LOL (Laugh or Lour)

Humor and Offense Detection from Sentences

CSCE 638 NLP Group Project, Spring 2021

Tian Liu, Ming-Hsien Chien
Liuyi Jin, Yijun Zhang

Apr 30, 2021

1 Abstract (By Liuyi Jin)

Humor detection is an important task as it is employed in various real applications. It's also a challenging problem due to its subjectivity. To show how to make the machine perform good in humor detection and related tasks, we implement ColBERT as the solution for 4 humor-related tasks: humor detection, controversy classification, humor extent regression, and offense extent regression. ColBERT is trained based on BERT pretrained model. We base this project on the Task 7 of SemEval2021[1], which focuses on humor and offense detection of sentences. We compare 3 BERT-based models (2 ColBERT variants, ALBERT) with 2 baseline models (Naïve Bayes, Perceptron). The results show ColBERT can achieve the best humor detection and controversy classification performance with 0.942 and 0.6242 F1 score, respectively. ColBERT is also performant in humor extent regression and offense regression with 0.5324 and 0.4678 root mean square error (RMSE), respectively. We analyze the pros and cons of different models and conclude with our top contest results.

2 Introduction (By Yijun Zhang, Ming-Hsien Chien)

Humor appreciation is a highly subjective phenomenon. People of different ages, gender, and socioeconomic status tend to have a different perception of a joke. And there seems to be no specific standard to tell if a sentence is generally humorous or not. For a joke, there is tough decide if it is only humorous or offensive. For example, if someone is in the specific group, he or she will understand the joke of the clique. Otherwise, he or she might feels uncomfortable for the same joke. To avoid the embarrassing situation, we are inspired to invest to the technique of humor and offense detection, which could help us construct the practical and convenient applications to human-centered interaction and artificial intelligence systems together such as chatbots and virtual assistants. In addition, it assists people to pick fun words, recommending fun content from social media, etc. The project was based on the Task 7 of SemEval 2021. This is the first task to combine humor and offense detection.

3 Dataset and Tasks (By Yijun Zhang)

3.1 Dataset

The dataset of the project was provided by the task organizer on Codalab. The dataset consists of 10,000 sentences and each comes with four labels including is_humor (binary classification 0-1), humor_rating (regression between 0 and 5), humor_controversy (binary classification 0-1), and offense_rating (regression between 0 and 5). The data was collected by the competition organizer from a balanced set of age groups from 18-70.

We split the dataset with 8,000 sentences for training and 1,000 for validation. For testing, we use 1,000 unlabeled sentences, and upload the output to CodaLab to get the testing scores. The following table shows several samples of the dataset.

Table 1: Sample items of the dataset

id	text	is_humor	humor_rating	humor_controversy	offense_rating
8001	What's the difference....	1	2.45	0	1.7
8002	Vodka, whisky, tequila...	1	2	0	0
8003	French people don't...	1	2.95	0	1.15
...	...	...	...	...	...

3.2 Tasks

There are four tasks for this project as given by the contest organizer:

1. Task 1a: is_humor (binary classification: 0 or 1)
2. Task 1b: humor_rating (regression between 0 to 5)
3. Task 1c: humor_controversy (binary classification: 0 or 1)
4. Task 2: offense_rating (regression between 0 to 5)

4 Challenges and Related Work

4.1 Challenges in Humor Detection (By Tian Liu, Yijun Zhang)

Detecting the humor in texts is not an easy task. One reason is that humor is a very subjective sensation. The extent to which a person may sense humor depends on his/her age, personal experiences, educational background, culture, religion, or even personality, etc. What seems humorous to one person may not be the same as another. In addition, there are various types of humor, for example, wordplay, irony, sarcasm, sexual, etc. Different jokes used different mechanisms to create humor. Moreover, some types of humor would require substantial external knowledge such as literature background, particular knowledge on some aspects, etc. All of these make humor detection in texts a challenging task.

The following three jokes give some examples of different types of humor.

"Did you hear about the guy whose whole left side was cut off? He's all right now."

"The one who invented the door knocker got a No Bell prize"

"Your village called. They want their Idiot back."

If a sentence is classified as a humorous sentence (a joke), we further decide whether it is offensive or not and predict its offense rating. It is challenging because of two reasons:

1. People have different opinions about offensive jokes. People with similar cultural background could easily decide a sentence is a joke or not, but they may have totally different opinions about whether it is offensive. For example, some people believe that offensive humor such as sexist or racist jokes can help break down barriers and challenge prejudice. Others simply find it appalling.
2. The data becomes less. Among the 8,000 sentences of our training data, there is only have 4,932 sentences are labeled as *"is_humor = 1"*. Hence, we could only train our models for task 1c using these 4,932 sentences.

4.2 Related Work (By Tian Liu, Yijun Zhang)

To capture the humor in texts, first we need to understand the latent humor structure, i.e. where does the humor come from. Yang et al. [2] analyzed the semantic structure behind humor from four perspectives: incongruity, ambiguity, interpersonal effect and phonetic style and designed a series of features for each structure as potential indicators of humor. Yang et al. also proposed an easy algorithm to extract the humor anchor in the texts.

With recent development in deep learning techniques, there have been a lot of studies on using deep learning for humor detection. Bertero et al. [3] used Long-Term-Short-Memory (LSTM) model to capture the latent sentence structures in humorous texts. Chen et al. [4] used Convolutional Neural Network (CNN) with augmented filter sizes and numbers to capture the humor features from texts. To ease the heavy training of neural networks, they implemented the concept of highway networks to allow information shortcuts.

There are studies in humor detection and applying BERT in humor detection. Annamoradnejad[5] use BERT to generate embeddings for sentences, and then use these embeddings for neural network training. Wang et al.[6] utilizes BERT model to understand item titles and relevance between items for e-commerce recommender systems.

Most of the previous research on humor detection has been focusing on text resources alone. Recently, some researchers have tried to detect humor from multimodal resources such as including text and speech information. Yang et al.[7] proposed using the time-aligned comments in videos to generate unsupervised labels and then train the model with the labelled data to predict the humor from texts and audio data. The project workload distribution is shown in Section 5.

5 Distribution of the project

5.1 Coding

- Tian Liu: ColBERT, Data Augmentation
- Ming-Hsien Chien: ALBERT
- Liuyi Jin: Naïve Bayes
- Yijun Zhang: Perceptron

5.2 Presentation

- Tian Liu
- Ming-Hsien Chien: Presenter

5.3 Report

Please also refer to the section titles for each team-members' work.

- Tian Liu: Challenges and Related Work, Approach, Humor Structure, ColBERT, Data Augmentation
- Ming-Hsien Chien: Introduction, Albert model, Conclusion
- Liuyi Jin: Abstract, Naïve Bayes, Evaluation and Discussion
- Yijun Zhang: Introduction, Dataset and Tasks, Challenges and Related Work, Perceptron, Conclusions

6 Our Approach (By Tian Liu)

In this section, we will discuss how we capture the latent humor structures using BERT sentence embedding. Also, we will introduce the different models we used. Specifically, we implemented the ColBERT model by Annamoradnejad[5] and explored variants of BERT embedding models.

To begin with, let us look at the general structures of jokes and how we can capture it.

6.1 Capture the Humor Structure with ColBERT (By Tian Liu)

There has been a lot of works in identifying the humor presence in sentences based on its content and structures. The Semantic Script Theory of Humor (SSTH) presented by Raskin [8] stated that a text has to have two distinct scripts that are opposite in nature, such as real/unreal, possible/impossible, for it to be humorous.

Therefore, we aim to capture two types of humor structure:

1. The relationships between sentences (the presence of punchline)
2. Word-level connections (synonyms and antonyms)

For example, in the joke *"Did you hear about the guy whose whole left side was cut off? He's all right now."*, the second part *"He's all right now."* is the punchline that brings contradiction into the story and makes the whole text laughable. So it's important for us to capture the relationships between sentences, especially the presence of punchline.

In another example, *"The one who invented the door knocker got a No Bell prize"*, the humor comes from the synonyms or antonyms, i.e. *door knocker* and *No Bell prize* in the sentence. Thus, it is also necessary for us to examine the word-level connections in the text (such as the presence of synonyms and antonyms) that might have affected the congruity of the text.

For this project, we used the ColBERT model to capture these two humor structures.

6.2 ColBERT (By Tian Liu)

For this project, we mainly focused on implementing and exploring the ColBERT model based on Annamoradnejad's work [5]. The ColBERT model uses BERT to generate sentence embeddings of a given text and uses these embeddings as input for parallel lines of hidden layers in a neural network to capture the mid-level features for each sentence. These output of these lines of hidden layers are then combined in the final layer to output the target values. Figure **??** shows the architecture of the ColBERT model.

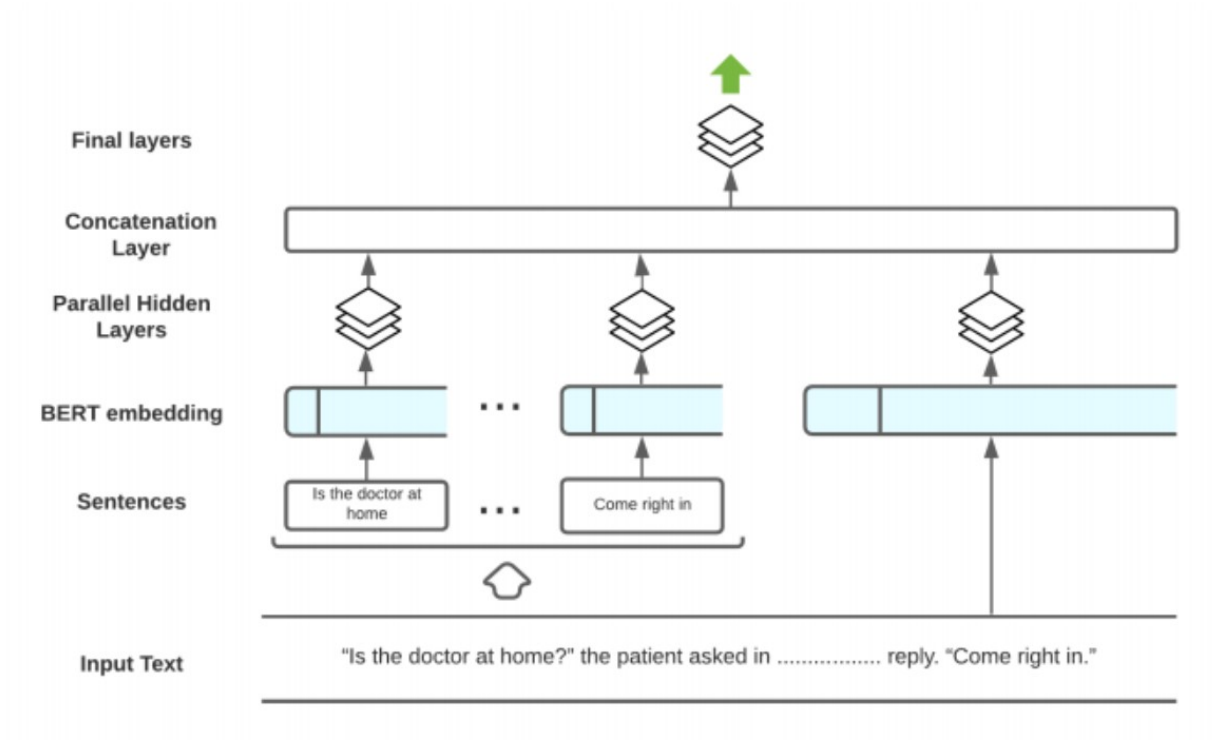

Figure 1: Architecture of ColBERT Model

The ColBERT model consists of several key steps:

1. The whole input sentences were separated and tokenized individually.
2. Then sentences were encoded using BERT sentence embedding. This is done on each separated sentences as well as on the whole text.
3. The encoded items were fed into several parallel hidden layers of neural network to extract the features between sentences (could be related to context, type of sentence, etc). That is how the presence of punchline is detected.
4. To capture the word-level connections, the BERT embeddings of the whole sentences were generated and fed into the a set of hidden layers of neural network, as shown in the right side of the Figure **??**.
5. Then all the features are concatenated in the concatenation layer and fed into the final layers of neural network to generate the target value for each task.

To emphasize our contributions, the original ColBERT model used BERT-Base-Uncased pretrained model for sentence embeddings, and it was built for humor classification task. In this project, we also explored several

variants of BERT model, including the BERT-Large-Uncased and ALBERT. The BERT-Large-Uncased model has much more parameters than the BERT-Base-Uncased model, and was trained on a much larger corpus dataset. The ALBERT model is a light version of BERT model, which has much fewer parameters. The results of different models are compared with the two baseline models: Naive Bayes and Perceptron.

On the other hand, to handle the regression tasks in this project (task 1b and task 2), the neural network was modified accordingly to generate desired target output.

6.3 Text Data Augmentation (By Tian Liu)

During the development of the ColBERT based on BERT-Large-Uncased model, it is found that for task 1b and task 1c, no matter how we tune the hyperparameters (training epochs, batch size, learning rate), the model will always overfit and predict a straight line. This is because the task 1b and task 1c are based on the humorous data in the dataset, which contains only half size of the original data set. And the BERT-Large-Uncased model has large amount of parameters, which can easily cause overfitting on a small dataset.

Thus, to solve the problem, data augmentation is necessary. In this project, we did the data augmentation using Back Translation using the translation sevice provided by Google. Specifically, we translate the humorous text from English to German and then translate it back to English. During the translation, some new words will be added to the back translated text but the general meaning will not change much. Finally, we assign the same humor rating, humor controversy and offense rating to the back translated text. In this way, we obtained twice the number of the original data points we have for task 1b and task 1c.

The results on task 1b and task 1c shows that the ColBERT-Large can work well with the augmented data in task 1b. However, for task 1c, the overfitting problem remains even with the augmented dataset.

6.4 ALBERT(By Ming-Hsien Chien)

A Lite Bert(ALBERT) is the improvement from the BERT model, which is applied by the pre-training model, but its parameters only have ten times as few as the BERT model. To mitigate the training size when scaling the pre-train model, the ALBERT model has three critical improvement to reduce the parameters.

The first one is dividing the size of hidden layer from the size of vocabulary embedding, which helps control the size of hidden layer when the increasing of the vocabulary embedding. The second technique provides the sharing across the different layers.The last one is the coherence loss between the sentences and it also attaches a binary classification to detect whether two blocks show up together in the input document. These three revision of the ALBERT model will improve the performance of downstream task a lot for the incoming multiple sentences.

6.5 Naïve Bayes(By Liuyi Jin)

Here we apply Naïve Bayes (NB) philosophy to the classification tasks (i.e., task 1a, 1c). Task 1a and 1c in this project are classification tasks. We leverage simple multinomial Bayes rule to classify the text samples. Note here we use bag of words to represent a text document. Given a text d and bayes equation shown in Equation 1, we predict the most probable class c for d with the maximized posterior class value as shown in 2.

$$P(c|d) = \frac{P(d|c)\,P(c)}{P(d)} \tag{1}$$

$$c_{MAP} = \underset{c \in C}{argmax} P(d|c)\, P(c) \tag{2}$$

The problem with Naïve Bayes method is that the massive parameters are needed. Multinomial Naïve Bayes assumes independency beween the probability of each word given a text. Thus we actually are applying Multinomial Naïve Bayes (MNB) method as shown in Equation 3, where i and j refer to each word index i in class c_j. Laplace smoothing and log space techniques are also applied to address the unknown words and underflow problems, respectively.

$$c_{MNB} = \underset{c_j \in C}{argmax} P(c_j) \prod_i P(x_i|c_j) \tag{3}$$

6.6 Perceptron (By Yijun Zhang)

Perceptron model is used in binary classification. We apply it to Task 1a: predicting if the text would be considered humorous (for an average user), and Task 1c: if the text is classed as humorous, deciding if the humor rating would be considered controversial.

The Perceptron Algorithm is implemented with parameter averaging, based on the method described in Ha Daume's book [9].

The model is tested with 1, 10, 50, and 100 iterations(Table 2) with the 1,000 validation data. The performance of Perceptron model does not improve significantly with interactions increased. Therefore, we choose 10 iteration and the results are as follows.

Table 2 Perceptron model performance for task 1a and task 1c

Iteration	Accuracy(task 1a)	F-1 Score(task 1a)	Accuracy(task 1c)	F-1 Score(task 1c)
1	0.774	0.872	0.501	0.668
10	0.774	0.872	0.523	0.687
50	0.781	0.877	0.500	0.666
100	0.781	0.877	0.521	0.684

We run the model and get the classification results for the 1,000 unlabeled sentences, and upload the output to CodaLab for testing scores. For task 1a, the test set accuracy and F-1 score are 0.8110 and 0.8464. For task 1c, the test set accuracy and F-1 score are 0.5154 and 0.4440.

Perceptron model performs not good enough in task 1a, and even bad in Task 1c, it could be caused by these reasons:

1. Limitations of Perceptron Algorithm: it could work only with linearly separable classes. However, binary labels in our dataset could be not linearly separable.
2. Sentences are short, and number of training data are not enough. So there is not enough words to calculate weights and train a good Perceptron model.
3. The original data does not have pattern of whether a joke is offensive or not, humans have different interpretation, making the data sparse and difficult to classify.

7 Results and Discussions(By Liuyi Jin, Tian Liu)

7.1 Comparison

In this section, we describe the performance comparison between ColBERT and other baseline models. Specifically, we look at 3 BERT variants and 2 baseline models. As described in Section 6.2, the 2 ColBERT variants are developed from BERT-Large-Uncased and BERT-Base-Uncased pretrained model, respectively. The ALBERT is actually a compressed version of BERT-Base-Uncased pretrained model(6.4). Naïve Bayes model(6.5) and Perceptron model (6.6). From Section 2, we know Task1a and Task1c are classification tasks, so we calculate accuracy and F1 score as the metric to evaluate these 2 tasks. Task1b and Task2a are regression tasks, we compute root mean square error (RMSE) as the metric to evaluate the model performance. Note that we have not implemented regression with Naïve Bayes and Perceptron algorithms. That's why we see only ColBERT and ALBERT regression performance in Figure 3 and Figure 7.1.

Figure 2 and Figure 4 show the 3 BERT-based models performance compared to the 2 baseline models on Task-1a and Task-1c, respectively. We see on Task-1a, ColBERT based on BERT-Large-Uncased exhibit superior accuracy and F1-score performance. This indicates intrinsic potential of ColBERT architecture to capture humor structure in the given test. Also, ColBERT based on BERT-Large-Uncased has the highest F1-score. As for Task-1b and Task-2a performance shown in Figure 3 and Figure 5, we see the lowest RMSE given by the ColBERT. In particular, ColBERT based on BERT-Large-Uncased pretrained model reduces RMSE to half of that given by ALBERT.

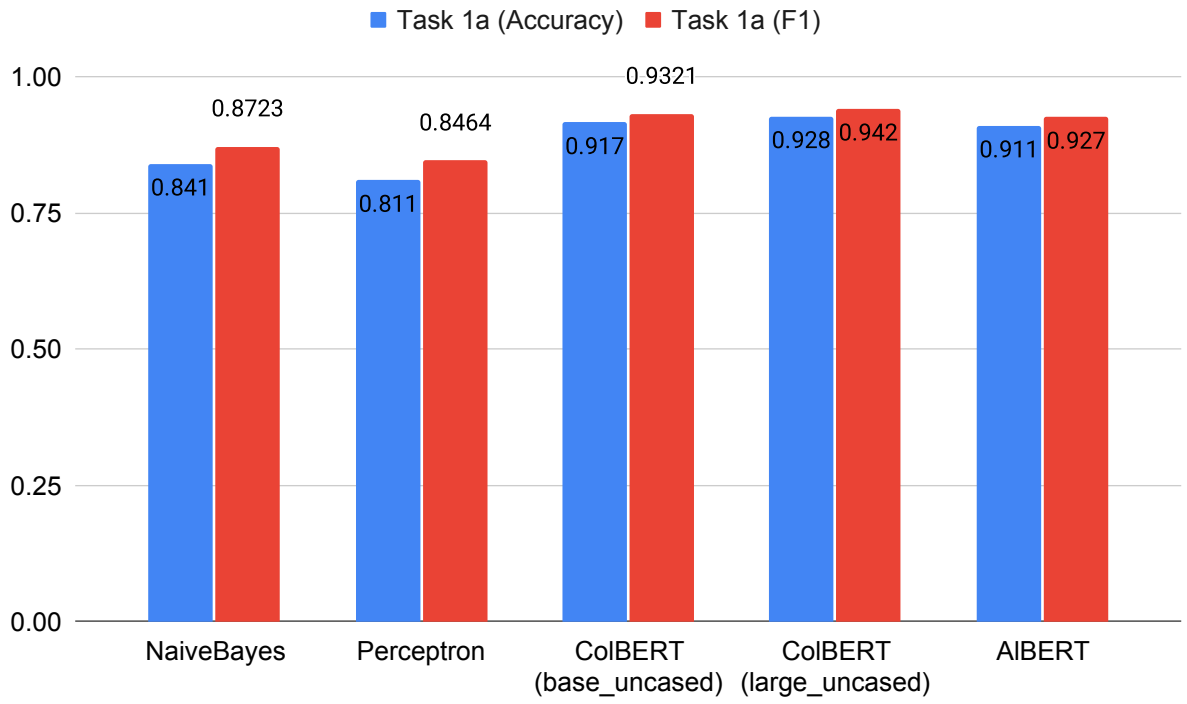

Figure 2: ColBERT Performance Comparison with Baseline Models on Task-1a.(By Ming-Hsien Chien)

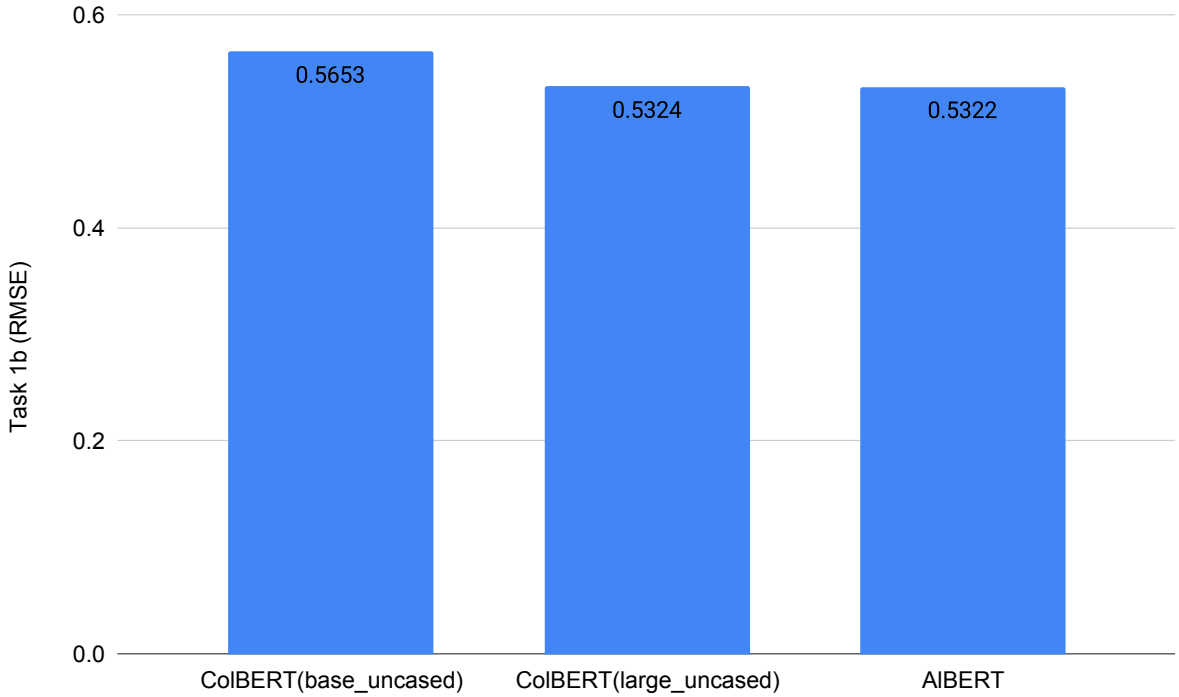

Figure 3: ColBERT Performance Comparison with Baseline Models on Task-1b.(By Ming-Hsien Chien)

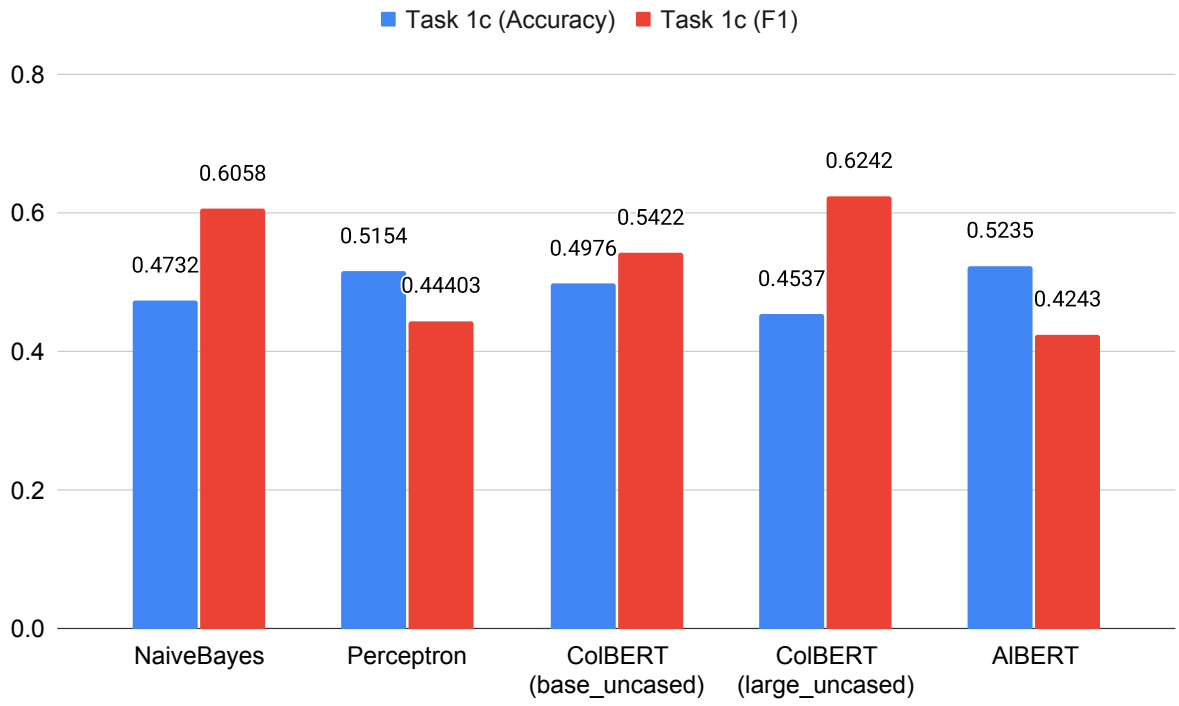

Figure 4: ColBERT Performance Comparison with Baseline Models on Task-1c.(By Ming-Hsien Chien)

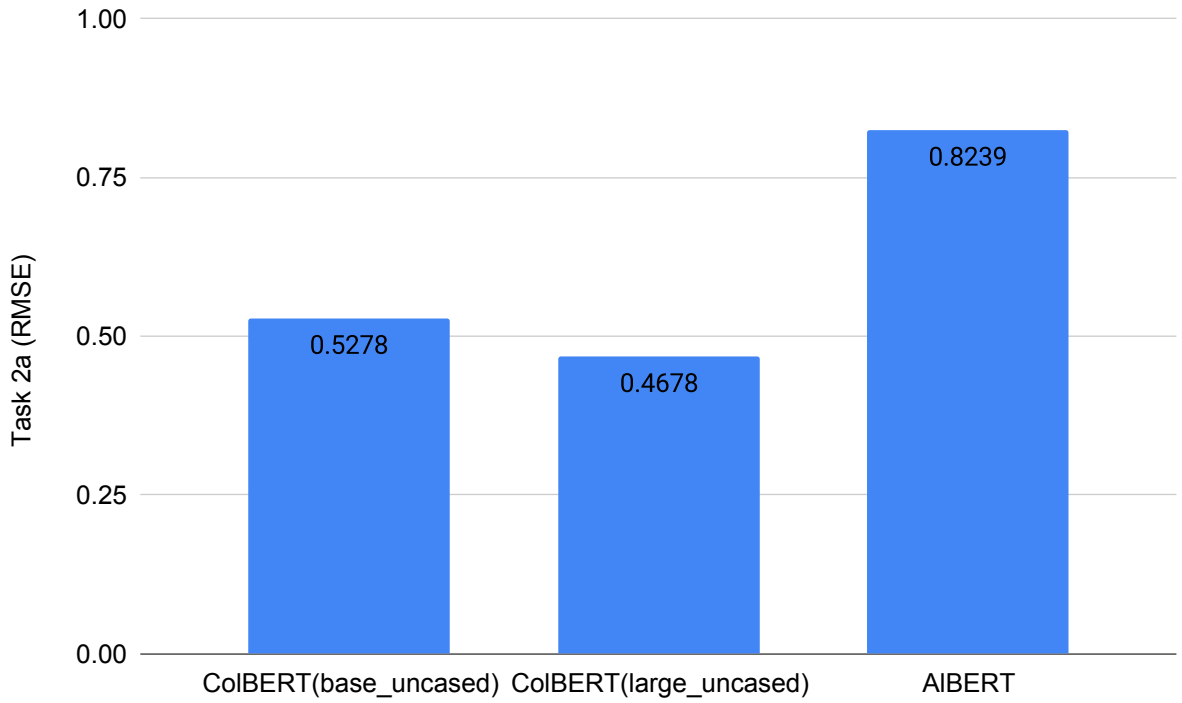

Figure 5: ColBERT Performance Comparison with Baseline Models on Task-2a.(By Ming-Hsien Chien)

7.2 Analysis and Discussion

In this project, we developed ColBERT based on 3 BERT pertrained model variants: BERT-Large-Uncased, BERT-Base-Uncased, and ALBERT. Our original assumption is that BERT Large can capture more semantic structures between sequential tokens and should achieve much better performance. However, in Figure 2 and Figure 4, we did not see much performance improvement. ColBERT based on BERT-Large-Uncased has little performance gains as observed from the accuracy and F1-score figures atop bars in Figure 2-5. What's worse, as a result of overfitting, ColBERT based on BERT-Large-Uncased has lower accuracy performance in Task-1c. This is because the large number of parameters in BERT-Large-Uncased pretrained model and limited data can easily lead to overfitting. Even after we did data augmentation, the overfitting problem remains for task 1c. Besides, considering the fact that ColBERT based on BERT-Base-Uncased cost much less training time than BERT-Large-Uncased model, we can employ the simpler ColBERT model as its performance is meeting the user needs.

A remarkable classification result performed by Naïve Bayes in Figure 2 and Figure 4 should be noted. Although Naïve Bayes cannot achieve ColBERT's performance, it's still comparable to ALBERT. Considering its near-zero training time compared to ALBERT, Naïve Bayes has a high potential in solving simple classification problems.

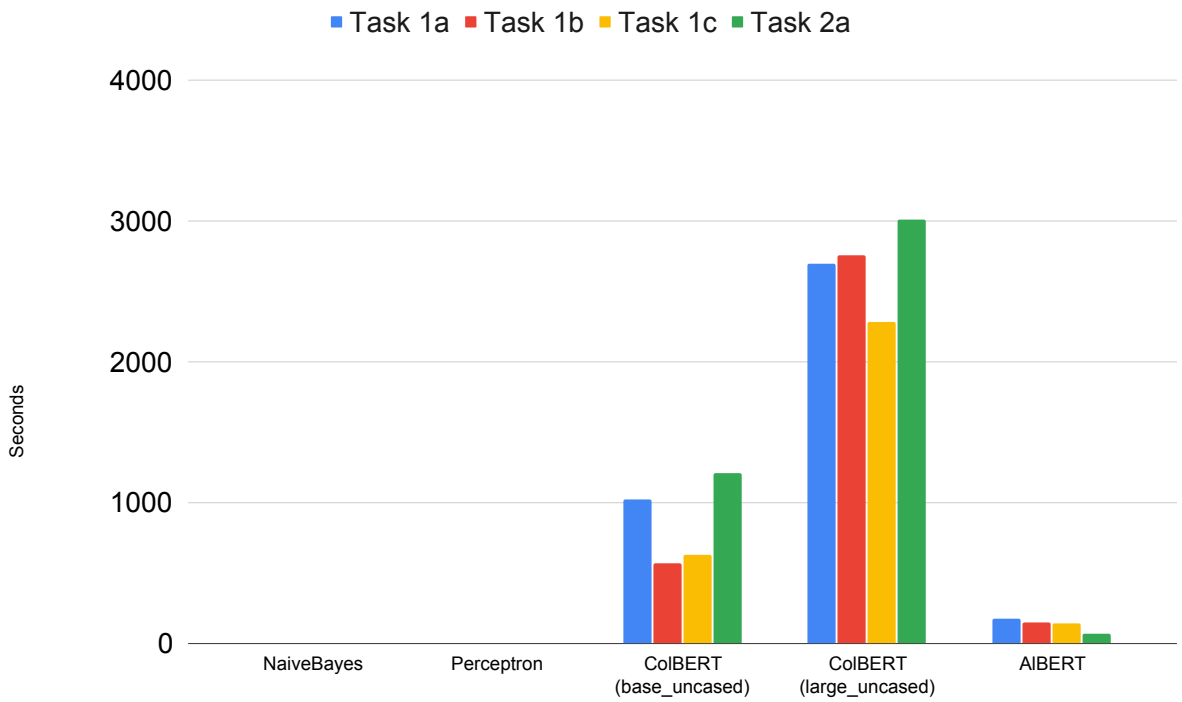

Figure 6: Training time for 8000 training texts. Y-axis is the training time in seconds, the lower the better.(By Ming-Hsien Chien)

7.3 2021 SemEval Submission Results

In this section, we look into the performance of ColBERT based on BERT-Large-Uncased when submitted to the contest. In Figure 7-10, we see we rank 28, 4, 5 and 10 globally for task 1a, 1b, 1c and 2a, respectively.

Task 1a Humor Detection

#	User	Entries	Date of Last Entry	Team Name	F-Score ▲	Accuracy ▲
1	DeepBlueAI	6	02/01/21	DeepBlueAI	0.9676 (1)	0.9600 (1)
2	dalya	32	02/02/21	SarcasmDet	0.9675 (2)	0.9600 (1)
3	ThisIstheEnd	14	02/03/21	EndTimes	0.9655 (3)	0.9570 (2)
26	mayukh	10	02/01/21	YoungSheldon	0.9468 (25)	0.9330 (18)
27	tmarchitan	13	02/07/21	Unibuc_NLP	0.9434 (26)	0.9290 (19)
28	TLIU	8	04/16/21		0.9421 (27)	0.9280 (20)
29	MihaiSamson	8	02/01/21		0.9374 (28)	0.9220 (21)
30	Zehao_Liu	5	02/19/21	UoR	0.9366 (29)	0.9210 (22)

Figure 7: Ranking for task 1a

Task 1b Average Humor Score

#	User	Entries	Date of Last Entry	Team Name	RMSE ▲
1	mmmm	6	02/01/21		0.5003 (1)
2	mayukh	10	02/01/21	YoungSheldon	0.5257 (2)
3	fdabek	1	02/01/21	Amobee	0.5271 (3)
4	TLIU	8	04/16/21		0.5324 (4)

Figure 8: Ranking for task 1b

Task 1c Humor Controversy

#	User	Entries	Date of Last Entry	Team Name	F-Score ▲	Accuracy ▲
1	CHAOYUDENG	111	04/20/21		0.6299 (1)	0.5089 (13)
2	dalya	32	02/02/21	SarcasmDet	0.6270 (2)	0.4699 (20)
3	ThisIstheEnd	14	02/03/21	EndTimes	0.6249 (3)	0.4553 (24)
4	mmmm	6	02/01/21		0.6247 (4)	0.4569 (23)
5	TLIU	8	04/16/21		0.6242 (5)	0.4537 (25)

Figure 9: Ranking for task 1c

Task 2 Average Offensiveness Score

#	User	Entries	Date of Last Entry	Team Name	RMSE ▲
1	DeepBlueAI	6	02/01/21	DeepBlueAI	0.4120 (1)
2	mmmm	6	02/01/21		0.4197 (2)
3	fdabek	1	02/01/21	Amobee	0.4406 (3)
4	stevenhuahua	2	02/21/21		0.4454 (4)
5	emran	1	02/09/21	ES-JUST	0.4467 (5)
6	dalya	32	02/02/21	SarcasmDet	0.4469 (6)
7	reynier	21	02/01/21	RoMa	0.4532 (7)
8	calamitylink	4	02/01/21	HumorHunter	0.4546 (8)
9	Amherst685	9	02/01/21		0.4564 (9)
10	TLIU	8	04/16/21		0.4678 (10)

Figure 10: Ranking for task 2a

8 Conclusions (By Yijun Zhang, Tian Liu, Ming-Hsien Chien)

The project is first task to combine the humor and offense detection and we implemented four approaches to solve the problem. In addition, we found that ColBERT-LARGE has the best accuracy and lowest RMSE, but requires more training time.

After uploading our predict result to the online competition platform, we achieved top rankings globally. Especially, we obtained the 4th, 5th and 10th in the task1-b, taks1-c and task2.

In the result, it indicated that humor detection (task 1a) is easy while deciding humor controversy(task 1c) is a challenging task. It could be caused by the ambiguity of whether a potentially offensive joke and even two human beings may have opposite interpretations of a joke.

References

[1] J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, and Walid Magdy. Hahackathon: Detecting and rating humor and offense. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

[2] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376, 2015.

[3] Dario Bertero and Pascale Fung. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 130–135, 2016.

[4] Peng-Yu Chen and Von-Wun Soo. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117, 2018.

[5] Issa Annamoradnejad. Colbert: Using bert sentence embedding for humor detection. *arXiv preprint arXiv:2004.12765*, 2020.

[6] Tian Wang and Yuyangzi Fu. Item-based collaborative filtering with bert. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*, pages 54–58, 2020.

[7] Zixiaofan Yang, Bingyan Hu, and Julia Hirschberg. Predicting humor by learning from time-aligned comments. In *INTERSPEECH*, pages 496–500, 2019.

[8] Victor Raskin. *Semantic mechanisms of humor*, volume 24. Springer Science & Business Media, 2012.

[9] Harold Charles Daume and Daniel Marcu. *Practical structured learning techniques for natural language processing*. Citeseer, 2006.