FRI NLP Project
Initial Report

Qualitative Research on Discussions - text categorization

Cristian Bassotto, Nikolay Kormushev and Ahmet Çalış

Abstract

Qualitative discourse analysis is an important way social scientists research human interaction. Large language models (LLMs) offer potential for tasks like qualitative discourse analysis, which demand a high level of inter-rater reliability among human "coders" (i.e., qualitative research categorizers). This is an exceedingly labor-intensive task, requiring human coders to fully understand the discussion context, consider each participant's perspective, and comprehend the sentence's associations with the previous discussion, as well as shared general knowledge. In this task, you create a model to categorize postings in online discussions, such as in a corpus — an online discussion about the story, "The Lady, or the Tiger?". We provide a coded dataset with a high inter-rater reliability and a codebook including definitions of each category with examples. Your task is building and training a highly reliable language model for this coding task that generalizes to other online discussions.

Keywords

NLP, LLM, Discourse analysis

Advisors: prof. Slavko

Introduction

Discourse can be defined as particular way of talking about and understanding the world (or an aspect of the world) [1]. On the other hand, qualititive discourse analysis is an attempt to understand how meaning is formed. While it traditionally has been conducted manually by linguists and researchers, the increasing volume and complexity of textual data call for automated solutions to expedite the process and enable analysis at scale as mentioned here [2].

With the rise of LLMs many solutions appeared that fine-tune these models for specific use cases. This requires a programmer to sit down and understand the domain for which he needs to fine-tune the model. Even if we just take into account only the read time this process can be quite time consuming [3] and with this paper we aim to develop a solution that can generalize and generate discourse analysis on multiple domains.

To achieve our goal of classification we will try three approaches: topic modelling for classification, transformers-based models and prompt engeneering using Mistralis LLM. By comparing these approaches, we seek to identify the method that generalizes the best. For us it is also important our analysis to be explainable and so we use an LLM for explaining our results.

Related Works

In the field of text categorization, researchers have explored various methods to better understand and classify written data. One key study [4] looked at different ways to classify texts using supervised machine learning techniques like Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine, and AdaBoost. These approaches represent the state of the art in text categorization and have historically served as foundational pillars in the field.

However, it's important to note that while these traditional methods remain relevant and are still utilized in various contexts, emerging techniques have demonstrated superior performance in certain scenarios. Another important review [5] surveyed the techniques used for text classification across different areas and over the years. They classified documents based on their content and language, using the models that we already discussed, but reaching also the most recent transformers-based models.

Recent research has focused on advanced models like BERT (Bidirectional Encoder Representations from Transformers). These models have shown promise in improving text classification accuracy. For example, one study compared BERT to another model called XLNet to see which one worked better for text classification tasks [6].

In the past few years the focuse changed to fine-tuning these models by either adding extra layers or adjusting hyperparameters and the corpus used. Despite these advancements, there have been instances where research didn't result in better outcomes. For example, in some cases, adding more layers actually led to poorer performance in specific studies [7]. It's important to note that there isn't a one-size-fits-all solution in text categorization. What works well for one dataset may not work as effectively for another. This study [8] focused on this and tried to better find specific BERT-based models for different specifics dataset, finding out that the best model is not always the most complex one.

Research in text categorization is constantly evolving, with new methods and technologies being developed to better understand and classify written texts. New approches focused on Large Language Models (LLMs) such as GPT-3, but according to the following research [9] their performances still significantly underperform fine-tuned models in this task. In this paper we also want to compare this different approches.

Data exploration

The input dataset contains gathered information from students chats. These chats show students discussing a topic and writing together a conclusion. They were collected from a student application used for collaboration CREW and alternative one that is just a normal chat. The dataset was used to compare the two to determine which improved collaboration between students.

The dataset is very chaotic. It has duplicate and misslabeled columns. A lot of unneeded information. Labeling from 2 different people which also often use different formats. In this section we will describe it in detail.

Overview

Lady and the Tiger dataset

The dataset has **609** samples and **25** features. We can see the features in Table 1.

There are duplicate features like Pseudonym, Message and Bookclub. We verified that their values are identical. Also we see some repeating columns. Some include R1 in their names and other R2 which represent the annotators. We were told to only trust R2 in the instructions of use. Their annotations also in some cases like DialogicSpell used different formats. Nevertheless we decided to measure agreement between the two with Cohen's kappa score the results can be seen Table 4. We conclude that excluding the question category the two annotators did not agree very much. Therefore we choose to follow the instructions and just use R2's annotations.

Labels

We have been given a codebook with taxonomies for all the classes we have for the messages like question, discussion type, pivot etc. Here we will look at the distribution of these classes. The plots can be seen in Figure 1

DiscussionType

From the discussion type distribution we can see most samples are a Seminar which is to be expected since students are discussing the book they are reading.

We can see some messages have multiple classes as well. Those messages ended up being a combination of two sentences each with a different class e.g. "Looks good! I guess we can complete the post survey and submit our assignment" where "Looks good!" is considered social and the rest is Deliberation.

Since these are single samples we will focus our efforts on single label classification by dividing the message into two messages each with 1 sentence and a different class.

Question

From the question plot we see once again multiple categories. Again those are cases of multiple sentences in most cases where we divide them into two separate samples. On one of the categories there is a '?' where I assume the annotator was unsure of how to categorise it. Here I replaced the label based on my subjective categorisation which coincides with R1's opinion. Also O-COT is a non existing category and again I annotated it based on R1 and my opinion.

Uptake

In the uptake plot we can see one of the categories from the codebook(Prompt) has 0 samples. Disagree also has quite a few samples.

Pivot

The pivot distribution shows us this type has a lot of variants. For all we have 10 samples or less which makes them very hard to predict. There are also multiple spelling errors in some of the categories.

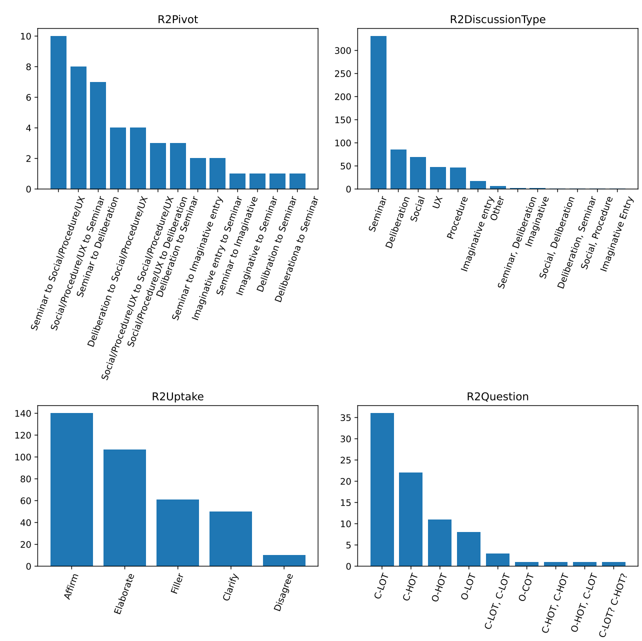

Figure 1. Classes distribution

Table 1. Feature Names

Feature Name	Description
Course	The name of the course
BookID	Confusingly named ID since the book is only one. 260 or 261 if CREW is used or not
Topic	The topic under discsussion.
Bookclub	The id of the group of users participating in a discussion
Chat0CREW1	Was a normal chat - 0 or the CREW application - 1 used
Pseudonym	Nickname of user
Message	The text message
MessageTime	When was the message sent
IsAnswer	A string with value Yes or No
Page	A number for the page of the book
ResponseNumber	The number of the response gotten. Saved in another table
R1DiscussionType	string representing discussion type based on code book
R2DiscussionType	string representing discussion type based on code book
R2DiscussionTypeInterpNothers	Same as discussion type but some types are removed and others renamed
CollapsR2DiscussionTypeInterpNothers	Discussion Type but with Social Procedure and UX types collapsed into 1 type
Chat0CREW1B	Chat0Crew1 but binary. Since both are 0 and 1 they have the same value
R1DialogicSpell	'Start' or 'End' showing start or end of a dialogic spell. Sometimes is a number
BinaryR1DialogicSpell	0 - out of spell or 1 - in spell
R1Uptake	The uptake classification of the message
BinaryR1Uptake	0 - not uptake or 1 - uptake
R2DialogicSpell	Number identifying a dialogic spell and which messages are in it
BinaryR2DialogicSpell	0 - out of spell, 1 - in spell
R2Uptake	The uptake classification of the message
BinaryR2Uptake	0 - not uptake or 1 - uptake
Pseudonym.1	duplicate column
Message.1	duplicate column
Bookclub.1	duplicate column
R1Question	type of question classification
R2Question	type of question classification
R1Pivot	shows if we have a change of DiscussionType
R2Pivot	shows if we have a change of DiscussionType
Memo	Notes on the dataset
OldCodeBook	labels based on old taxonomy for discussion type

Table 2. Cohen's Kappa Scores Lady and The Tiger

Feature Name	Kappa Score
Question	0.7568
Pivot	0.1937
Uptake	0.2753
Discussion Type	0.2985

Climate change dataset

The climate change dataset has **288** samples and **31** features. There were no provided instructions for this dataset. During exploration, we noticed that there were 4 columns for each target feature, consistently starting with **SS**, **MG**, **ZH**, and **NT**, which we assumed to be different annotators. Since there were no instructions on which labels were more reliable, unlike the Lady and Tiger dataset, we decided to take the majority vote among these 4 annotators. Labels are only provided for **DiscussionType** and **Uptake** and **DialogicSpell** is partly labelled.

After replacing each target feature with the majority vote, the following features were the only features missing in climate change dataset compared to the Lady and Tiger dataset: **ChatCrew**, **Course**, and **ResponseNumber**. Table 3 shows the Cohen's Kappa scores for each pair of annotators (excluding NT due to full null values). We observed that annotators showed minimal agreement on the Uptake class and moderate agreement on Discussion Type. These results supported our decision to use the majority vote instead of just picking one annotator's decision.

Labels

The only labels provided were for Uptake and Discussion Type unlike in the Lady and the Tiger.

Table 3. Cohen's Kappa Scores Climate Change

Annotators	Uptake	Discussion Type
MG - SS	0.2277	0.5568
MG - ZH	0.2745	0.4302
SS - ZH	0.2955	0.5459

Discussion Type

Most samples belong to the Deliberation and Seminar classes. The other classes have a very low number of samples, making the learning task harder.

Uptake

Unlike the Lady and Tiger dataset, the Uptake class has samples in every category. Most samples are part of the Clarify and Elaborate categories, while Disagree and None have the fewest samples.

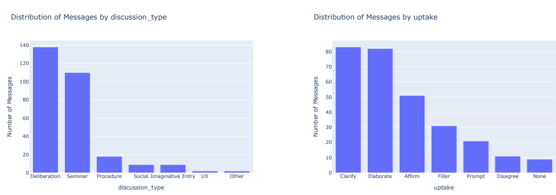

Figure 2. Discussion Type and Uptake Distributions

Optimal Transport Dataset Distance (OTDD)

OTDD measures the "distance" or dissimilarity between two datasets by computing the optimal transport cost required to transform one dataset into another. Lower values indicate that the datasets are more similar, while higher values indicate greater dissimilarity.

We calculated distance between the Lady and Tiger dataset and Climate Change dataset for Uptake and Discussion Type classes.

Table 4. Optimal Transport Dataset Distance

Feature Name	OTDD
Discussion Type	0.5095
Uptake	0.5094

Since both distances are around 0.5, we can consider that the two datasets are similar for both comparison.

Methods

General cleanup

Lady and the Tiger dataset

We removed all duplicate columns like Message by first verifying one column is not empty while the other filled. If that was the case the values were merged into the empty column. We also used only R2 as the label column.

Label cleaning

Lady and the Tiger dataset

We needed to clean the classification labels, because they contained errors.

- *Multilabel* cases: Some of the columns had multiple labels like R2DiscussionType and R2Question. For those we either reduced it to one label based on R1s classification or divided the sample by hand into two that have a single class. This was only done during training for predicting that label.

- *Misspelled* cases: A lot of cases had multiple names e.g. Imaginative entry and Imaginative Entry are separate labels for R2DiscussionType. For those cases we unified the classes.

- *Pivot*: The pivot class has very few labeled samples. This might cause issues for training and we will probably need to use simpler models. 521 entries are not labeled and after only 17 of the top class are labeled.

XGBoost with FastText

After the cleaning, in order to use **XGBoost Classifier** (or any other classic method), we needed to embed the messages and encode the rest of the categories. To do so, we perform some preprocessing steps to clean the messages in order to embedded them after.

In particular, we remove some *stopwords*, *tokenize* and *lowercase* the tokens. Also we remove tokens like emoticons and punctuation that are not alphanumeric.

After this process, we use **FastText** to embed the processed messages because we assume that a lot of words are misspelled since they are written by children and FastText performs better in this case.

We transform also the other labels using *LabelEncoder* for pseudonym and course (even though this will not generalize good after), we extract the information about the *time* in different columns and fill the *NaN* values with 0.

After this process, we *split* the data for training and testing, maintaining the distribution of the final labels.

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art pre-trained language model that uses transformers to understand word context bidirecationally. It was developed by Google and revolutionized natural language processing tasks.

DeBERTa (Zero Shot Learning)

Zero-shot text classification is a task in natural language processing where a model is trained on a set of labeled examples but can classify new examples from previously unseen classes. We used DeBERTa for this approach to predict labels of Discussion Type and Uptake.

Finetuning

For finetuning we used BERT and added an extra fully connected layer with a ReLU that acted as a classifier. The loss we used is cross-entropy since we are doing multiclass classification and we tried to use also to weigh the classes so the classes with less samples are also predicted and not only the most common ones.

DiscussionType

For the DiscussionType we tried feeding in only the message to the model. Also the message with the DiscussionTypes of the previous 3 utterances because they often indicate the next one. Seminars often come after seminars or maybe lead to deliberation etc. Also the pseudonyms of the users were added to the DicussionTypes because maybe they would indicate a user changing topic or something like that. I was worried we might overfit on some users and if they often bring up social topics but so far it seems to generalize well.

Uptake

For the Uptake we tried adding the DiscussionType of the current utterance, also the Uptake of the previous utterances if it is available again hoping that for example a Disagree and Affirm maybe are related if 2 people disagree with the same statement. If not Uptake is missing we use an empty string. As another test we also added the pseudonym and the DiscussionType of the previous utterances. Lastly we tried also adding the page.

Question

We divide this into a few models. One says if the message is a question or not. After a second model decides if we are HOT or LOT. It is trained only on questions data and expects only questions. The third model is resposible for figuring out if it is closed(C) or open(O).

Finetuned LLMs

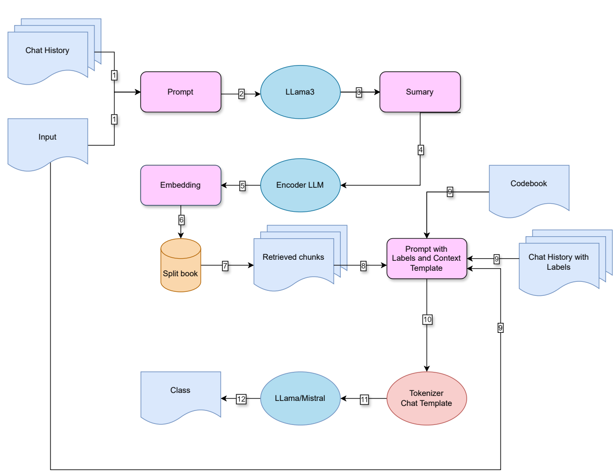

Figure 3. LLM Pipeline

In this section we will look at our LLM pipeline displayed in Figure 3. From this point on due to time constraints we will focus more on studying different approches on DiscussionType and Uptake.

Prompts

For the prompt we create a generic prompt that can be used both with LLama and Mistral. Then we use the respective tokenizers apply_chat_templates method to generate the final prompt optimized for the models.

As seen in the diagrame the final prompt contains the codebook with the definitions of the labels and examples and the previous history and their labels formatted as a chat with our model. The prompt also contains as context two sections of the book that the students are discussing extracted using RAG. For exact steps refer to 1-8 in Figure 3.

When writing our prompt we also take care to tell the model what to do and not what to not do since models do not work well with negations.

Here is an example of the final prompt for LLama. The main difference between this and the prompt for Mistral are the meta tags given by the template:

Listing 1. Example Prompt

```
<|begin_of_text|><|start_header_id|>system<|
    end_header_id|>

You are an AI expert in categorizing sentences
    into classes.
Another assistant has retrieved some documents
    that might be useful for you to understand the
     context of the conversation, do not use them
    if not relevant.
page\_content="<section from book>"
page\_content='<section from book>'
You can use the following codebook (with classes,
    definitions and examples) to help you
    categorize the sentence:
### IMPORTANT CODEBOOK:
### Class: Affirm ###
### Definition: The action or process of affirming
     something or being affirmed, showing
    agreement. ###
### Example: Thats a good point. I agree. ###
<More examples here>
###
You need to categorize the new sentence into one
    of the following classes: [<classes>].
If you fail to categorize the sentence, return '
    None' instead of coming up with a wrong class.

You can use the history of the conversation to
    help you categorize the sentence.[eos]<|
    start_header_id|>user<|end_header_id|>

I'm not sure how to word it, but I think their
    barbaric nature made them irrational and maybe
     incapable of love? Something like that...[eos
    ]<|start_header_id|>assistant<|end_header_id|>

Class: Disagree[eos]<|start_header_id|>user<|
    end_header_id|>

I like it! Good sentemce.[eos]
```

LLM Models

For the LLM models we finetuned LlamaForSequenceClassification, LlamaForCausalLM and MistralForCausalLM. The first model is finetuned on a classification task, while the latter two are given the task of text completion and are given not only the inputs and history but also the label of the input.

Lora Finetuning

To finetune the three models we used the LoRA technique in order to reduce time and costs of training and memory. Moreover, LoRA feets nicely with our task, since we can change adapter without having to reload the model if we want to change from one category to another.

Ensemble

Once we had the results from all our models we decided to use them as an ensemble with few-shot learning. As input we used our LLM prompts and we modified them to add the prediction of the other models and make it choose one of the predictions given to it which are from XGBoost, BERT, LlamaForClassification and MistralCausalLM.

Explanation

With the development of AI explainability of models has become a huge topic. It is important that our non-expert lingiust users can understand why a specific category was assigned to that specific input. To make it as easy as possible to use we generate textual explanations using LLama and our original prompt modify adding the final prediction and a question that ask for further explanation.

An LLM for explainability can enrich the experience of our end users by making the results much more understandale but it can also confuse them because it is suceptible to hallucinations [10]. To avoid this we ask the model to refer to input prompt when explaining its decision.

Results

Evaluation

To asses our models we will compare their Precision, Recall, Accuracy and F1 scores. While training we focused on improving mainly F1 to account for both Precision and Recall. For clarity all our metrics represent the weighted average for the model.

Discussion type

Among the models we tested, LlamaForSequenceClassification did the best overall with an F1-score of around 0.9. This is reasonable since the model is trained for this task and has the most number of parameters over all the others. MistralForCausalLM also reach some good result in a task that is not related to the one that it is trained on, showing it's potential in generalization over some other models that are specific for this task like BertForClassification or XGBoost. Zero-shot DeBERTa performs the worst, enfatizing the importance of training with respect to common knowledge.

Table 5. Model Performance Discussion Type

Model	Precision	Recall	F1	Accuracy
LlamaForSequenceClassification	0.90	0.90	0.89	0.90
MistralForCausalLM	0.81	0.80	0.81	0.80
Ensemble	0.75	0.77	0.74	0.77
BERT	0.64	0.69	0.65	0.69
XGBoost	0.57	0.61	0.59	0.61
DeBERTa (Zero shot)	0.35	0.13	0.11	0.13

Uptake

In the results we can see that BERT is outperforming LlamaForSequenceClassification. This is likely due to the fact we did not have time to finetune the model due to a bug we had. As a result it is likely possible to get much better results. On the other hand, MistralForCausalLM performs a bit better than BERT, while also DeBERTa performs better than before. This means that for prediction Uptake, common knowledge has a positive impact in the performance because probably the labels are universally defined and understandable.

Table 6. Model Performance Uptake

Model	Precision	Recall	F1	Accuracy
MistralForCausalLM	0.67	0.66	0.66	0.66
BERT	0.66	0.65	0.65	0.64
LlamaForSequenceClassification	0.61	0.58	0.57	0.58
Ensemble	0.44	0.13	0.16	0.13
XGBoost	0.53	0.52	0.53	0.52
DeBERTa (Zero shot)	0.26	0.24	0.21	0.22

Question

For Question we managed to build a model that recognises if the message is a question or not with 99% accuracy so we tested and trained these models to just recognise the type of question.

It is interesting to note that XGBoost has performed better than BERT as seen in Table 7. We believe this to be due to the smaller sample size.

Table 7. Model Performance Question

Model	Precision	Recall	F1	Accuracy
LlamaForSequenceClassification	0.65	0.56	0.65	0.56
MistralForCausalLM	0.47	0.47	0.47	0.47
XGBoost	0.47	0.41	0.47	0.43
BERT	0.35	0.61	0.35	0.36

The context is usefull?

We wanted to know also if the context that we decided to add to the prompt for the LLMs was actually usefull or just noise added to the input.

On the Table 8, we can compare various results with or without these features. In particular:

- History refers to the previous messages from other users.

- Past Labels is available only with the history because refer to the predicted class of the past messages (that can be seen also as an improvement of the few shots technique).
- Context refers to the documents retrieved by RAG.
- Pseudonym is an extra that we decide to try. It adds the pseudonym of the author in front of the message to distinguish better the flow of the conversation.

Table 8. Model F1-score based on context added

History	Past labels	Context	Pseudonym	F1-score
x	x	x	x	0.79
x	x	x		0.78
x	x		x	0.80
x	x			0.76
x		x	x	0.75
x		x		0.72
x			x	0.73
x				0.76
		x	x	0.74
		x		0.73
			x	0.81
				0.59

The results were quite difficult to interprete since they were very similar. However, we can say that in general, adding more information does not descrease the performances, but on average it increase them. In our case, the context seems to create some issues, probably because we only had the story of one dataset and not of the second one, so the retriever could not find related documents for the article dataset. In future, the context could be more relevant adding all the story or documents from the IMAPBook and could be able to actually bring an improvement on prediction.

On the other hand the pseudonym seems to have a good impact in the score. In this case, probably the transformer is able to catch the flow of the conversation and retrieve the total message of the current user that might have been splitted when sent. Only knowing the pseudonym seems to increase the performances in prediction, but this is probably just an overfitting result that probably is taking more in consideration the name of the author over his message. We made this conclusion also because, using early stopping, the model actually train for over than 5 epochs, while the prompts with more context train only for 1 to 3 epochs.

We also notice that with more context the model learns faster and overfit faster (after 2 epochs with all context vs after 4 with less context).

Explanations

For the explanation results refer to Listing 2 and 3 which show the prompt we used with the model and the response we got. For brevity the tokens of the prompt are ommited. We can see that the explanation is pretty good. The model is figuring out that the text is about a narrative and gives a good explanation as to why it considers it a seminar. When we allowed for longer responses it also explained why it is not the other classes with good reasoning.

We tested this only by hand with a few prompts and the responses were overall good. Neverthless sometimes when we made them shorter they could be vague.

Listing 2. Prompt

```
You are an AI expert in categorizing sentences
    into classes.
You can use the following codebook (with classes,
    definitions and examples) to help you
    categorize the sentence:

<original prompt with codebook and without the
    context from RAG>

Hi Teresa Stewart, I felt that if he truly knew
    her, he would know her true character and her
    barbaric side. The story also noted that she
    had seen her lover talk to the lady before,
    and assumed that her jealousy would be known
    to him prior to his tribunal.

Class: Seminar

Based on the previous chat, can you explain to me
    why you have chosen this class in a two
    sentences? Use at most one hundred words.
    Please reference parts of the input message
    when explaining your decision.
```

Listing 3. Response

```
The reason for choosing "Seminar" classification
    lies in phrases such as "...he would know her
    true character...", which indicate analysis
    and consideration given by the speaker towards
     understanding characters within the narrative
     context. Additionally, statements made
    regarding assumptions ("her jealousies")
    suggest interpretive discussions between
    individuals sharing their perspectives upon
    examining specific events from various angles
    - characteristic features often found under
    the umbrella term "Seminars".
```

Conclusion

Our conclusions are focused on the new techniques taken into account. In particular, LLamaForSequenceClassification seems to outperforms the other models. This is understandable since it has more parameters (8 bilion) and it's trianed on this specific task. A curious result is instead given by MistralForCausalLM that even in a task on that it is not trained on, seems to generalise well, aciving better results than specific native models that we used as baselines.

Mistral and LLama are also usefull as easy to use ensembles and to generate good explanations over an already made prediction.

References

[1] James Paul Gee. *An introduction to discourse analysis: Theory and method*. routledge, 2014.

[2] Thomas Jacobs and Robin Tschötschel. Topic models meet discourse analysis: a quantitative tool for a qualitative approach. *International Journal of Social Research Methodology*, 22(5):469–485, 2019.

[3] Rolf A Zwaan, Gabriel A Radvansky, Amy E Hilliard, and Jacqueline M Curiel. Constructing multidimensional situation models during reading. *Scientific studies of reading*, 2(3):199–220, 1998.

[4] Ishaan Dawar, Narendra Kumar, Sakshi Negi, Sayeedakhanum Pathan, and Shirshendu Layek. Text categorization using supervised machine learning techniques. In *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, pages 185–190. IEEE, 2023.

[5] Enock Osoro Omayio, Indu Sreedevi, and Jeebananda Panda. Language-based text categorization: A survey. *Digital Techniques for Heritage Presentation and Preservation*, pages 11–36, 2021.

[6] Parsa Sai Tejaswi, Saranam Venkata Amruth, Prakya Tummala, and M Suneetha. Automatic documents categorization using nlp. In *ICT Infrastructure and Computing: Proceedings of ICT4SD 2022*, pages 215–225. Springer, 2022.

[7] Samin Mohammadi and Mathieu Chapon. Investigating the performance of fine-tuned text classification models based-on bert. In *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1252–1257. IEEE, 2020.

[8] Rukhma Qasim, Waqas Haider Bangyal, Mohammed A Alqarni, and Abdulwahab Ali Almazroi. A fine-tuned bert-based transfer learning approach for text classification. *Journal of healthcare engineering*, 2022, 2022.

[9] Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text classification via large language models. *arXiv preprint arXiv:2305.08377*, 2023.

[10] Patrick Huber and Giuseppe Carenini. Towards understanding large-scale discourse structures in pre-trained and fine-tuned language models. *arXiv preprint arXiv:2204.04289*, 2022.