Humour Detection Project

Emily Wang
University of Calgary, emily.wang3@ucalgary.ca
Rohinesh Ram
University of Calgary, rohinesh.ram@ucalgary.ca
Osman Yaseen
University of Calgary, oayaseen@ucalgary.ca

Members Names	% of Contribution	Data Collection/Labelling	Coding	Write-up
Emily Wang	33	The initial data from the paper was collected and chosen as a team. Array 3 of the 'DataAdditional' csv table, as per the attached databricks notebook, was labelled by Emily as well as one third of the misclassified labels.	Label Distribution, Preprocessing, Feature Extraction, Machine Learning Pipelines.	Introduction, Discussion, Results, Abstract
Rohinesh Ram	33	The initial data from the paper was collected and chosen as a team. Array 1 of the 'DataAdditional' csv table, as per the attached databricks notebook, was labelled by Rohinesh as well as one third of the misclassified labels. Roh located the additional 3000 data points to be labelled.	Primary coordination between team members regarding coding in the notebook. Rohinesh shared the primary notebook on the screen while the code was peer-programmed, debugged, and trouble-shooted between teammates. Label Distribution, Percent Agreement and Cohen Kappa, Data Stats, Feature Extraction, Machine Learning Pipelines.	Discussion, Results, Abstract
Osman Yaseen	33	The initial data from the paper was collected and chosen as a team. Array 2 of the 'DataAdditional' csv table, as per the attached databricks notebook, was labelled by Osman as well as one third of the misclassified labels.	Label Distribution, Percent Agreement and Cohen Kappa, Data Stats, Feature Extraction, Machine Learning Pipelines.	Contribution, Discussion, Results, Conclusion, References, Abstract

LINK TO DATABRICKS NOTEBOOK:
https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/3717240578883042/1491066902451607/2304799003668431/latest.html

Abstract

Context

The original ColBERT paper uses BERT Sentence Embedding for featurization in a humour detection classification model. The concept is that the neural network in combination with the BERT Sentence Embedder increases accuracy of the ML model because the BERT encoder creates features which correctly identify the context of words in texts.

Objective

The objectives of the paper were (1) to verify the difference in accuracy between the ColBERT Sentence Embedding model versus baseline models and (2) determine if the baseline model could also distinguish between satire and non-satire humour.

Method

In order to achieve objective (1), the baseline models are replicated and the hyperparameters are tuned in order to achieve the highest possible accuracy for a baseline model. This is used to compare to the accuracy of the ColBERT Sentence Embedder model. In order to achieve objective (2), 1000 additional data points of satirical and non-satirical humour are added to the dataset. This allows us to test the accuracy on the original + new dataset.

Results

Logistic Regression ML model with 0.1 (regularization hyperparameter) performed at an F1-Score of 93.2%. Multinomial Naive Bayes with a 0 (smoothing parameter) performed at an F1-Score of 93.1%. Both results are using the combined datasets (including satire data points).

Conclusion

Overall, we were able to (1) extend the baseline model from the original paper to satire vs non-satire, and (2) verify that the 93.2% F1-Score achieved by the baseline models is still 6% less than the 98.2% F1-Score achieved by the ColBERT model.

1. Introduction

The demand for automatic humour detection has been rapidly on the rise. This demand stems from the various unique use-cases such as optimizing the effectiveness of communication with consumers for chat-bots and virtual assistants [1]. For customer service virtual assistants, it would aid in making the conversation more personable and realistic if humour was able to be detected [2]. This technology can even cater to the mental health industry in helping people with mental incapacities who are unable to understand or detect humour to the same extent. It could even help reduce the spread of fake news by being able to distinguish the difference between satirical and non-satirical news headlines. All these use-cases are why humour detection or natural language processing has been gaining so much traction and attention.

The paper that the team investigated was “ColBERT: Using Bert Sentence Embedding for Human Detection” by Issa Annamoradnejad and Gohar Zoghi [3]. The authors used bidirectional encoder representations (BERT), a transformer for natural language processing which looks at the context of a statement and generates the vector of a word to create their dataset. Sentence embeddings are created from the BERT model which are fed into a neural developed by the authors. This neural network has eight layers in total: one input, three hidden, one concatenation, and three final layers to ultimately determine the congruency of the sentences as well as discern the reader’s view on the sentence after reading the punchline. They used a 80% to 20% split and that resulted in a high accuracy and F1 score of 98.2%.

The purpose of the project is to verify whether the high accuracy was a result of their BERT Sentence Embedding neural network. The team replicated their results using a baseline machine learning model used in the paper, multinomial naive bayes, as well as with an unused model, logistic regression. For the multinomial naive bayes machine learning model they used a method called TF-IDF which first uses count vectorizer and the IDF vectorizer to generate numerical word representations. It was not mentioned in the paper that the authors performed hyperparameter tuning on their baseline models. Therefore, the team performed hyperparameter tuning to ensure validity of how much their neural network out-performed the baselines. Another goal was to determine if the model was able to detect humour in satirical contexts rather than just as regular jokes.

In terms of data collection, the satirical news headlines were extracted from The Onion [4] and the real news headlines or serious sentiments were taken from The New York Times, Breitbart, CNN, Business Insider, and Fox News [5]. Following the paper, the team filtered the two data sets by removing texts by removing texts that did not fall under a character length between 30-100 characters and 10-18 words. The two datasets were combined and filtered in alphabetical order to randomize the texts. Each of the group members manually labelled the 1000 texts, labelling 1 as satirical, and 0 as serious. The Cohen Kappa [6] was calculated as the metric for user agreement because it considers the possibility that

agreement occurred by chance. A highest agreement of the Kappa score of 66% was found between users 1 and 3.

Logistic regression performed slightly better on the original dataset in comparison to the old dataset and new dataset combined. For logistic regression, both the original and combined datasets performed the best with a regularization parameter of 0.1 with accuracy scores and F1-scores of 93.5% and 93.2%, respectively. For multinomial naive bayes, the model performed better on the combined dataset. The original dataset performed the best with a 0.2 smoothing parameter, and the combined dataset performed the best with a 0 smoothing parameter with accuracy scores of 90.3% and 93.0%, respectively. The F1 scores were 90.5% and 93.1%, for the original and combined datasets respectively.
The misclassifications of the best performing model was done on logistic regression with a lambda of 0.1. The misclassification analysis of humorous texts showed that the majority were due to the joke requiring readers to have previous knowledge about the contents of the joke (category 3) of 28% of the total misclassifications according to **Appendix A: Figure I**. In the current day, a lot of texts utilize sarcasm or puns that incorporate knowledge of social and political issues to induce humour. Similarly for non-humorous texts, the most misclassifications also fall under requiring the reader to have an external understanding of the social, political, or environmental context (category 10) as 7% of the total labels. A leading cause of the misclassification was therefore due to the model unable to detect satirical sentiments which often relies on a user's understanding of real world issues or situations. From **Appendix A: Figure I**, it is also evident that the model misclassified more humorous text as non-humorous than the other way around.

2. Results

2.1 Data Collection and Labelling

The team collected the satirical news headlines from The Onion, a famous American digital media company [4]. The team also collected news headlines with serious sentiment from various sources such as The New York Times, Breitbart, CNN, Business Insider, and Fox News [5].

The team filtered the two data sets by removing texts that did not have character length between 30-100 characters and 10-18 words to ensure that the 1000 additional texts would fit the criteria of data discussed in the paper. After the filtering step there were 1500 satirical texts and 1500 serious texts. The two datasets were combined and alphabetized to randomize the texts, thereby preventing any labeller to cheat while labelling. One-thousand texts were kept for the labelling step.

The team approached labelling by having each team member manually label the 1000 texts, labelling 1 as satirical, and 0 as serious. **Table I** shows the agreement between the data labellers using percent agreement and Cohen Kappa. Cohen Kappa was chosen as a metric for agreement because it considers the possibility that agreement happened by chance. A kappa score of 0.41– 0.60 is considered moderate agreement while a kappa score of 0.61-0.8 is considered as substantial agreement [6].

Table I: Agreement between data labellers

Labeller	1 and 2	1 and 3	2 and 3
Percent Agreement	0.76	0.83	0.76
Cohen Kappa	0.52	0.66	0.55

Consensus on the final label for each text was reached by choosing the final label as the majority label amongst the three labellers. For example, if labeller 1 and labeller 3 labeled a text as humorous but labeller 2 labelled it serious, the final label was considered to be humorous.

2.2 Additional Data Compared to Original Data

The approach to comparing the additional data to the original data was to calculate the summary statistics outlined by the authors in the paper. For example, the authors calculated statistics involving number of characters, words, unique words, punctuations, duplicates, and number of sentences. **Table II** shows summary statistics of the original ColBert dataset generated by the authors of the paper. **Table III** shows the same summary statistics generated by the team for the additional 1000 texts. The statistics for the ColBert and additional 1000 texts are similar for the number of characters, words, and sentences, but are different for the number of punctuation and duplicate words. The additional 1000 texts have fewer punctuation and duplicate words because the additional dataset consists of news headlines. News headlines typically do not contain punctuation nor conjunctions which are likely the source of duplicate words in the Colbert dataset.

Table II: General Statistics of the ColBert Dataset

	#chars	#words	#unique words	#punctuation	#duplicate words	#sentences
mean	71.561	12.811	12.371	2.378	0.440	1.180
std	12.305	2.307	2.134	1.941	0.794	0.448
min	36	10	3	0	0	1
median	71	12	12	2	0	1
max	99	22	22	37	13	2

Table III: General Statistics of the New Dataset

	#chars	#words	#unique words	#punctuation	#duplicate words	#sentences
mean	63.074	12.413	12.204	0.924	0.209	1.049
std	10.525	2.073	1.944	0.978	0.462	0.216
min	36	10	9	0	0	1
median	56	11	11	0	0	1
max	88	21	20	5	3	2

Figure I, II, and III show the distribution of the labels for the additional, Colbert, and Colbert + additional datasets respectively. The number of serious labels in the additional dataset was 491 while the number of humorous labels was 509. The number of serious labels and humorous labels in the ColBert dataset was 100,000 for each. The combined dataset therefore had 100,491 serious labels, and 100,509 humorous labels. All three of the datasets are balanced overall.

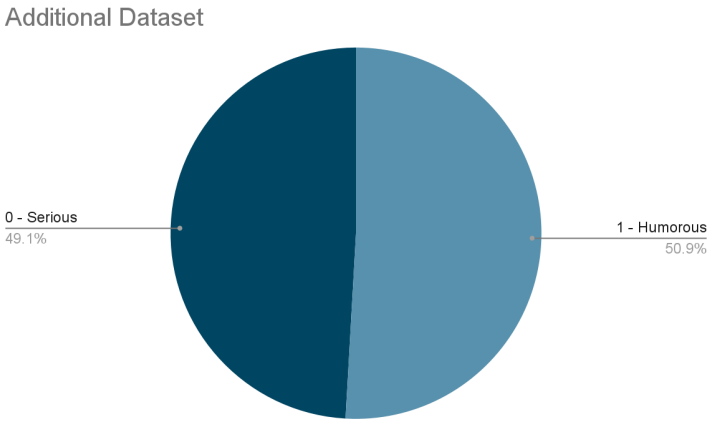

Figure I: Pie chart showing the distribution of serious texts to humorous texts in the additional 1000 texts

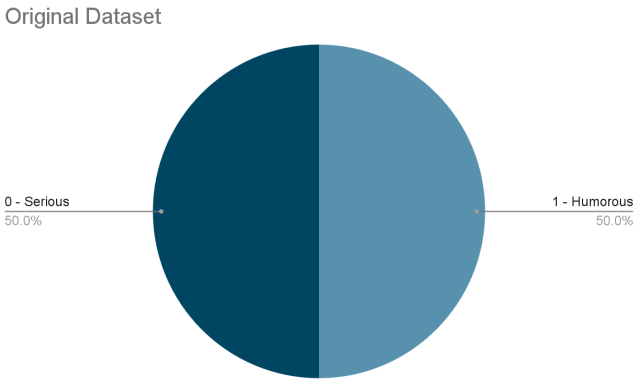

Figure II: Pie chart showing the distribution of serious texts to humorous texts in the ColBert dataset

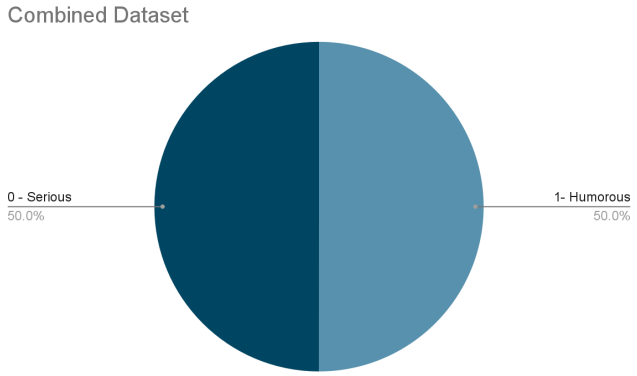

Figure III: Pie chart showing the distribution of serious texts to humorous texts in the ColBert + additional dataset

2.3 Additional Data Pre-Processing

In this section, the approach was to follow the preprocessing steps discussed by the authors in the paper. As mentioned earlier, texts that did not have 30-100 characters and 10-18 words were discarded. The authors of the paper also applied sentence case formatting and expanded contractions. For example, “isn’t” was expanded to “is not”. Stop words and punctuation were not removed from the texts as they were not removed by the authors in the ColBert dataset.

The team used term frequency inverse document frequency (TF-IDF), more specifically the combination of CountVectorizer and IDF from Apache Spark MLLib, to generate the feature vectors. TF-IDF was used because the statistic reflects the importance of a word in the context of the document which, as discussed by the authors in the paper, is important to distinguish if a text is humorous or non-humorous. **Table IV** shows the summary statistics of the data after preprocessing. The summary statistics of the data after preprocessing are identical to the data before preprocessing.

Table IV: Summary statistics of the additional data after preprocessing

	#chars	#words	#unique words	#punctuation	#duplicate words	#sentences
mean	63.074	12.413	12.204	0.924	0.209	1.049
std	10.525	2.073	1.944	0.978	0.462	0.216
min	36	10	9	0	0	1
median	56	10	11	0	0	1
max	88	21	20	5	3	2

2.4 Model Selection: Hyper-Parameter Tuning

The team selected two machine learning algorithms for humour prediction: Logistic Regression and Multinomial Naive Bayes (MNB). The hyperparameters available for logistic regression in Apache Spark MLLIB were: regularization parameter and elastic net, and for multinomial naive bayes the only parameter available was smoothing. The hyperparameter that was selected to be manipulated for logistic regression was the regularization parameter. The hyperparameter that was selected to be manipulated for multinomial naive bayes was the alpha, or smoothing parameter since that was the parameter used in the paper.

Hyper-parameter tuning in combination with cross validation were used on the training dataset to maximize the F1-score of the two models. Note that F1-score was used as this was the metric published by the authors of the paper.

The selected hyperparameter for logistic regression was the lambda parameter which is used to control regularization. Lambda values of 0, 0.1, and 1 were used. **Figures IV and V** show the precision, recall, F1, and accuracy scores of the logistic regression model over the range of lambda values for the ColBert and combined datasets respectively. The best F1-scores for logistic regression on both datasets was achieved with the lambda parameter set to 0.1. The F1 scores on the ColBert and combined datasets were 0.9349 and 0.9316 respectively.

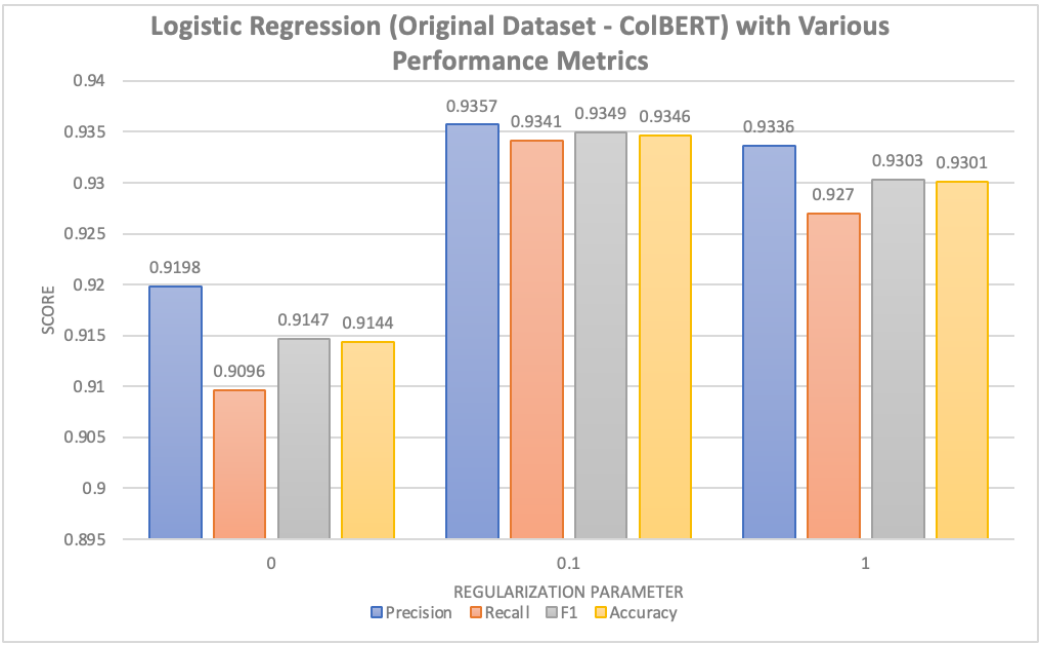

Figure IV: Logistic Regression Model with The Original Dataset and 0, 0.1, and 1 as the Regularization Parameters to Determine the Overall Score

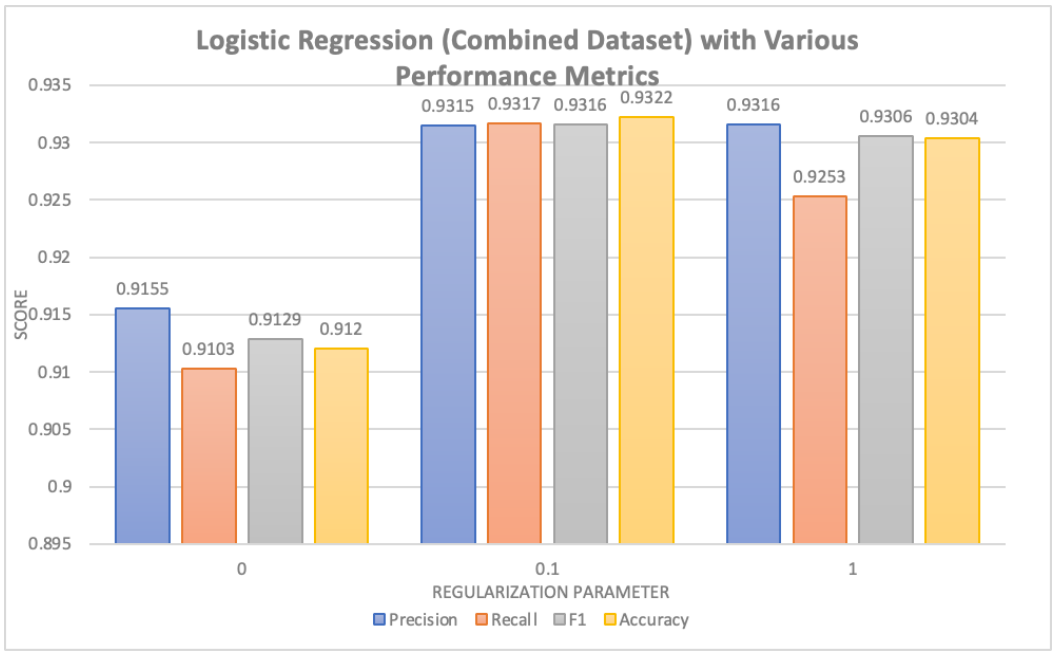

Figure V: Logistic Regression Model with The Combined Dataset and 0, 0.1, and 1 as the Regularization Parameters to Determine the Overall Score

The selected hyper-parameter for MNB was the smoothing parameter. In the paper, the authors used multinomial naive bayes with the smoothing parameter set to 0.2 as one of their baseline models. The

smoothing parameters selected by the team were 0, 0.2, and 0.4 to calculate precision, recall, F1-score, and accuracy. **Figures VI and VII** show scores of the multinomial naive bayes model over the range of smoothing values for the ColBert and ColBert + additional datasets respectively. The best MNB model on the ColBert dataset had a smoothing parameter set to 0.4 and an F1-score of 0.905 which is 2.6% higher than the score for MNB published by the authors of the paper (0.882). The best MNB model on the ColBert + additional dataset had a smoothing parameter set to 0 with an F1-score of 0.931.

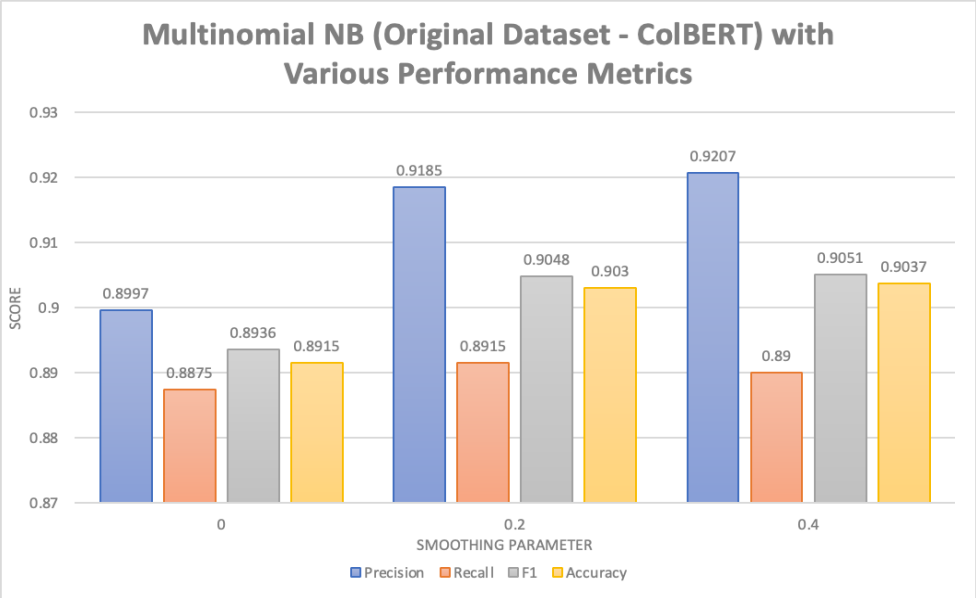

Figure VI: Multinomial Naive Bayes Model with the ColBert Dataset and 0, 0.2, and 0.4 as the Smoothing Parameters to Determine the Overall Score

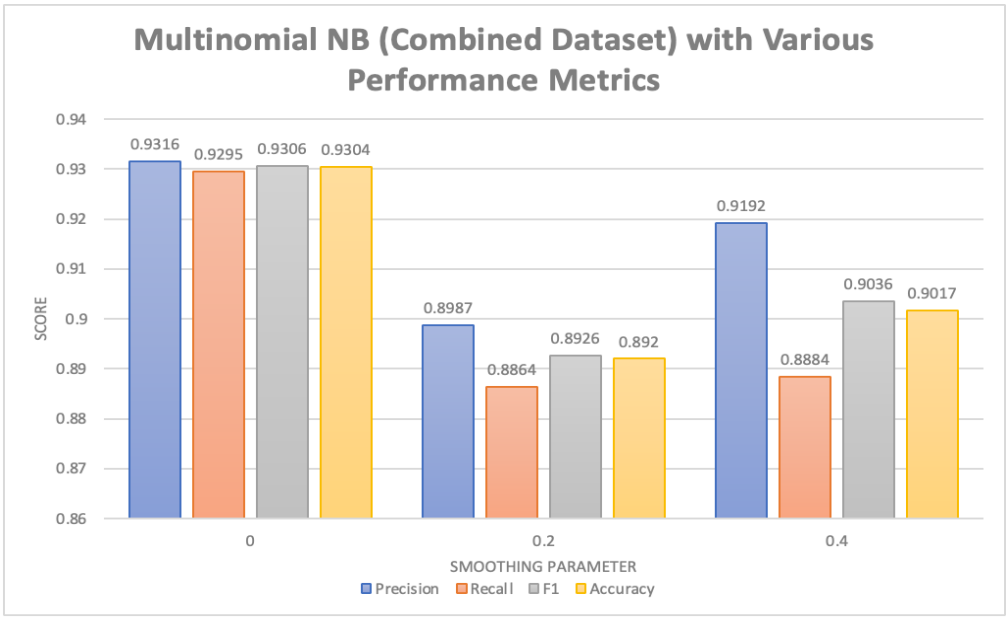

Figure VII: Multinomial Naive Bayes Model with the Combined Dataset and 0, 0.2, and 0.4 as the Smoothing Parameters to Determine the Overall Score

Overall, the best model was logistic regression with lambda set to 0.1 on both the ColBert and combined datasets.

2.5 Model Performance

The approach to this section is to follow the paper and calculate the accuracy, precision, recall and F1-score for selected models to compare with the author's proposed model.

As discussed in section 3.4, the best performing model on both datasets was logistic regression with the lambda parameter set to 0.1. **Figure VIII** shows the confusion matrix for the 20% test data on the two datasets. Logistic regression classified more text as humorous when they were labelled non humorous on the ColBert dataset compared to the amount of texts it classified as non humorous when the texts were labelled humorous. On the other hand, the model classified more texts as non humorous when they were labelled humorous on the combined dataset compared to the amount of texts the model classified as non humorous when they were labelled humorous. This suggests that the model has a harder time detecting humour in the satirical texts of the additional data.

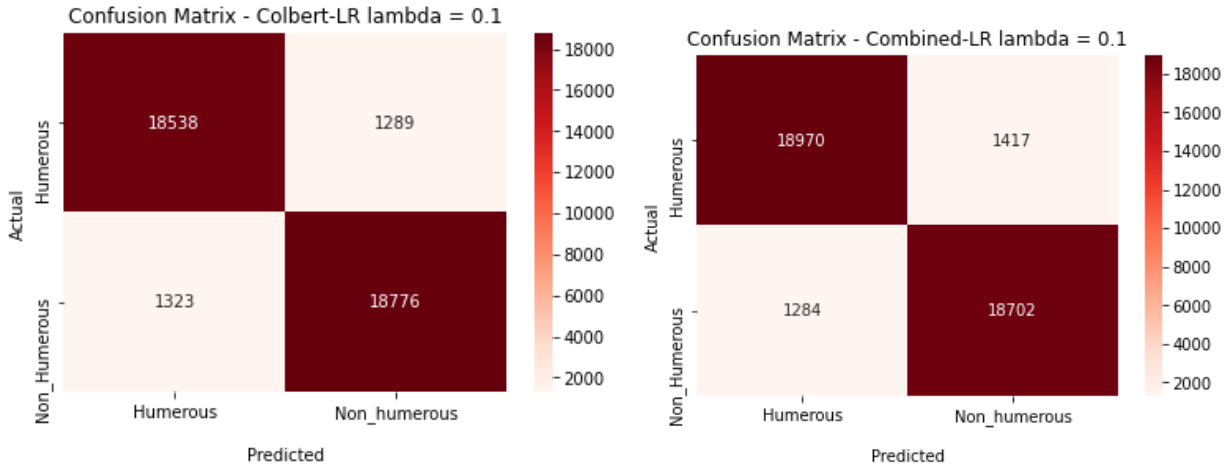

Figure VIII: Confusion matrix of the test dataset on the ColBert and combined datasets with Logistic Regression Lambda = 0.1

Table V shows the comparison between the baseline models selected by the authors in the paper with the logistic regression model developed by the team. The model developed by the team outperforms the baseline models (decision tree, SVM, MNB, XGBoost, XLNet) in the paper but does not perform better than the neural network developed by the authors. Logistic regression is a model that performs well for natural language processing because linear models perform better in high dimensions with lots of features. [7] In this paper, the feature count of the feature vectors resulting from count vectorizer and

inverse document frequency were 54,550. Furthermore, the number of data points was 200,000 which further contributed to the high performance of the logistic regression model.

Table V: Scores for Models in the Paper Compared to Logistic Regression lambda = 0.1 on the ColBert Dataset

Method	Configuration	Accuracy	Precision	Recall	F1-Score
Decision Tree		0.786	0.769	0.821	0.794
SVM	sigmoid, gamma = 1.0	0.872	0.869	0.88	0.874
MNB	alpha = 0.2	0.876	0.863	0.902	0.882
XGBoost		0.720	0.753	0.777	0.813
XLNet	XLNet-Large-Cased	0.916	0.872	0.973	0.920
Neural Network		0.982	0.99	0.974	0.982
Logistic Regression	**lambda = 0.1**	**0.935**	**0.936**	**0.934**	**0.935**

Table VI shows the performance of the logistic regression model on the combined data set. The performance of the model on the combined dataset is slightly lower than the performance of the model on the ColBert dataset which suggests that the model had a harder time distinguishing between humorous and non-humorous texts with the addition of the satirical news headlines.

Table VI: Scores for Logistic Regression Lambda = 0.1 on the Combined Dataset

	Accuracy	Precision	Recall	F1
Logistic Regression	0.932	0.932	0.932	0.932

2.6 Misclassifications of LogisticRegression Lambda = 0.1

The approach for this section was to gather 200 misclassified for the best performing model (Logistic Regression lambda = 0.1) to determine where and how the model failed. Twelve categories were created for misclassification and are summarized in **Table VIII**. Categories 1-7 explain the cases where the true classification was 1 for humorous but classified by the model to be 0 for non-humorous. Categories 8-12 explain the cases where the true label was 0 for non-humorous but classified by the model to be 0 for non-humorous.

Table VIII: Table of Misclassification Labels with Explanation with an Example

Misclassification Category	**Misclassification Explanation**	**Example**
1	Team mislabelled a non-humorous text as humorous	Airlines have banned passengers from taking tweezers on board

2	Joke requires the reader to have knowledge regarding pop culture to understand the joke	Ama request paul mccartney how big of an impact has kanye been to your music career blowing up
3	Requires reader to have previous knowledge about the contents of the joke (e.g political)	Lindsey graham gazes longingly at happy rubio campaign workers through window
4	Joke is based on readers background such as age, culture ethnicity and other environmental factors	Man feeling pressure to live up to conversation between barber and customer in next chair
5	Joke is ironic and requires the reader to understand the context of the joke not expressed in the contents of the joke	I'm a conservative says horrifying man 25 years from now
6	Joke is sarcastic in nature and requires the reader to understand the context of the joke not expressed in the contents of the joke	Lethal injection least effective drugs man took while in prison
7	Joke is pun and requires the reader to understand the context of the joke not expressed in the contents of the joke	A car pool is an extravagant waste of water
8	Serious text or headlines have unrelated subjects and sentiment in their contents, and therefore follows the typical pattern of a joke	A husband a house a mortgage a baby a light bulb moment
9	Serious text or headlines have opposite subjects and sentiment in their contents, and therefore follows the typical pattern of a joke	Amanda lost 110 pounds it easy no was it worth it yes
10	Serious text requires reader to have previous political, social, environmental knowledge	An open letter to all stepdads and especially my husband
11	Serious text is absurd in nature	All it takes to buy a stolen password on the internet is 55 cents
12	Team mislabelled a humorous text as non-humorous	Baby can not even comprehend how cool his ball popper toy is

Figure V is a barchart showing the distribution of misclassifications over their respective categories. The model misclassified 165 texts as non-humorous when the true label was humorous and 35 texts as humorous when the true label was non-humorous. The most frequent reason for misclassification was in category 3, which required the reader of the text to have previous knowledge regarding the subjects of the text, for example political, to understand the humour. The model failed to classify these correctly because the humour cannot be detected within the contents of the text, but rather the overall meaning of the subjects.

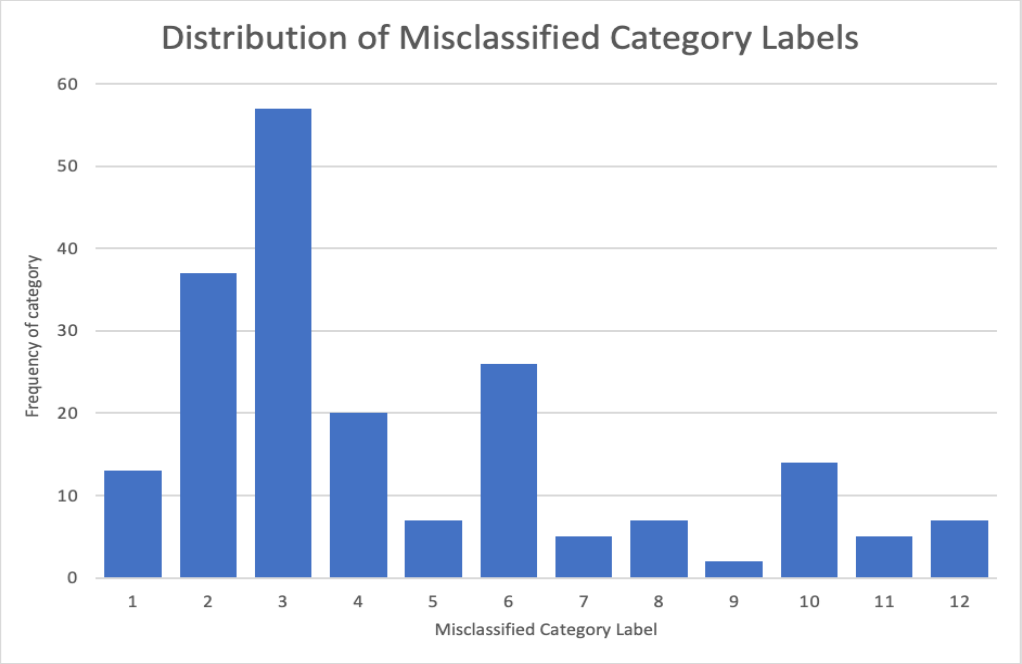

Figure V: Distribution of Misclassified Category Labels for 200 texts

Appendix A Figure I also shows a pie chart of the percentage of misclassification for each category. It shows that category 3 for misclassification of humorous sentiments and category 10 for misclassification of non-humorous sentiments accounted for the majority of the mislabelled data. Both of these categories follow the reasoning that the reader must have previous knowledge on external information.

3. Discussion

As shown in the results, the implication was that the models cannot detect satire as well as regular jokes. This can be seen in the decrease in the accuracy score when the team's labelled dataset was added. However, a part of this could have been caused by human error where the team mis-labelled a sentiment. Based on the results, developing a model that understands satirical and sarcastic texts as well as having prior knowledge or context of the text could further develop the advancement of these use-cases as listed below.

Implication 1 (Emily) - Humour is a highly intelligent commutative activity that is used in daily life as a sociological function. Helping understand humour could help establish stronger social relationships, and better understand a person's overall intention. In the real world, individuals struggling with social incapacities such as aspergers, down syndrome, or autism may find these natural language models to be extremely useful for daily functioning. With further development and optimization, these models could be automated and expanded into an app to help these patients detect humour and better understand the overall intent of a person. This could aid in eliminating social barriers and problems such as bullying, thus enhancing the lives of millions. This is difficult to accomplish though because of a user's background such as age, culture and ethnicity, and other environmental factors.

Implication 2 (Osman) - With chatbots and virtual assistants on the rise it is crucial for computers to understand and detect humor. It would help in predicting human intention and provide overall better customer service for machine-human interaction systems [1]. However, different types of humour requires an external or past knowledge such as irony or sarcasm, making automated humour detection an even more difficult topic. With chatbots, there are two components: humor generation and understanding humour. In this project, the team has only uncovered and attempted to optimize mechanisms for the model to understand humour, but not generate it [2].

Implication 3 (Rohinesh) - Satirical and non-satirical new headlines classification.
A third implication the team uncovered was the use of distinguishing between satirical and non-satirical news headlines. This is an important use case as the abundance of fake or satirical news headlines is misleading and only works if an individual understands the humour. Otherwise, it is a leading cause of the mis-spread of fake news; it operates under the disguise of real news for monetary and political gain. With a model that can detect the difference between satirical news headlines versus non-satirical news headlines to a high accuracy, an individual will be less susceptible to sharing a news article that promotes false information.

4. Conclusion

While the purpose may vary, it was clear there is a need for automated humour detection in society today and into the future. Therefore, using machine learning in this space can be applied in a variety of ways. This includes assisting those who are unable to detect humour on their own, optimizing chatbots and virtual assistants, and establishing satire versus non-satire in news and other forms of media.

Through the use of the "ColBERT: Using BERT Sentence Embedding for Humor Detection" paper, written by Issa Annamoradnejad, Gohar Zoghi, the baseline results were not only replicated, but also the model was extended to distinguish the difference between satire and non-satire, using the additional dataset. With this addition of 1000 labelled additional records, the team ran the combined dataset through two machine learning pipelines: logistic regression and multinomial naive bayes. The hyperparameters were tuned for each pipeline. For logistic regression, best performing models were with the regularization parameter being 0.1 for both the original and combined dataset. For multinomial naive bayes, the original dataset performed equally good using a smoothing parameter of 0.2 and 0.4, but the combined dataset outperformed using 0 as the smoothing parameter.

While the models did not achieve the F1-score of 98.2% as the ColBERT model using sentence embedding with neural networks, this was expected. The goal with this paper was to replicate, verify and where possible, outperform, the baseline models used in the original ColBERT model by tuning the two chosen ML model's hyperparameters. Although in the paper, the authors did not mention hyperparameter tuning, the team discovered that with this extra step, the baseline model (multinomial naive bayes) on the original dataset hit as high as 90.5% for the F1-score whereas in the model in the paper using an alpha of 0.2 was only able to achieve 88.2%. This goes to show that the authors did not intend to configure the baseline model to their optimal performance to set a fair comparison. The authors claimed that there was a 10% difference between the baseline multinomial naive bayes model and their proposed neural network. The team was able to decrease the gap 7.7% difference in the F1 score after introducing hyperparameter tuning.

Overall, the team was able to extend the baseline model from the original paper to satire versus non-satire, and verify that the F1-score achieved by the baseline models is still 7.7% less than the 98.2 % achieved by the ColBERT model. The team was able to improve the base-line model performance by introducing the hyperparameter tuning step.

5. Appendix A:

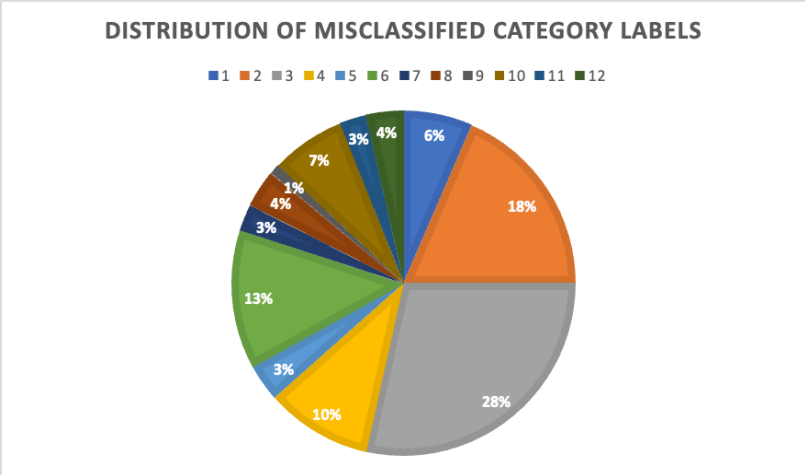

Figure I: Pie Chart of the Distribution of Misclassified Category Labels

6. References

[1] P.-Y. Chen and V.-W. Soo, “Humor recognition using deep learning.” [Online]. Available: https://aclanthology.org/N18-2018.pdf. (Accessed: 10-Dec-2021).

[2] T. Vishwakarma, “‘humorist bot: Bringing computational humor in a chat bot system,’” *Medium*, 28-Jun-2017. [Online]. Available: https://medium.com/@tusharpce536/a-review-on-humorist-bot-bringing-computational-humor-in-a-chat-bot-system-7438c15d2813. (Accessed: 11-Dec-2021).

[3] I. Annamoradnejad and G. Zoghi, “Colbert: Using Bert Sentence embedding for humor detection.” [Online]. Available: https://arxiv.org/pdf/2004.12765.pdf. [Accessed: 01-Dec-2021].

[4] Lukefeilberg, “GitHub - Lukefeilberg/onion: Dataset of The Onion articles and real ‘Onion-like’ news articles from the subreddit r/NotTheOnion, along with a jupyter notebook extracting the dataset and performing classification.,” *GitHub*, Feb. 25, 2020. https://github.com/lukefeilberg/onion (Accessed Dec. 14, 2021).

[5] A. Thompson, “All the news,” *Kaggle*. https://www.kaggle.com/snapcrack/all-the-news (Accessed Dec. 14, 2021).

[6] M. L. McHugh, “Interrater reliability: the kappa statistic,” *Biochemia Medica*, pp. 276–282, 2012, doi: 10.11613/bm.2012.031.

[7] Andreas C. Muller and Sarah Guido, Introduction to Machine Learning with Python. First Edition. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O’Reilly Media Inc, 2016