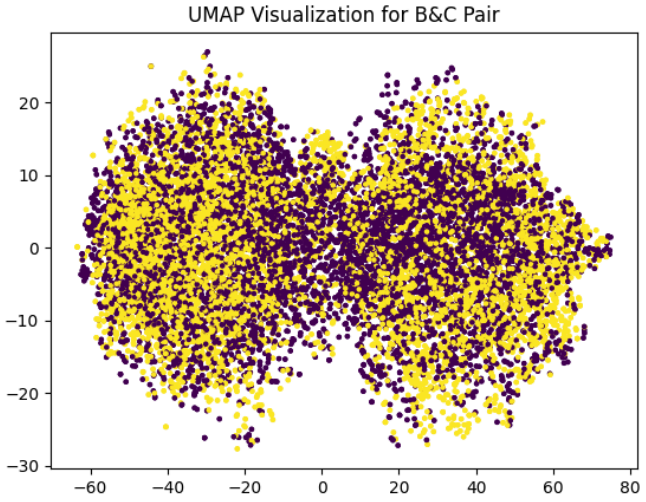




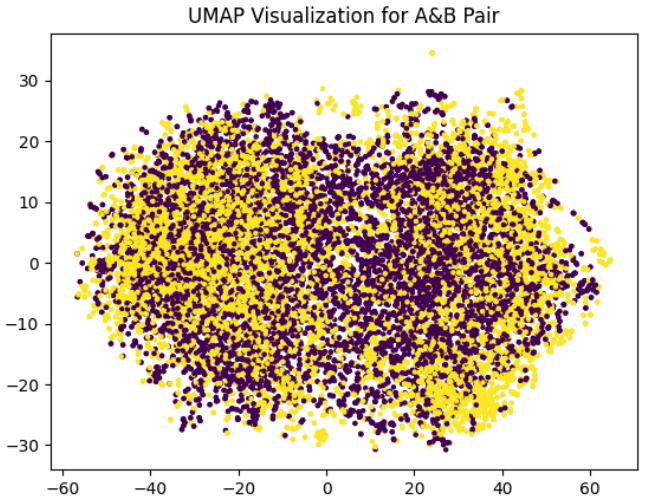
On Word2Vec Words(With SW):

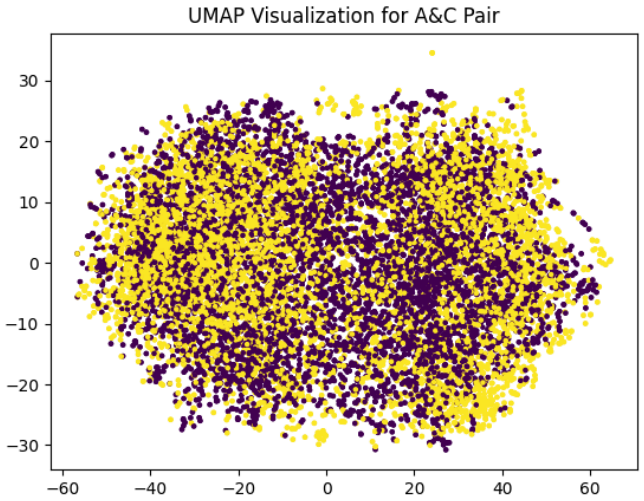


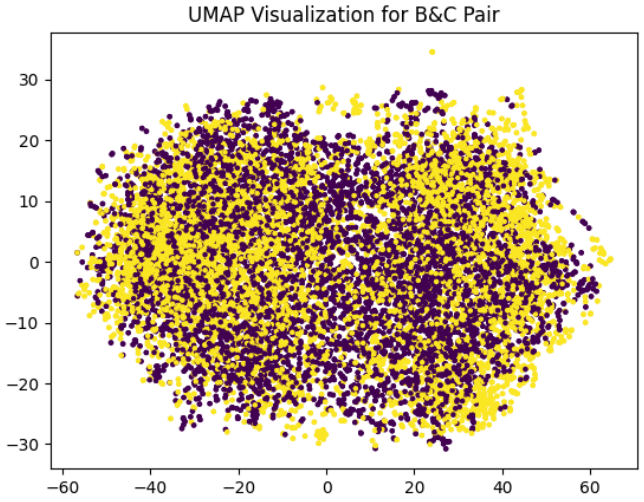




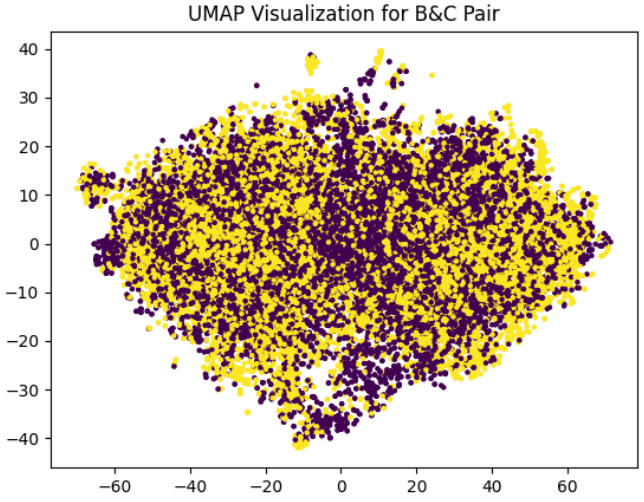
On Word2Vec Words(Without SW):

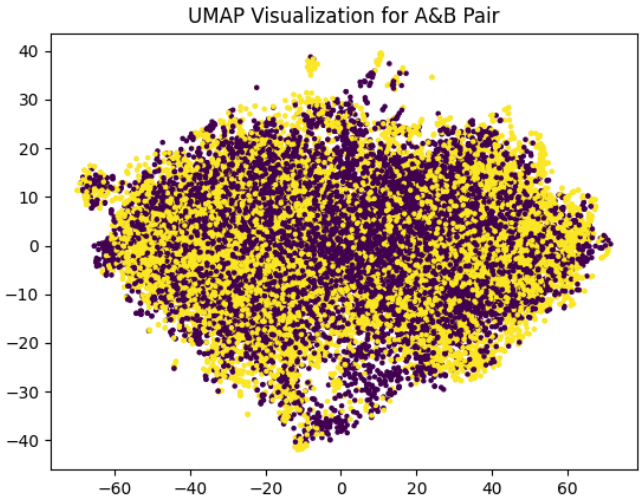


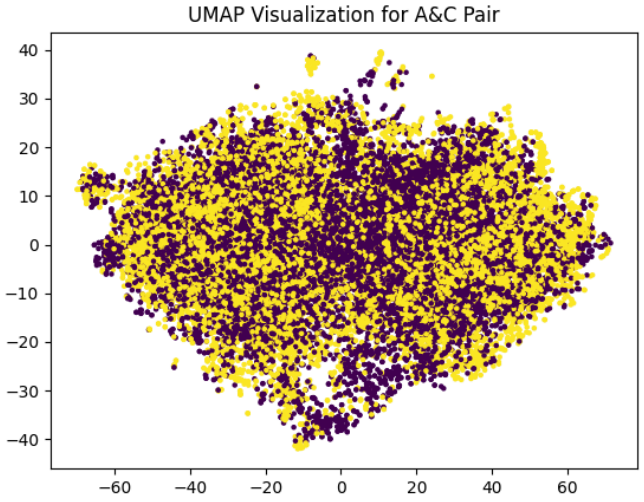

UMAP Visualization for A&C Pair
30
20
10
0
−10
−20
−30
−60
−40
−20
0
20
40
60


UMAP Visualization for B&C Pair
30
20
10
0
−10
−20
−30
−60
−40
−20
0
20
40
60

On Bert:

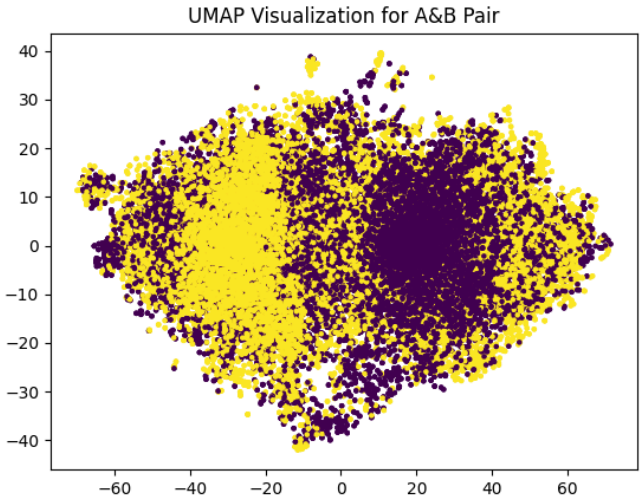

UMAP Visualization for B&C Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

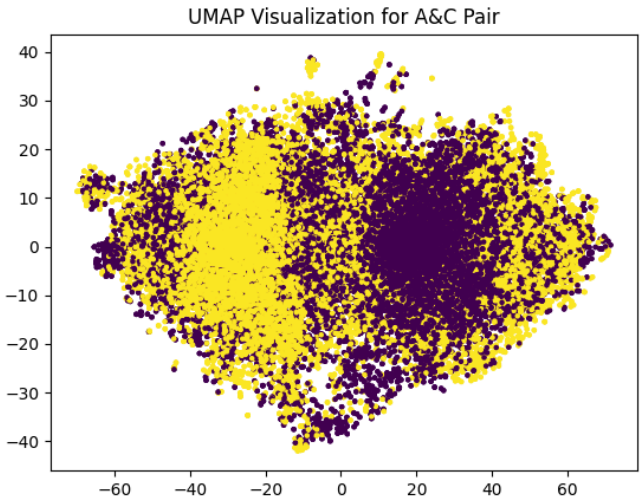

UMAP Visualization for A&B Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

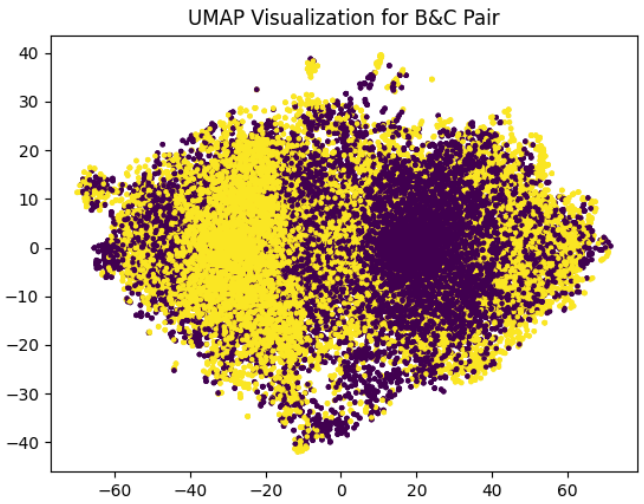

UMAP Visualization for A&C Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

LoR:

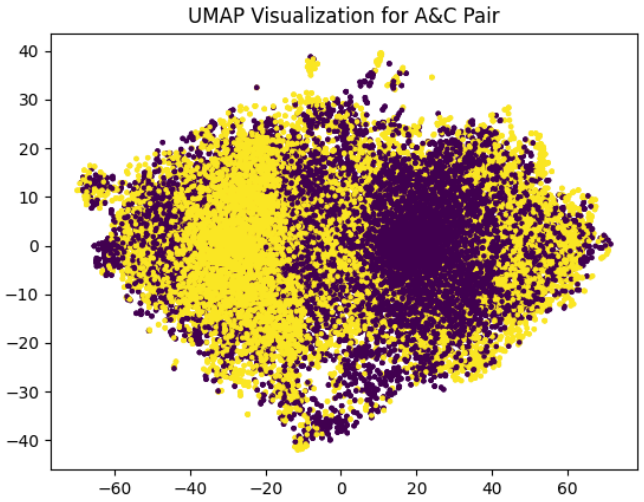
On TF-IDF Words:

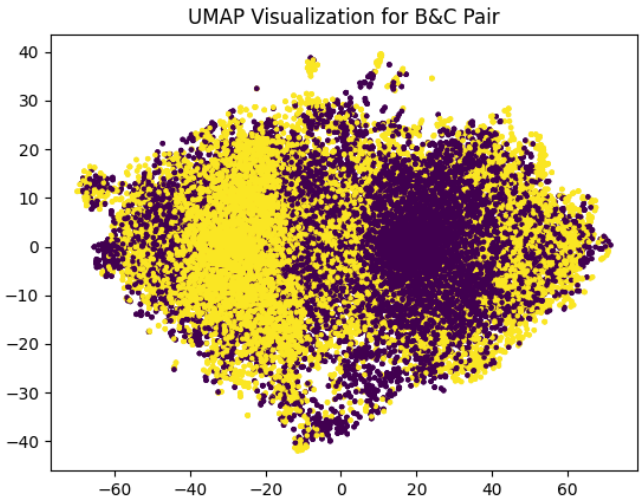


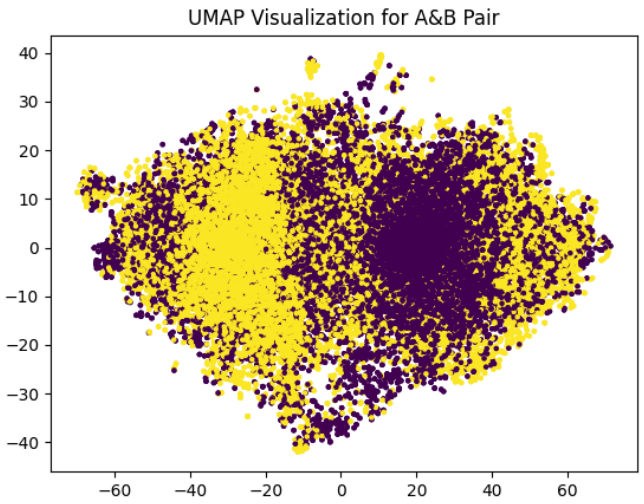




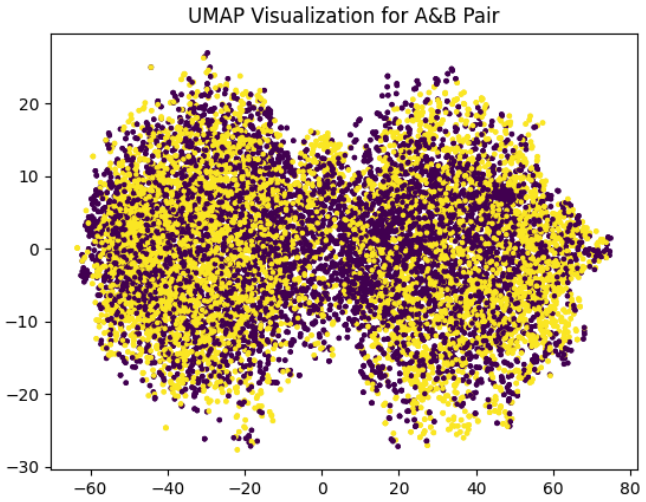
On TF-IDF Lemm:

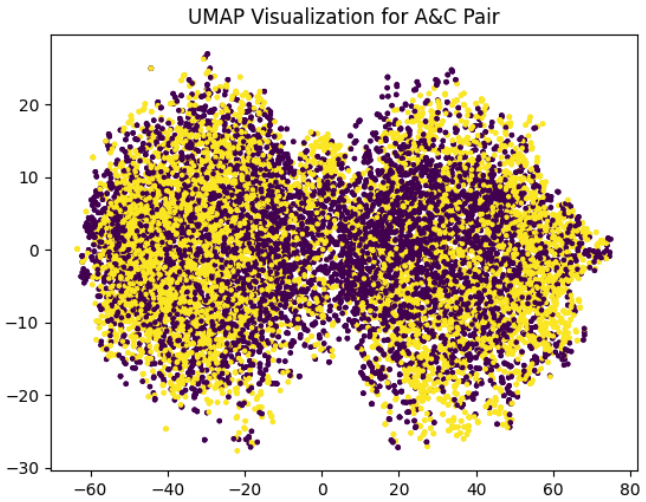


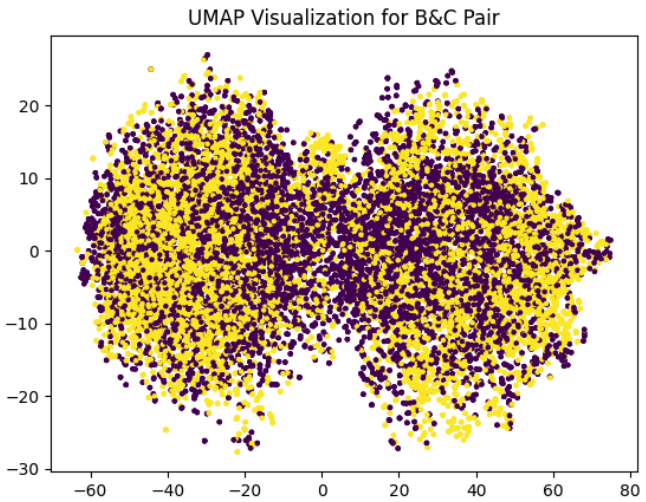




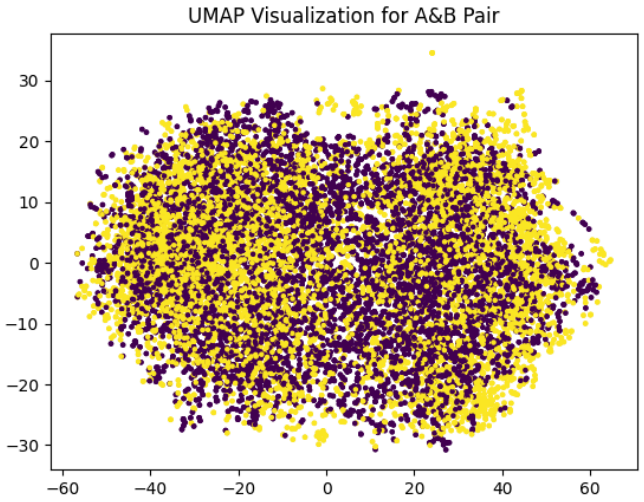
On Word2Vec Lemm(With SW):

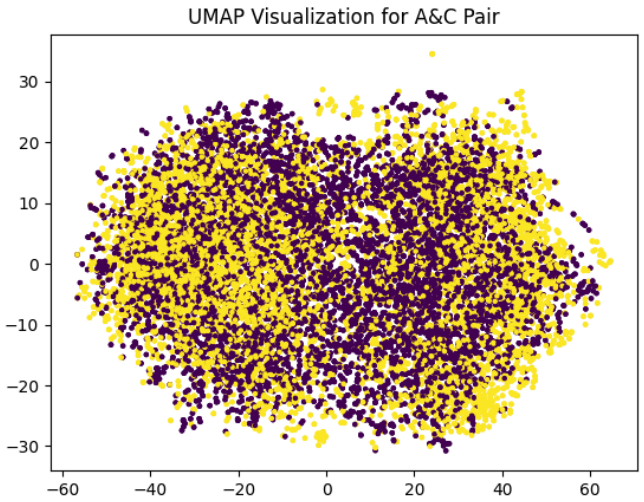


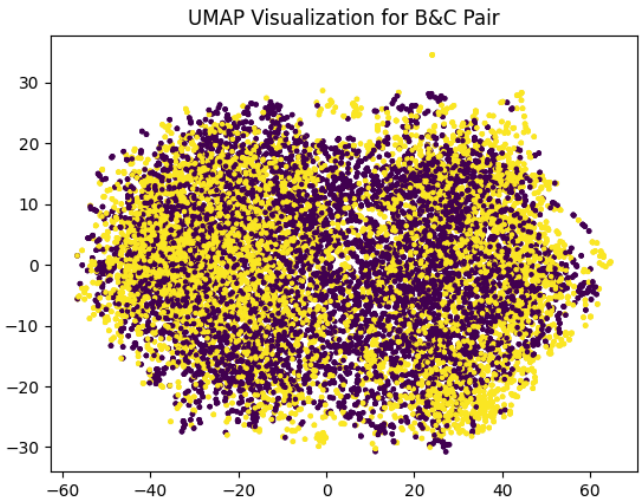




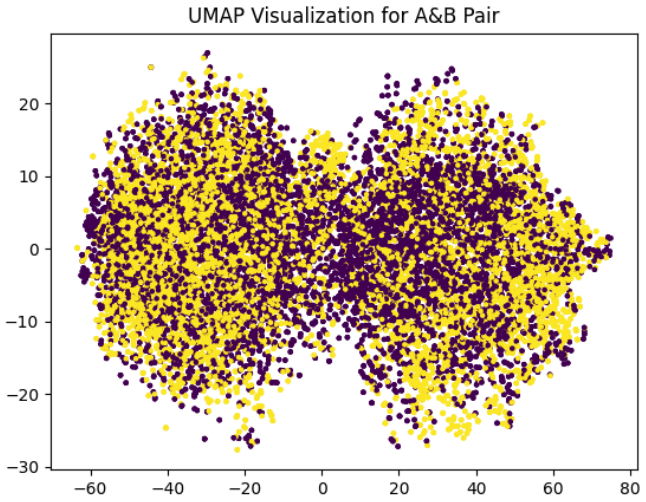
On Word2Vec Lemm(Without SW):

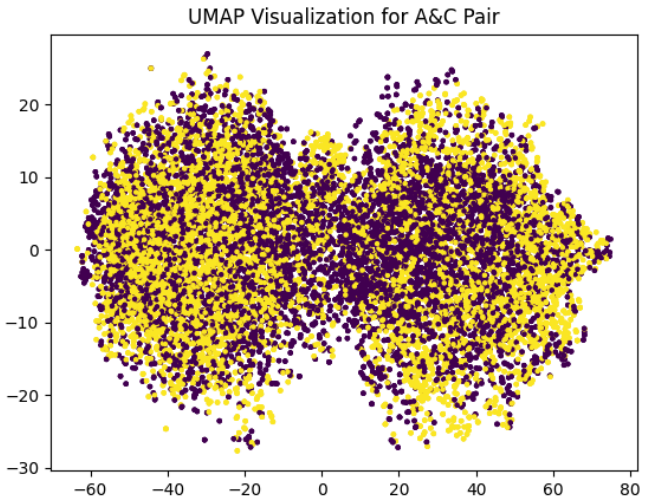


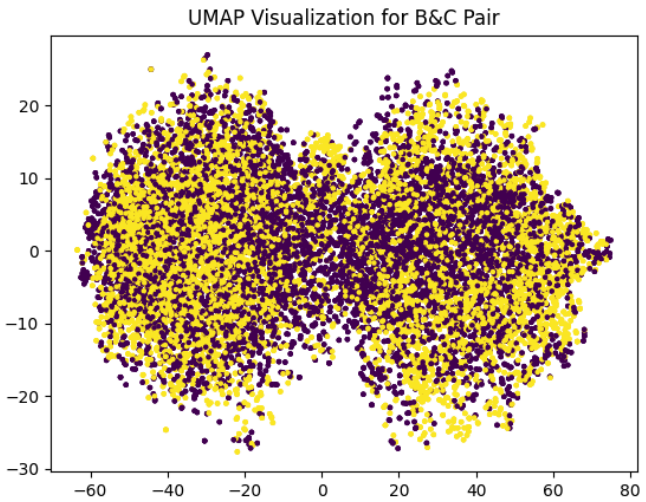




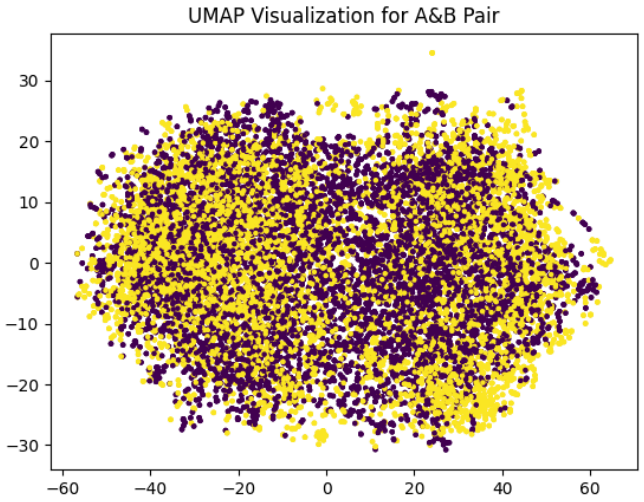
On Word2Vec Words(With SW):

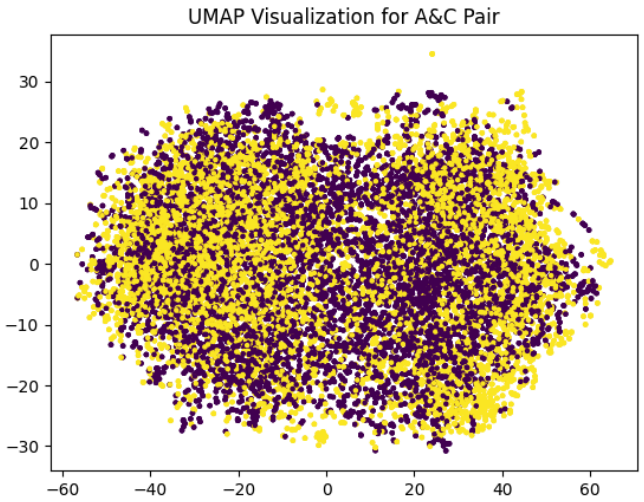


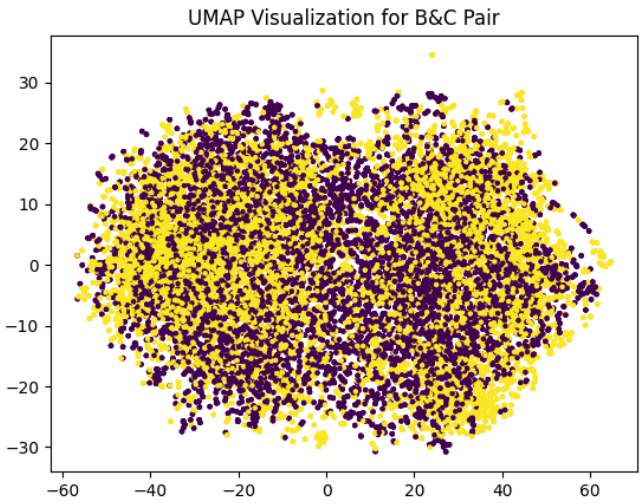




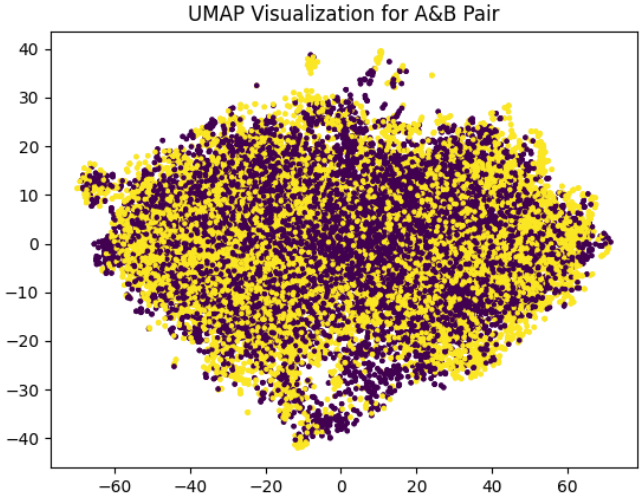
On Word2Vec Words(Without SW):

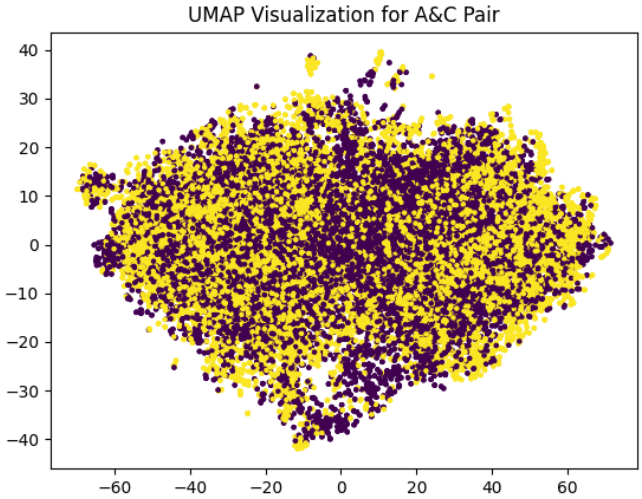


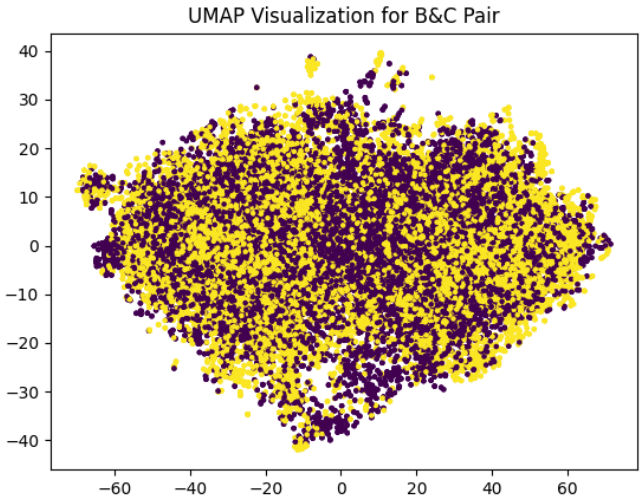




On Bert:

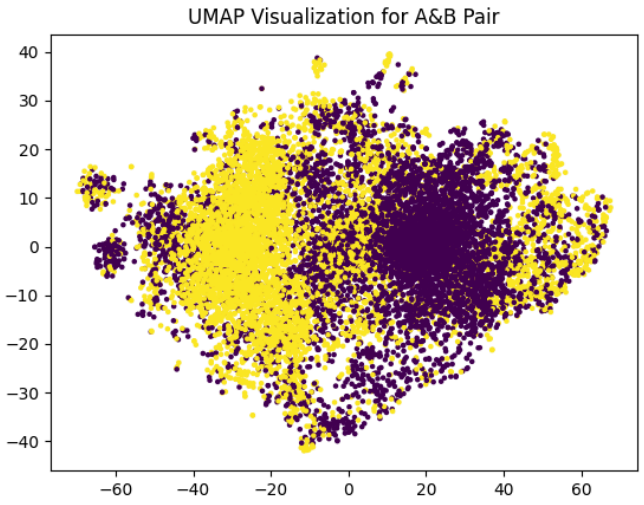

UMAP Visualization for A&B Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

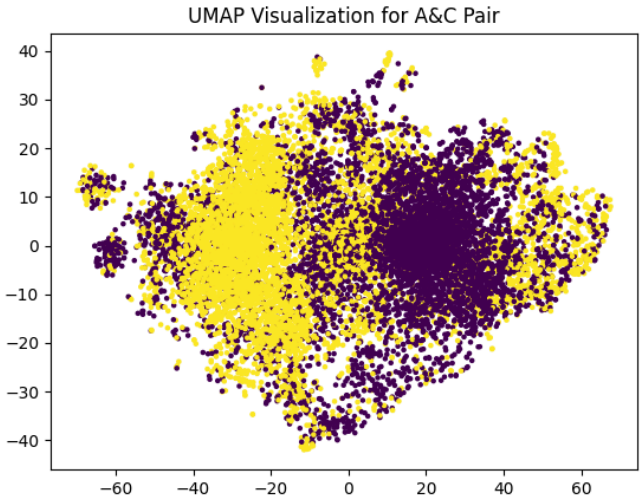

UMAP Visualization for A&C Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

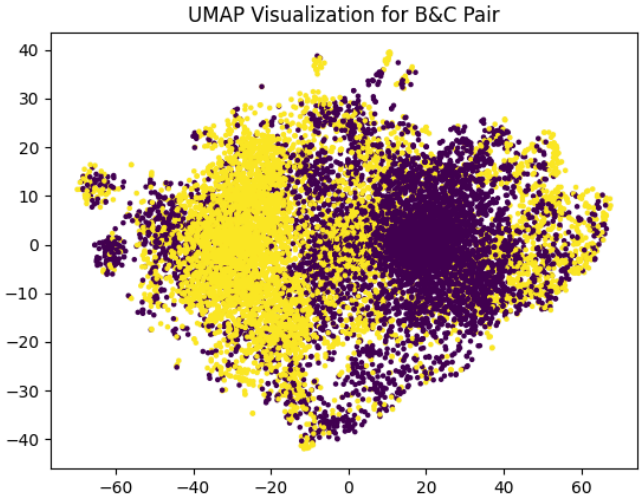

UMAP Visualization for B&C Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

ANN-relu:

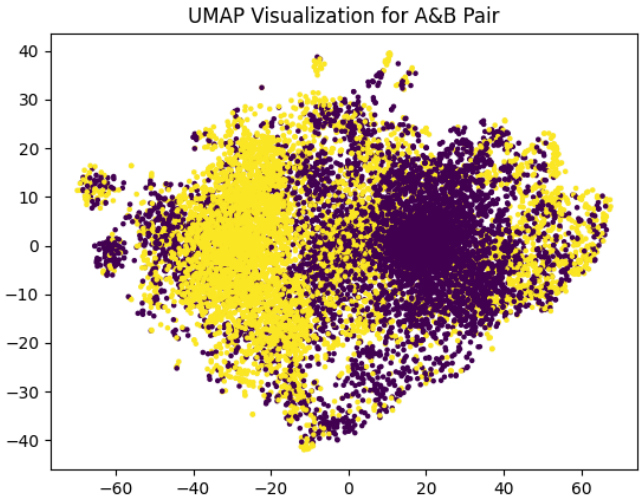
On TF-IDF Words:

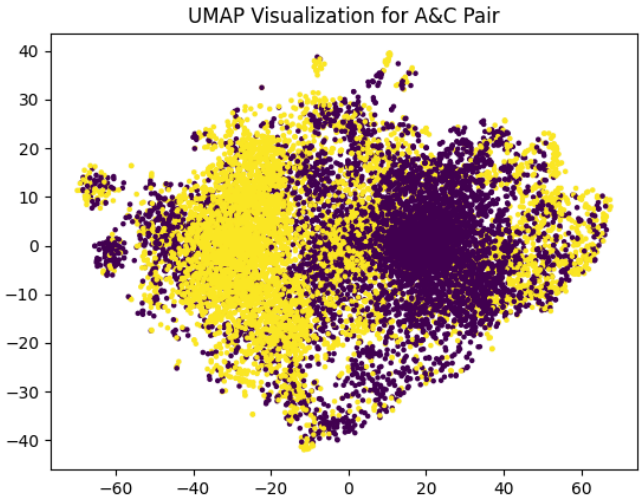


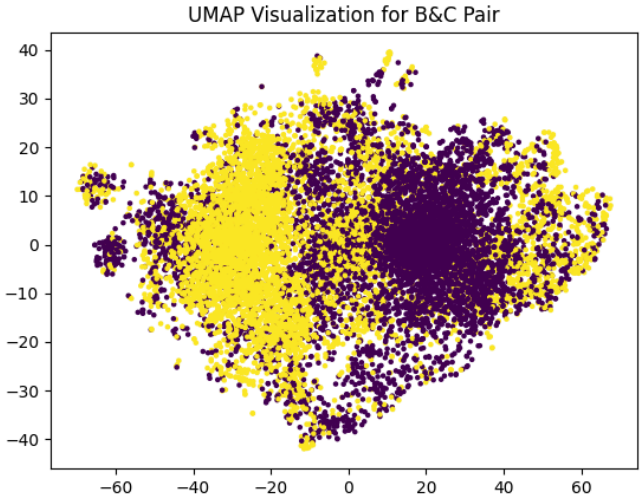




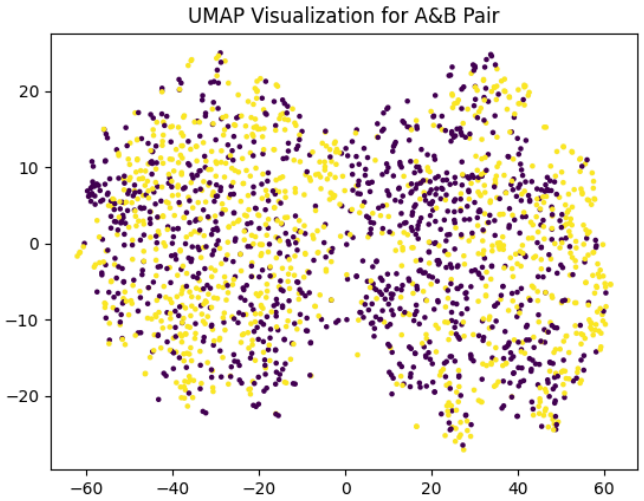
On TF-IDF Lemm:

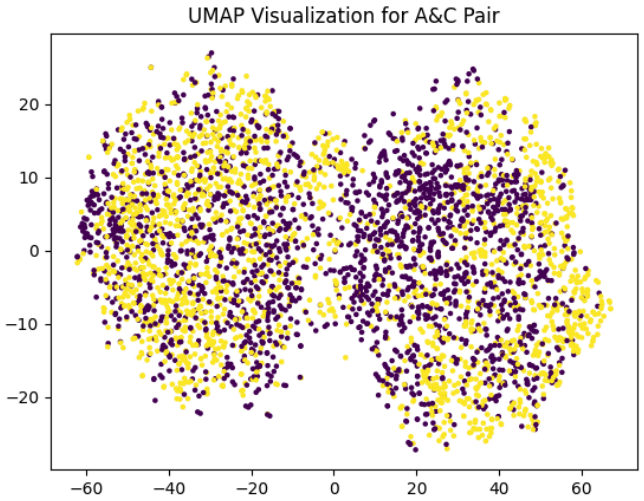


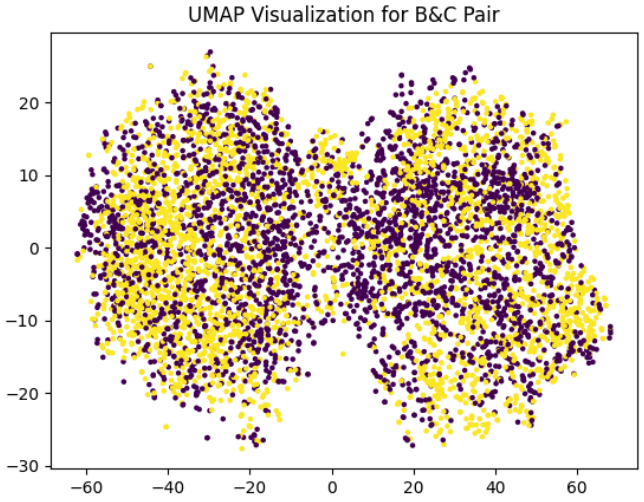




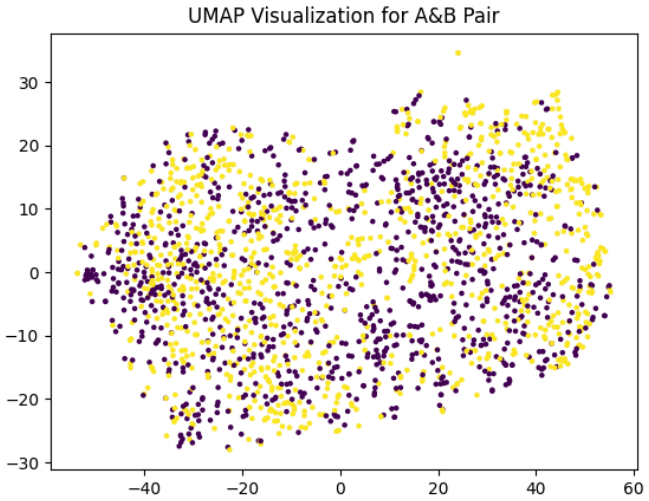
On Word2Vec Lemm(With SW):

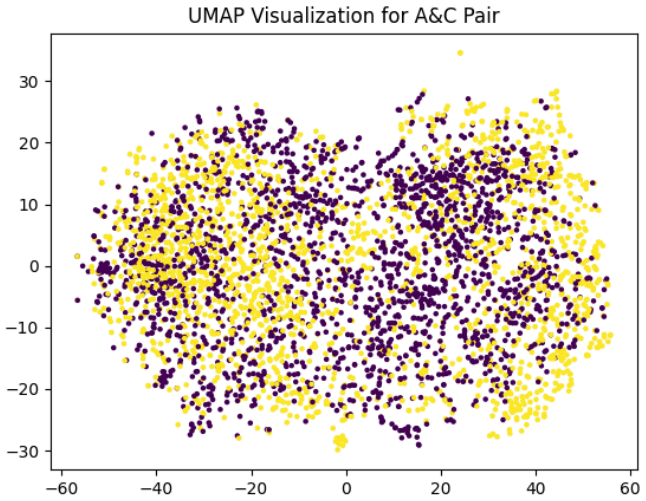


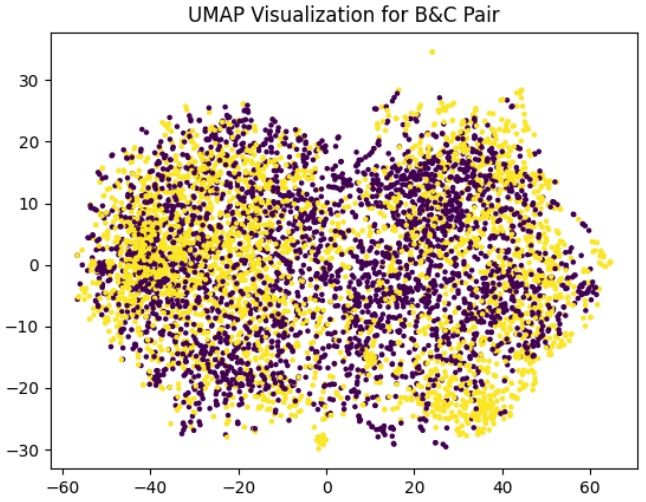




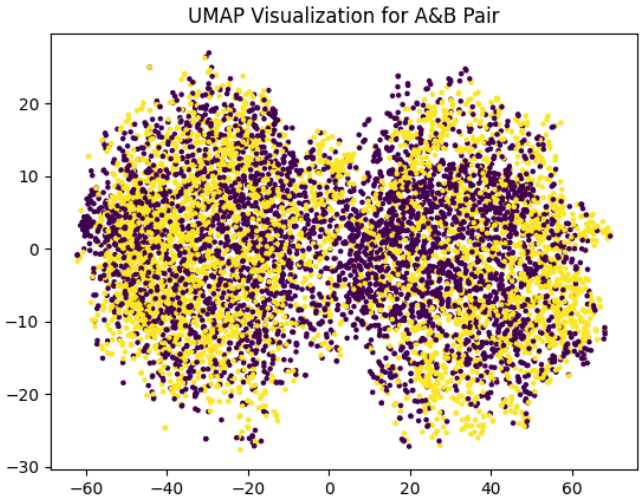
On Word2Vec Lemm(Without SW):

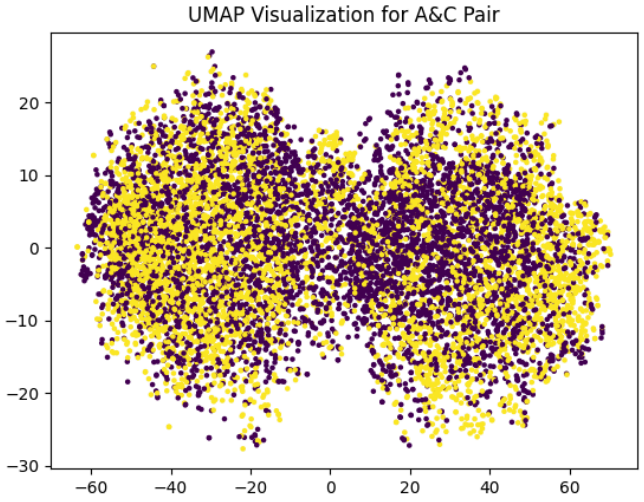


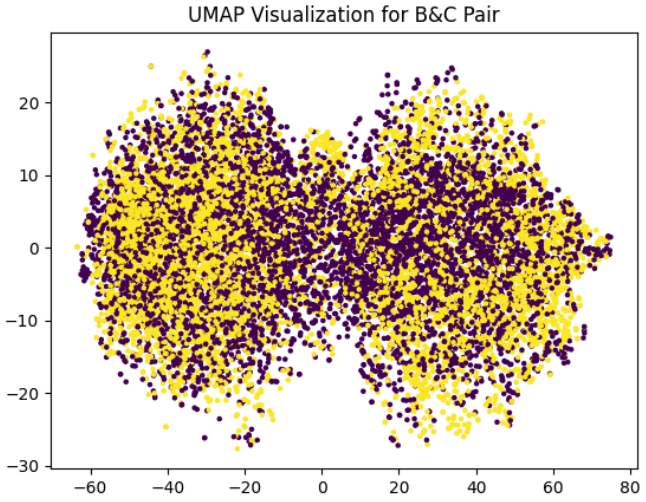




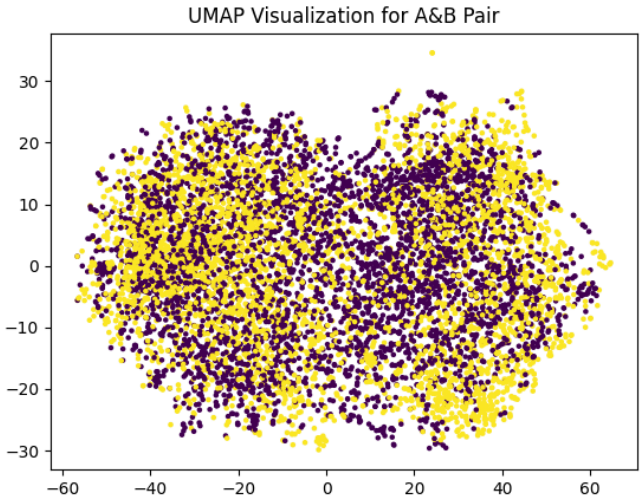
On Word2Vec Words(With SW):

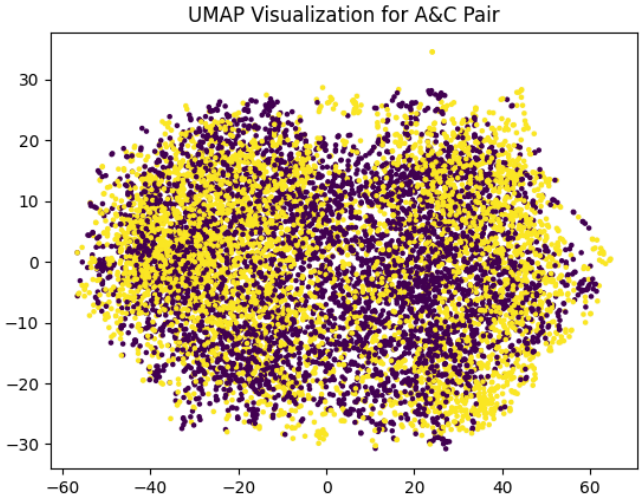


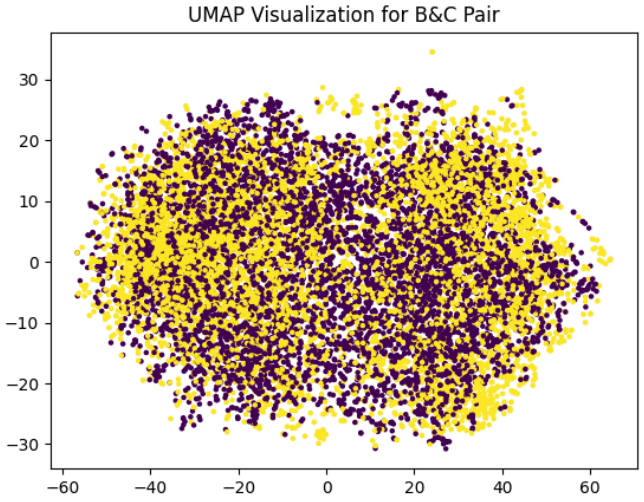




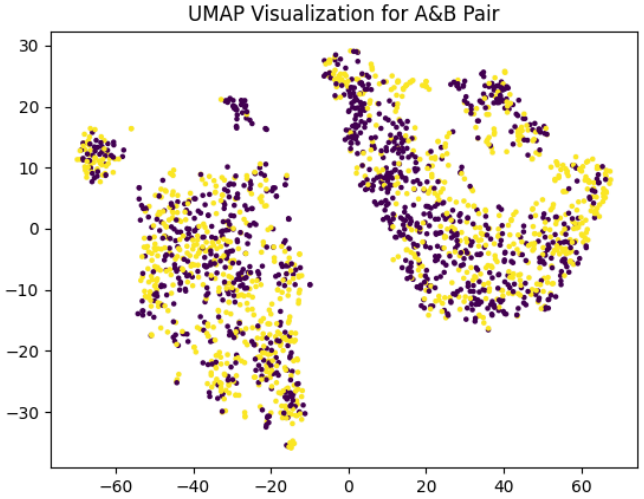
On Word2Vec Words(Without SW):

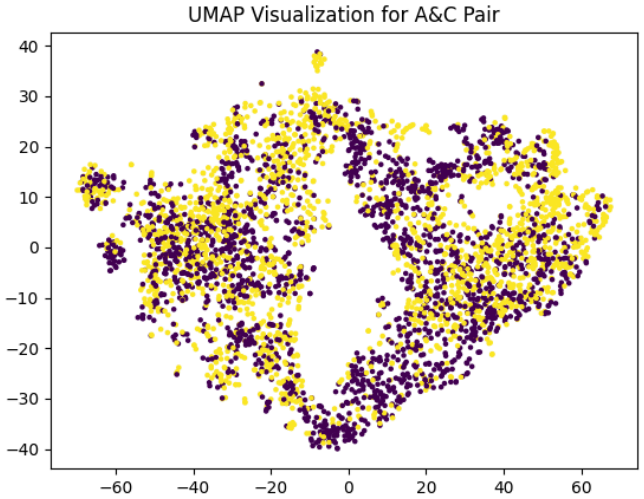

UMAP Visualization for A&B Pair
30
20
10
0
−10
−20
−30
−60
−40
−20
0
20
40
60


UMAP Visualization for A&C Pair
30
20
10
0
−10
−20
−30
−60
−40
−20
0
20
40
60


UMAP Visualization for B&C Pair
30
20
10
0
−10
−20
−30
−60
−40
−20
0
20
40
60

On Bert:


UMAP Visualization for A&B Pair
30
20
10
0
−10
−20
−30
−60
−40
−20
0
20
40
60


UMAP Visualization for A&C Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

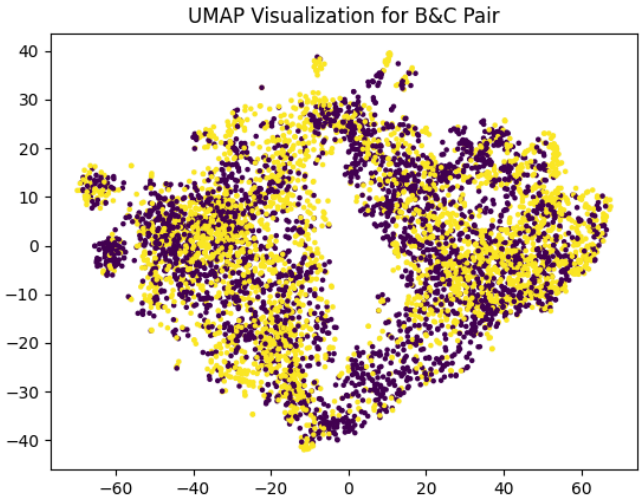

UMAP Visualization for B&C Pair
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60

Table of the results

			A Vs' B		B Vs' C		A Vs' C	
	matrix	**method**	**F1**	**ACCU**	**F1**	**ACCU**	**F1**	**ACCU**
מידה לא פוקחת	**TF-IDF Words**	**K-means**	0.66	0.5	0.59	0.49	0.65	0.55
		DBSCAN	0.67	0.5	0.62	0.55	0.65	0.58
		GMM	0.66	0.5	0.59	0.48	0.64	0.55
	TF-IDF Lemm	**K-means**	0.6	0.5	0.54	0.51	0.64	0.55
		DBSCAN	0.59	0.54	0.54	0.52	0.6	0.56
		GMM	0.6	0.5	0.03	0.43	0.64	0.55
	Word2Vec Lemm (with SW)	**K-means**	0.55	0.52	0.58	0.5	0.56	0.48
		DBSCAN	0.47	0.51	0.45	0.5	0.48	0.51
		GMM	0.53	0.55	0.52	0.5	0.55	0.52
	Word2Vec Lemm (without SW)	**K-means**	0.60	0.53	0.59	0.50	0.60	0.52
		DBSCAN	0.54	0.52	0.50	0.51	0.54	0.53
		GMM	0.55	0.55	0.52	0.5	0.55	0.56
	Word2Vec words (with SW)	**K-means**	0.58	0.47	0.60	0.5	0.58	0.48
		DBSCAN	0.41	0.51	0.41	0.5	0.42	0.51
		GMM	0.54	0.52	0.51	0.5	0.55	0.52
	Word2Vec words (without SW)	**K-means**	0.60	0.52	0.6	0.5	0.62	0.53
		DBSCAN	0.44	0.51	0.42	0.50	0.43	0.49
		GMM	0.54	0.55	0.51	0.5	0.55	0.56
	BERT	**K-means**	0.54	0.5	0.57	0.52	0.57	0.52
		DBSCAN	0.57	0.55	0.49	0.50	0.57	0.56
		GMM	0.5	0.49	0.5	0.49	0.5	0.49

			A Vs' B		B Vs' C		A Vs' C	
	matrix	**method**	**F1**	**ACCU**	**F1**	**ACCU**	**F1**	**ACCU**
למידה מפוקחת	**TF-IDF Words**	**NB**	0.82	0.84	0.85	0.85	0.91	0.91
		LoR	0.88	0.88	0.88	0.88	0.94	0.94
		SVM	NA	NA	NA	NA	NA	NA
		ANN-relu	0.87	0.87	0.92	0.92	0.87	0.87
		ANN-GELU	NA	NA	NA	NA	NA	NA
	TF-IDF Lemm	**NB**	0.73	0.76	0.83	0.82	0.84	0.85
		LoR	0.87	0.87	0.89	0.89	0.94	0.94
		SVM	NA	NA	NA	NA	NA	NA
		ANN-relu	0.87	0.87	0.88	0.88	0.92	0.92
		ANN-GELU	NA	NA	NA	NA	NA	NA
	Word2Vec Lemm (with SW)	**NB**	0.36	0.55	0.63	0.66	0.38	0.56
		LoR	0.67	0.69	0.7	0.68	0.75	0.75
		SVM	NA	NA	NA	NA	NA	NA
		ANN-relu	0.66	0.68	0.67	0.64	0.66	0.5
		ANN-GELU	NA	NA	NA	NA	NA	NA
	Word2Vec Lemm (without SW)	**NB**	0.38	0.56	0.39	0.59	0.41	0.57
		LoR	0.70	0.7	0.71	0.71	0.74	0.74
		SVM						
		ANN-relu	0.7	0.69	0.65	0.52	0.64	0.55

		ANN-GELU	NA	NA	NA	NA	NA	NA
	Word2Vec words (with SW)	**NB**	0.35	0.55	0.63	0.56	0.35	0.55
		LoR	0.66	0.67	0.68	0.67	0.71	0.71
		SVM	NA	NA	NA	NA	NA	NA
		ANN-relu	0.70	0.66	0.66	0.5	0.66	0.5
		ANN-GELU	NA	NA	NA	NA	NA	NA
	Word2Vec words (without SW)	**NB**	0.38	0.56	0.65	0.57	0.39	0.57
		LoR	0.67	0.67	0.70	0.69	0.73	0.73
		SVM	NA	NA	NA	NA	NA	NA
		ANN-relu	0.68	0.56	0.66	0.52	0.66	0.52
		ANN-GELU	NA	NA	NA	NA	NA	NA
	BERT	**NB**	0.34	0.55	0.44	0.55	0.42	0.58
		LoR	0.74	0.74	0.8	0.8	0.87	0.87
		SVM	NA	NA	NA	NA	NA	NA
		ANN-relu	0.66	0.49	0.75	0.71	0.66	0.49
		ANN-GELU	NA	NA	NA	NA	NA	NA

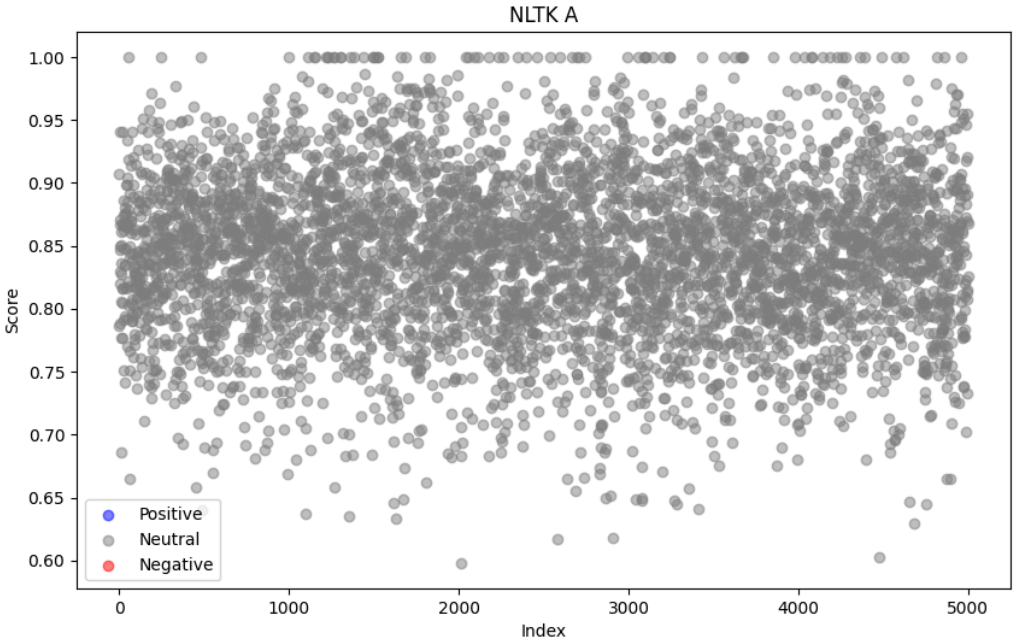
The results seem coherent for the majority, since from the majority, the results obtained without stop words are better, there is coherence between the results of the supervised algorithms. There is coherence between the unsupervised algorithms . As expected the supervised algorithms have better results than the unsupervised ones.

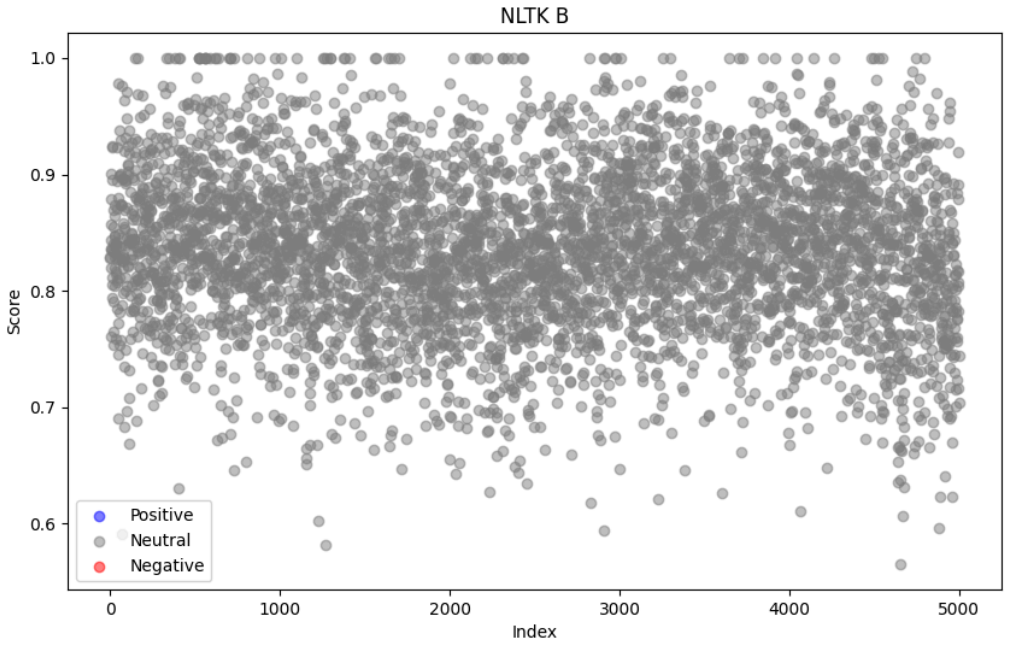
Sentiment Analysis

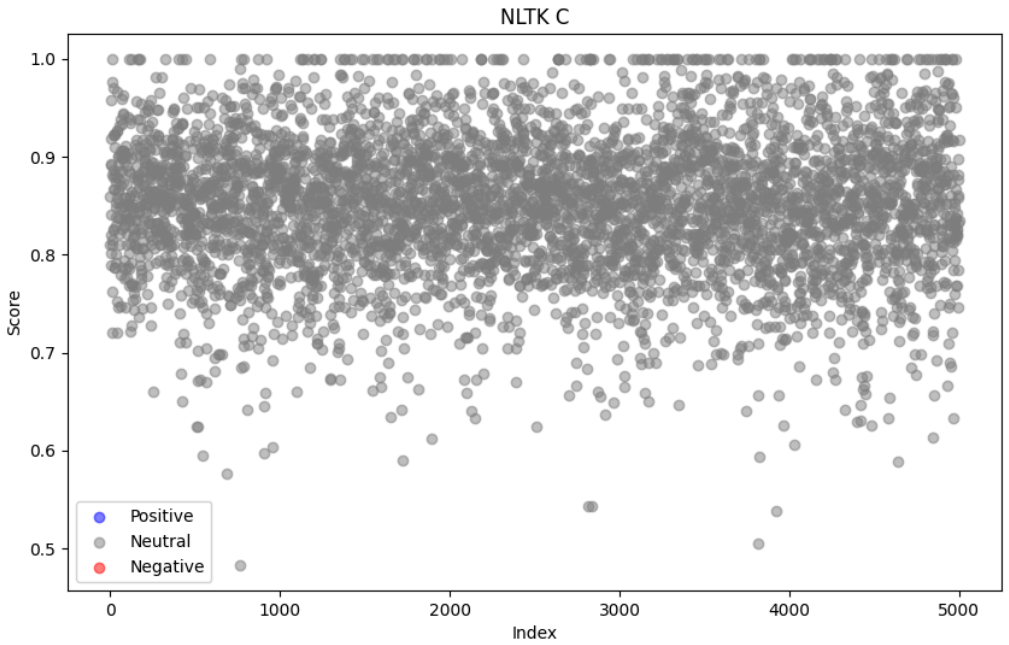
The global results of each sentiment analysis model used are listed in the Result Table for clarity. Each model result has its own excel file attached to this file.

NLTK sentiment VADER

In the first step of sentiment analysis we translated the 15000 Hebrew texts to english in order to use the nltk model to try sentiment analysis.
It didn’t go as we expected because we got 100 % neutral on the 3 text groups A,B and C. We think that the fact that these texts are for a great part articles in the topic of Israel politic, there are a lot of names and words that are positive or negative in Hebrew that would be depicted here as neutral. We finally got a neutral result.
For example while 'חמאס' is 96% negative in Hebert it's 100% neutral on nltk


NLTK A
Score
Index
Positive
Neutral
Negative
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
1000
2000
3000
4000
5000


NLTK B
Score
Index
Positive
Neutral
Negative
1.0
0.9
0.8
0.7
0.6
0
1000
2000
3000
4000
5000

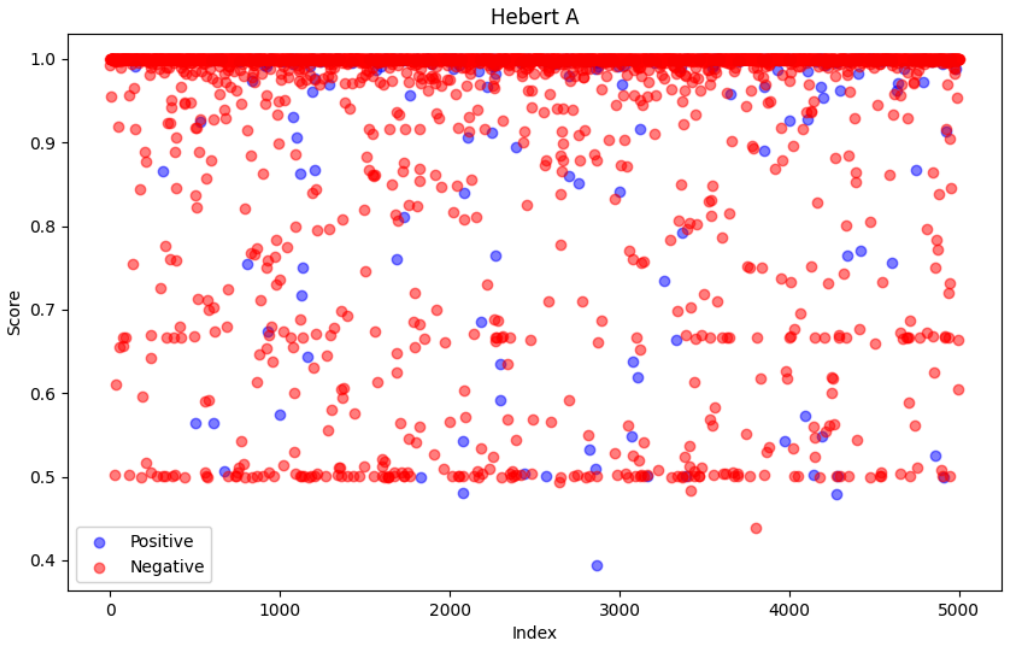


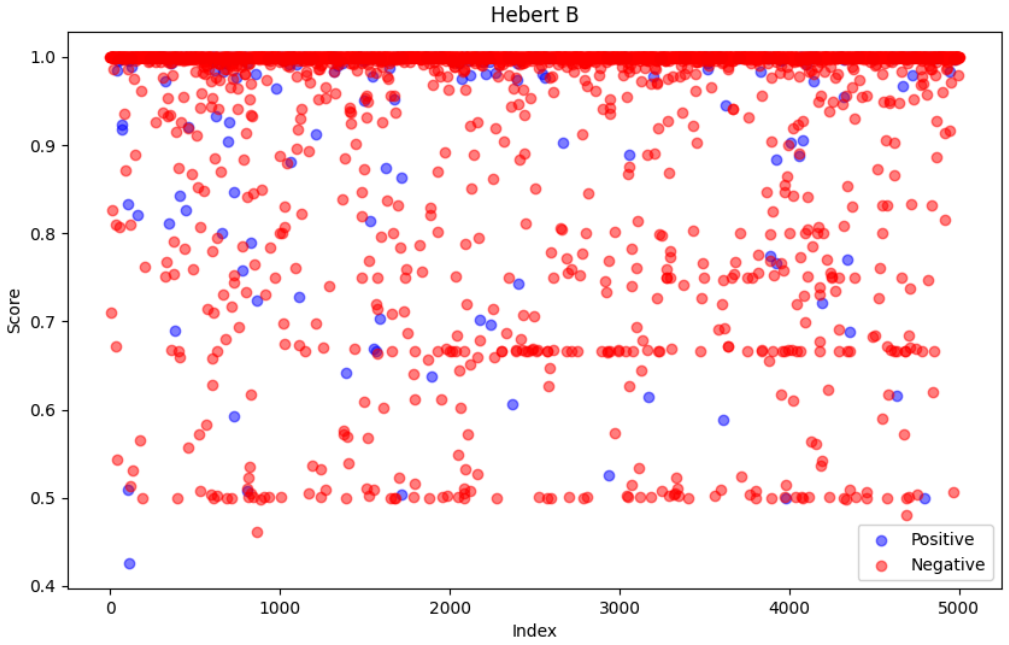
AlephbertGuimmel NER

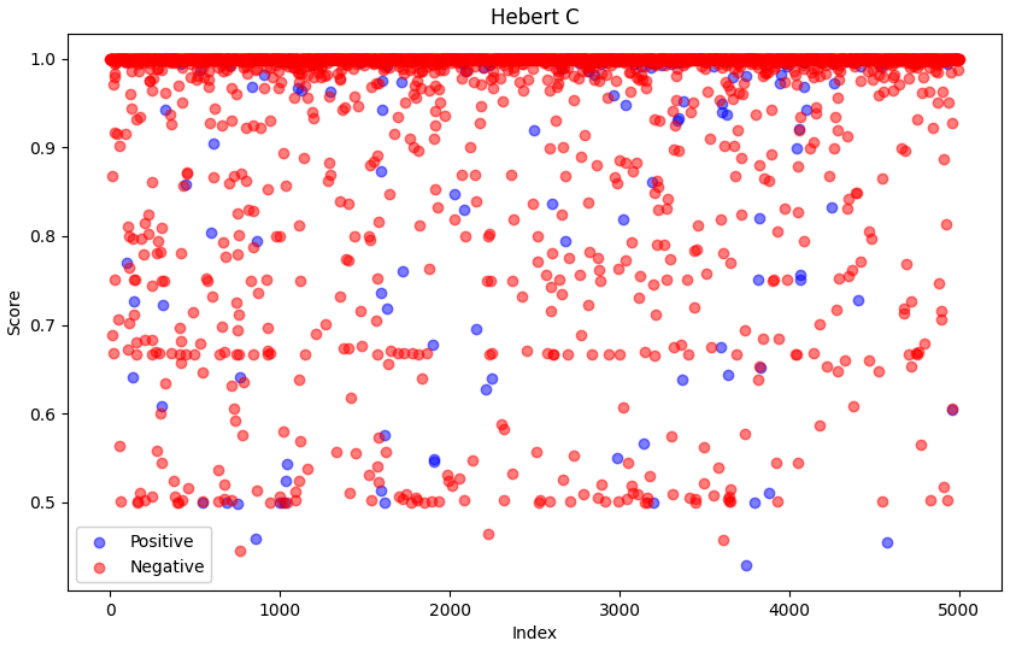
As a second step we took our 15000 texts and processed them with the name entity recognition algorithm from AlephBertGuimmel in order to create a new file which contains the 15000 texts but in this file, each name recognized by the algorithm is changed with a random first name and a random last name. The goal of the manipulation is to check in the next steps if the names have an influence on the sentiment of the texts, if yes to which degree.

HeBert

As a third step we processed our files with the Hebert model using the pipeline shown in their github. The results we got were highly negative for the three groups which is really different from the nltk model after translation, which was 100 % neutral.
We then used the model on the files with names changed, the results were still highly negative, a bit less than without the names changed but not an impactful difference. (See the result table)

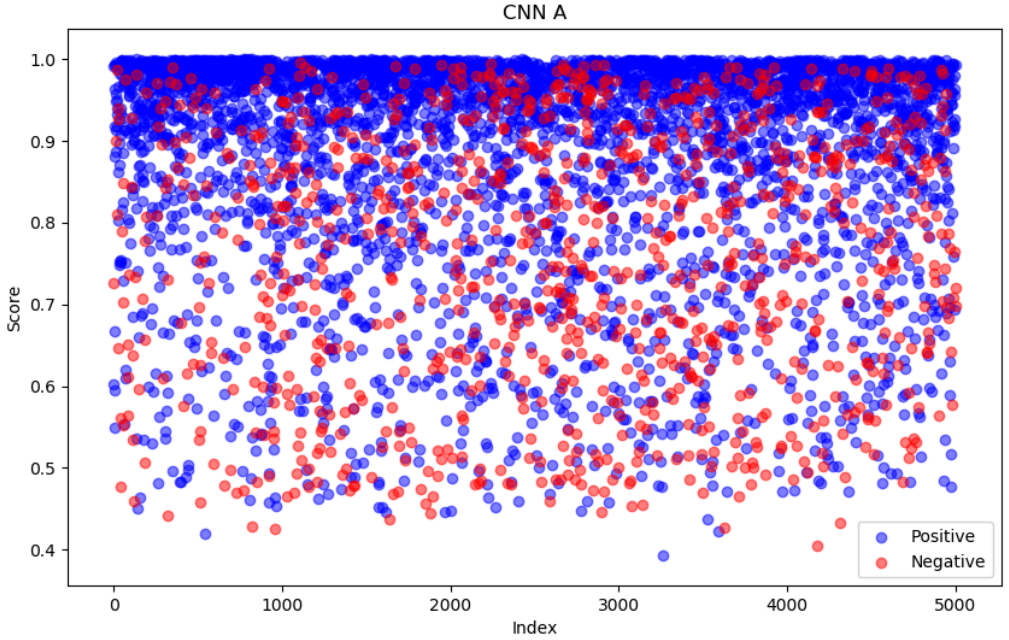

Hebert A
Score
Index
Positive
Negative
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000

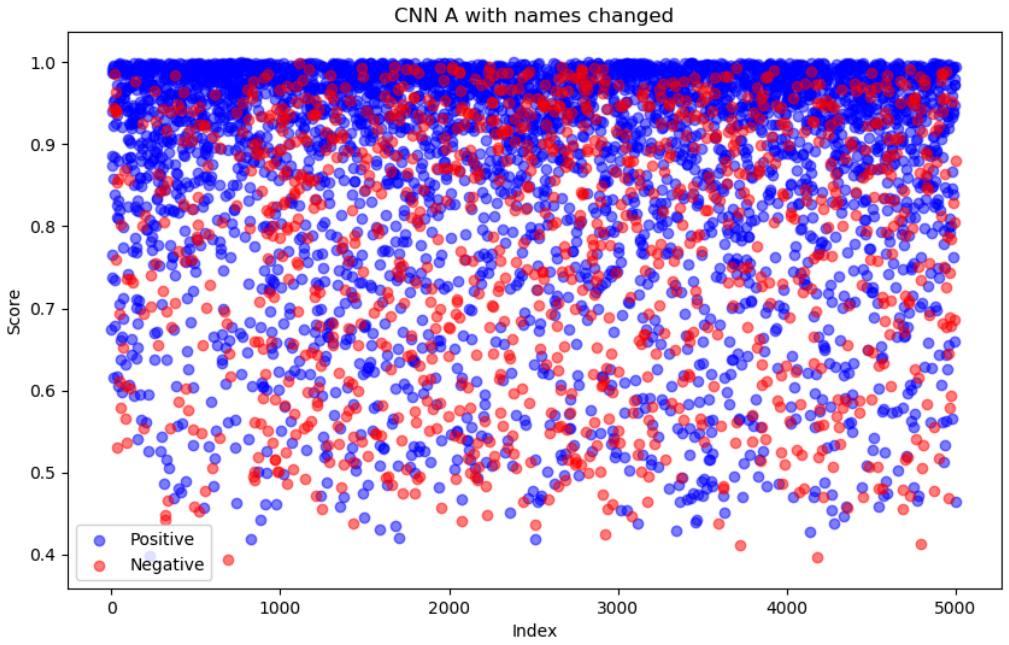

Hebert B
Score
Index
Positive
Negative
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000

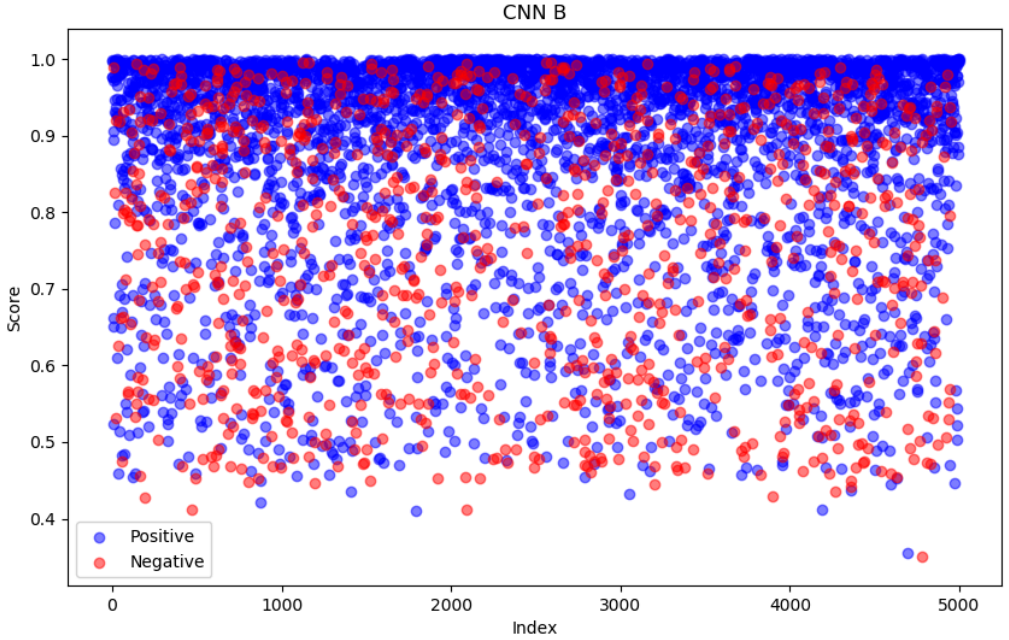


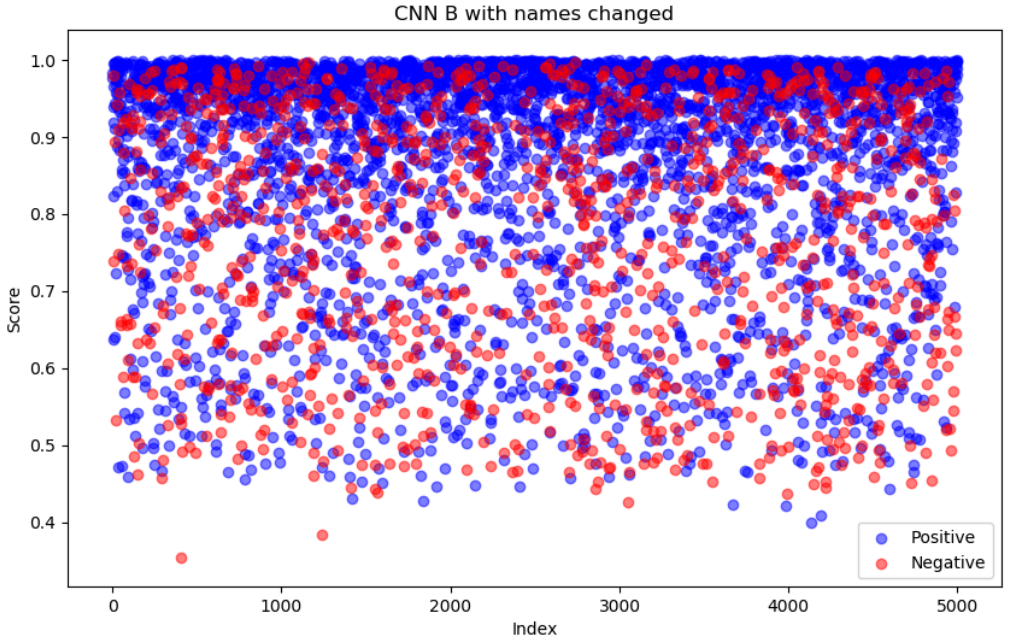
CNN-Neural sentiments

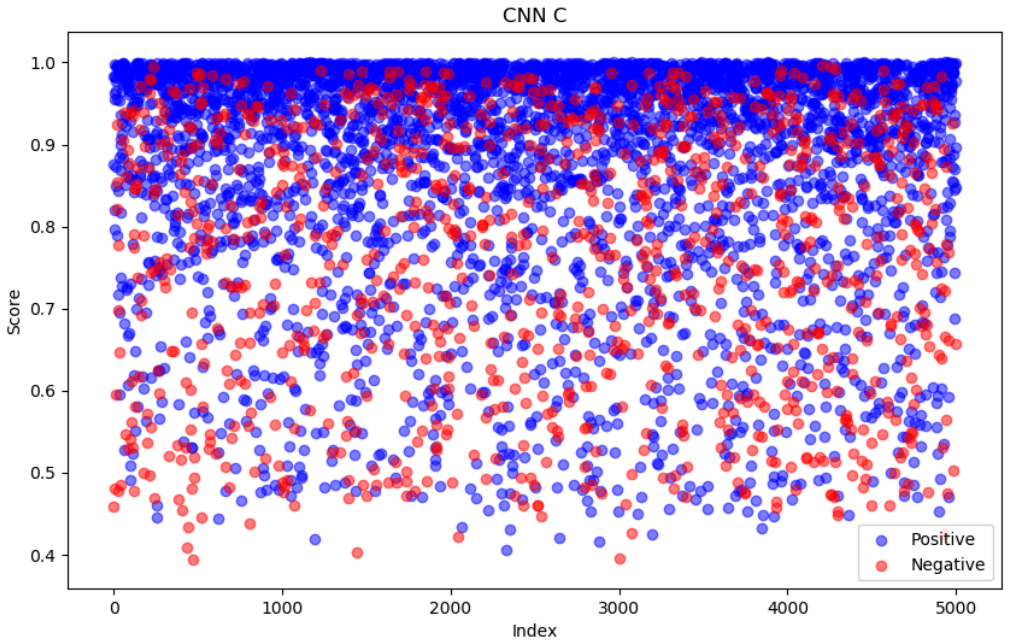
In the last step we used an available model hebrew sentiment analysis model using convolutional neural network, we trained the network with the given training data and then used the model on both: original file and names changed file.
The result is still highly negative but less negative than when using the Hebert model.
Note on the model:
The model is trained with 10 000 lines only, that are probably resulting from facebook comments,the training on the model seems to be mainly focus on positives and negatives probabilities (there is very few '2' lines in the training
wich correspong to neutral), we did some test and the model really struggles in neutrals sentances, resulting in scores around 45/45/10 pos/neg/neu.

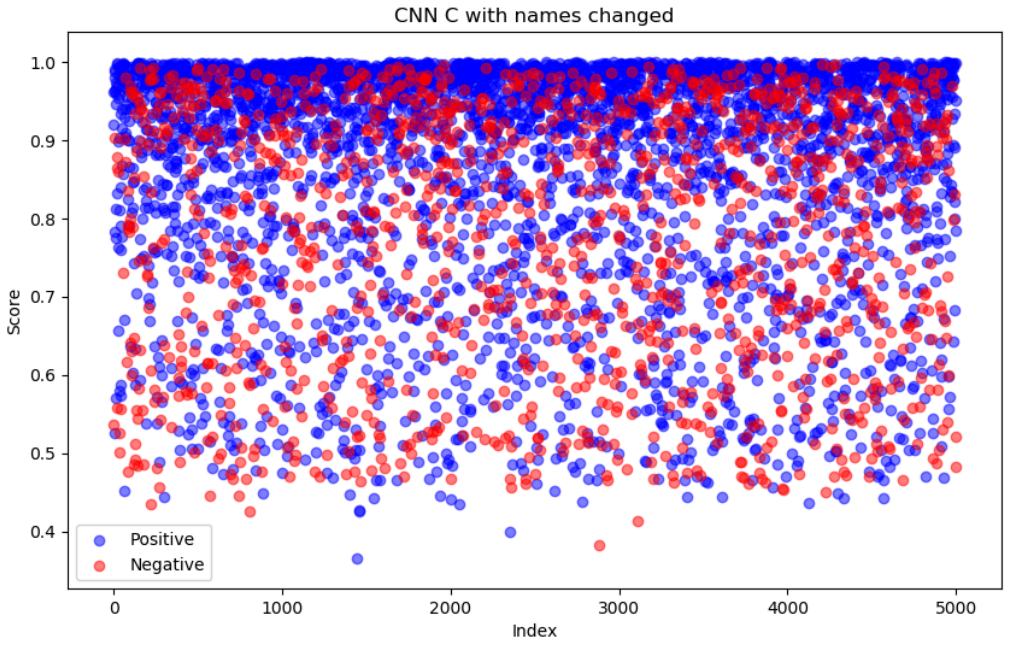

CNN A
Score
Index
Positive
Negative
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000


CNN A with names changed
Score
Index
Positive
Negative
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000


CNN B
Score
Index
Positive
Negative
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000


CNN B with names changed
Score
Index
Positive
Negative
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000





Sentiment analysis conclusion

It seems that after processing the texts through different models it seems that the sentiments of the text are with a high majority of negatives. We can also conclude that doing sentiment analysis from Hebrew to English translated texts, using English sentiment analysis models seems not appropriate. It would be interesting to train a model with Hebrew texts translated to English to compare the results with the Hebrew models.

Links: Hebert:https://github.com/avichaychriqui/HeBERT/tree/main
CNN: https://github.com/omilab/Neural-Sentiment-Analyzer-for-Modern-Hebrew
AlephbertGuimmel Ner:https://huggingface.co/dicta-il/dictabert-ner

			A			B			C		
			חיובי %	שלילי %	נייטרל %	חיובי %	שלילי %	נייטר לי %	חיובי %	שלילי %	נייטרלי %
סנטימנט		תרגום	0	0	100	0	0	100	0	0	100
	עם שמות מקוריים	HeBERT	4.54	91.72	3.74	4.20	92.34	3.46	4.66	90.64	4.70
		CNN	19.30	80.68	0.02	19.46	80.52	0.02	20.02	79.98	0.00
	עם החלפת שמות	HeBERT	4.54	91.04	4.42	4.38	91.90	3.72	4.68	90.08	5.24
		CNN	22.28	77.68	0.04	22.88	77.08	0.04	25.52	74.44	0.04

Insights and Analysis

Unsupervised vs. Supervised Learning

The exercises demonstrated the strengths and limitations of both unsupervised and supervised learning models in text analysis. Unsupervised learning models, including K-means, DBSCAN, and GMM, were adept at discovering hidden patterns and grouping texts based on thematic or stylistic similarities without prior labeling. However, these models generally yielded lower accuracy and F1 scores compared to supervised models. This discrepancy underscores the importance of labeled data in achieving higher precision in classification

tasks. Supervised models, such as Naive Bayes, Logistic Regression, and Artificial Neural Networks, showed superior performance, particularly in distinguishing between the different categories of texts. The difference in performance can be attributed to the supervised models' ability to learn from labeled examples, enabling them to make more accurate predictions.

Sentiment Analysis: Translation vs. Native Language Processing

The sentiment analysis results highlight a crucial aspect of language-specific sentiment detection. The use of the NLTK VADER model on translated texts resulted in a significant loss of sentiment nuance, leading to an overwhelming classification of texts as neutral. This outcome emphasizes the limitations of sentiment analysis across languages, particularly when dealing with the subtleties of sentiment expressed in political texts. In contrast, models trained on native Hebrew texts, such as HeBert and the CNN-based model, were able to detect a predominance of negative sentiment. This difference suggests that sentiment analysis models are more effective when applied to texts in their original language, as translation can dilute the emotional and contextual cues necessary for accurate sentiment classification.

Impact of Named Entity Recognition (NER)

The application of NER to replace names with generic placeholders had a minimal effect on the overall sentiment detected by the models. This finding indicates that while names may carry emotional weight in certain contexts, the overarching sentiment of the texts is influenced more by the thematic content than by specific entities. This observation is particularly relevant for political texts, where the sentiment is often driven by the broader discourse rather than individual actors.

Conclusion

This report has presented a comprehensive analysis of text processing, feature extraction, clustering, classification, and sentiment analysis applied to a dataset of 15,000 documents. Through a combination of unsupervised and supervised learning models, the study explored the nuances of text analysis, from basic preprocessing to advanced sentiment detection. The comparison between unsupervised and supervised methods revealed the indispensable role of labeled data in achieving accurate classification. Furthermore, the sentiment analysis underscored the challenges of cross-language sentiment detection and the importance of using language-specific models for more nuanced insights.

The findings from this study offer valuable contributions to the field of data retrieval and text analysis, demonstrating the potential of machine learning models to uncover latent structures within large textual datasets. Moreover, the insights regarding the limitations of translation in sentiment analysis and the minimal impact of named entities on overall sentiment highlight critical considerations for future research and applications in text analysis.

Overall, this report underscores the complexity of text analysis and the importance of methodological rigor in processing and interpreting textual data. The advancements in machine learning and natural language processing provide powerful tools for uncovering the rich insights contained within text, offering both academic and practical implications for a wide range of domains.

Bibliography

https://github.com/omilab/Neural-Sentiment-Analyzer-for-Modern-Hebrew
https://github.com/amir-zeldes/HebPipe
https://huggingface.co/dicta-il/dictabert-ner
https://github.com/avichaychriqui/HeBERT