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.

Initial Ideas

Preprocessing

General

In our investigation, we intend to explore diverse preprocessing techniques and embeddings to evaluate their impact on classification performance. Alongside classic methods such as word2vec embeddings, we will examine alternatives including BERT embeddings and fastText. Additionally, we will consider various preprocessing strategies such as tokenization, stemming, and lemmatization, aiming to optimize the quality and representational power of the input data for each embedding method.

Topic Modelling

Topic modelling deals with generating topics from a corpus of documents and then assigning them to a document. As a topic in this context we mean a set of key words that were extracted from the document. These topics can be generated using models like BERTopic [10] or LLMs [11]. In [12] we see that supplying the topic to the model used for classification increases the resulting accuracy so we wanted to try to see how big of an improvement we could get for both approaches. First we will generate the topics for our input text and then we supply it as part of the input to our models. In this regards different approches tried to study online discussions throw topic modeling and manual categorization [13] [14], but we want to implement an automated approach.

Models

BERT

In this study, we adopt a simple approach to automate discourse analysis by fine-tuning the BERT (Bidirectional Encoder Representations from Transformers) model on our dataset. By establishing this baseline, we aim to assess BERT's effectiveness in capturing discourse patterns and set a benchmark for evaluating alternative methodologies.

Prompt Engineering

Inspired by [15] we decided also to try to see how simple prompt engineering compares to classification with BERT. In this case we plan to use Mistralis and send it prompts asking it to classify our sentences.

In the topic modeling paper they achieved a much better performance comapred to solutions like BERTopic which shows it can be a valid approach. The benefits of this approach is that we do not require to fine-tune our model. We just run it and ask it questions. Althought this is a trade off because we need to find appropriate prompts to classify with a good accuracy. Here we need to be carefull since Mistralis is also trained on a bigger data corpus which means it is likely it will perform much better when generalising.

Evaluation

To evaluate how our model generalises we will scrape a small sample from forum sites like Reddit or Quora and evaluate the results ourselves and decided. This is not optimal but we do not have an already labeled dataset we can use for the task so the alternative is to create one which requires an expert and time which we do not have.

Explainability

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 plan to generate textual explanations using an LLM like Mistralis or ChatGPT by feeding it our inputs and outputs and asking to explain our models conclusions. Using an LLM for explainability can enrich the experience of our end users by making the results much more understandale but it can als mean it can also confuse them because it is suceptible to hallucinations [16]. Because of this we will ask the LLM to return a section of our input based on which it made its conclusions.

We are also planning to try to use topic modelling to aid explainability. By genereting the keywords from the text we classify we believe it can become clearer to the user why a classfication decision was made.

Methods

Data Exploration

The input 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 list some of the issues we found and what cleanup was done.

In Figure 1 we can see the word cloud of the messages, where more frequent messages are bigger.

In Figure 2 we can see the most active users based on the number of messages they send.

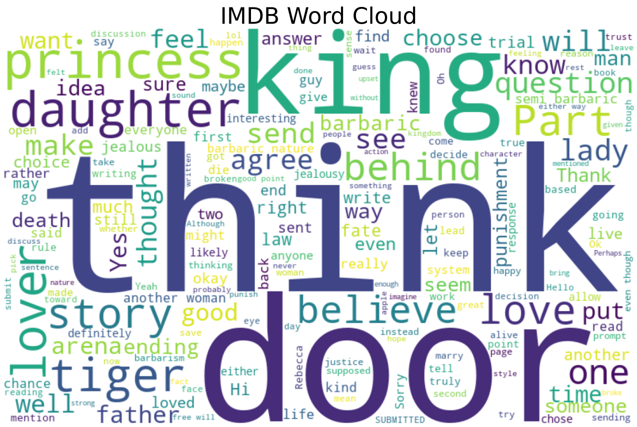

Figure 1. Message WordCloud

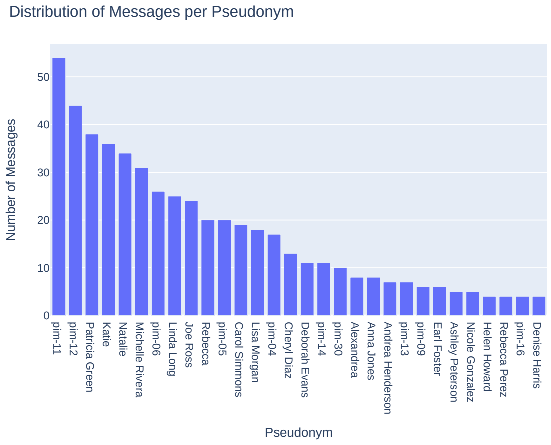

Figure 2. Message per User

General cleanup

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 have two labelings from R1 and R2. They follow sometimes different formats. Because of this and the task instructions we chose to trust R2 and just use his labeling. It happened that R1 had labelled columns that R2 had not. In those cases we trusted R1 was good and we used his labels to get more samples.

Some categories have only one unique label, so we decide to remove them because they gave no useful information about the result.

Label cleaning

We needed to clean also the labels to classify, because the expert made some errors or the data was not ready to be predicted.

- *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.

Finetuning

For finetunning 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

For now I am just feeding into the model the message and the DiscussionType. I believe that the type might be useful since it is more likely to have an open question that require higher order thought (O-HOT) during a seminar than a delibaration. This still needs to be verified.

Pivot

Not attempted with BERT due to the small sample size of each class. It would be hard to even assess the quality of the model since we have less than 30 samples in all classes which is statistically insignificant.

Results so far

Evaluation

To asses our results we used the F1 score so we can both asses Recall and Precision. In the tables below for now we listed all our results for all categories.

XGBoost

DiscussionType

As we can see on Table 1, with a classic method like XGboost we manage to reach only 0.72 of F1-score. This result is mostly given by the right prediction of the most frequent classes, while less frequent classes like *Other* or *Imaginative Entry* are never right predicted.

Class-wise Metrics				
Class	Precision	Recall	F1	Support
Deliberation	0.68	0.72	0.70	18
Imaginative Entry	0.00	0.00	0.00	4
Other	0.00	0.00	0.00	2
Procedure	0.71	0.56	0.62	9
Seminar	0.81	0.88	0.84	67
Social	0.64	0.64	0.64	14
UX	0.50	0.44	0.47	9
Overall Metrics				
Metric	Precision	Recall	F1	Support
Macro avg	0.48	0.46	0.47	123
Weighted avg	0.70	0.73	0.72	123
General	0.70	0.73	0.72	123
Accuracy	-	-	0.73	123

Table 1. Results DiscussionType for XGBoost

BERT

DiscussionType

Class-wise Metrics				
Class	Precision	Recall	F1	Support
General	0.817	0.756	0.754	-
Accuracy	0.756	-	-	-
Deliberation	0.56	0.78	0.65	18
Imaginative entry	1.00	0.75	0.86	4
Other	0.00	0.00	0.00	1
Procedure	0.70	0.70	0.70	10
Seminar	0.92	0.85	0.88	67
Social	1.00	0.29	0.44	14
UX	0.42	0.89	0.57	9
Overall Metrics				
Metric	Precision	Recall	F1	Support
Micro avg	0.76	0.76	0.76	123
Macro avg	0.66	0.61	0.59	123
Weighted avg	0.82	0.76	0.75	123
Samples avg	0.76	0.76	0.76	123

Table 2. Results DiscussionType

In Table 2 we see the best results we got so far. This is with everything I mentioned we tried. A huge improvement was given by the history of DicussionTypes (increased F1 by 6) and a smaller one when adding the pseudonyms (increased F1 score by 1). Here class weighing ended up giving worse results about 72 F1 and without I got 75 F1. If I weight Other is also detected but without weighin it is ignored. I believe Other is not as important as the other classes so I outputed the better results.

Uptake

Class-wise Metrics				
Class	Precision	Recall	F1	Support
General	0.662	0.648	0.649	-
Accuracy	0.648	-	-	-
Affirm	0.60	0.54	0.57	28
Clarify	0.75	0.60	0.67	10
Disagree	1.00	0.50	0.67	2
Elaborate	0.60	0.55	0.57	22
Filler	0.42	0.67	0.52	12
None	0.76	0.77	0.76	48
Overall Metrics				
Metric	Precision	Recall	F1	Support
Micro avg	0.65	0.65	0.65	122
Macro avg	0.69	0.60	0.62	122
Weighted avg	0.66	0.65	0.65	122
Samples avg	0.65	0.65	0.65	122

Table 3. Results Uptake

In Table 3 we see the best results we got for uptake so far. Again the improvements were similar. But both previous uptakes and DiscussionTypes were needed and gave together an increase of 13 to the F1 score. Then the pseudonyms added 2-3 to the score. I mentioned I tried also adding the page with the hope that maybe if people are on different pages it would indicate uptake but this gave negative results. Here class weighing for the loss ended up to be really important. If disabled my F1 dropped by 13.

LLM Approach - Prompt Engineering

Uptake

Our experimentation included the utilization of the openhermes LLM model, specifically the 8-billion-parameter version, tailored for the uptake task through prompt engineering. Employing a prompt engineering technique known as few-shot prompt engineering, we provided the model with a select number of examples from each uptake category along with their corresponding labels. The objective was for the model to generate labels for unseen data based on this input. This process involved feeding each line of data to the model individually, obtaining labeled data as output.

However, the performance of the model yielded disappointing results, achieving only a 27% accuracy rate. A significant factor contributing to this subpar performance was the model's tendency to predict NaNs (Not a Number) with alternative labels. Although the accuracy for non-NaN categories reached 45%, this still fell short compared to the efficacy of alternative approaches we experimented with.

Additionally, the training duration for this model proved to be considerably lengthy, rendering it less advantageous compared to other methods we explored.

Question

Ran but terrible results. Not finetuned

Pivot

Not ran

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] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[11] Dominik Stammbach, Vilém Zouhar, Alexander Hoyle, Mrinmaya Sachan, and Elliott Ash. Re-visiting automated topic model evaluation with large language models. *arXiv preprint arXiv:2305.12152*, 2023.

[12] A Glazkova. Using topic modeling to improve the quality of age-based text classification. In *CEUR Workshop Proceedings*, pages 92–97, 2021.

[13] Lingyao Li, Zihui Ma, Lizhou Fan, Sanggyu Lee, Huizi Yu, and Libby Hemphill. Chatgpt in education: A discourse analysis of worries and concerns on social media. *Education and Information Technologies*, pages 1–34, 2023.

[14] Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. A bibliometric review of large language models research from 2017 to 2023. *arXiv preprint arXiv:2304.02020*, 2023.

[15] Chau Minh Pham, Alexander Hoyle, Simeng Sun, and Mohit Iyyer. Topicgpt: A prompt-based topic modeling framework. *arXiv preprint arXiv:2311.01449*, 2023.

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