16. Analysis of dialogue corpus

Eetu Hyypiö, Sami Karhumaa, Rainer Laaksonen
Github - https://github.com/Ehyyp/NLP_Project16_2024

***Abstract*—The project achieved all of the specifications of the project, except for the dialogue act correlations with emotions and sentiments.**

I. Introduction

Emotion and sentiment analysis has emerged as a critical area of research in natural language processing (NLP), enabling systems to detect, interpret, and respond to the affective states of users. Such capabilities are essential for applications in customer service, mental health support, social media moderation, and education, where emotional awareness can greatly improve user experience and engagement. Also dialogue systems are a central component in this kind of modern NLP applications, underpinning a variety of real-world interactions, from customer service chatbots to personal digital assistants.

For this project we use DailyDialog dataset [1], which is a widely used dialogue corpus and offers a structured dataset that provides natural and diverse conversations grounded in everyday topics. With its annotated dialogues covering various emotional and communicative aspects, DailyDialog serves as an invaluable resource for exploring the dynamics of human dialogue and informing the development of NLP applications. Another key annotation in DailyDialog is the categorization of dialogue acts, which represent the communicative intent behind each utterance (e.g., question, statement, request, or inform). Some basic statistics of DailyDialog are given in Table 1.

Total Dialogues	13,118
Average Speaker Turns Per Dialogue	7.9
Average Tokens Per Dialogue	114.7
Average Tokens Per Utterance	14.6

TABLE I
STATISTICS OF DAILYDIALOG

Emotion and sentiment analysis are critical components of NLP, especially in dialogue systems where the ability to understand user emotions can improve engagement and enhance user experience. Sentiment analysis typically categorizes text as positive, negative, or neutral, whereas emotion analysis involves identifying specific emotions such as joy, sadness, anger, or surprise.

Emotion detection in dialogue systems has gained considerable attention, especially with the increasing availability of conversational datasets like DailyDialog. Early models relied on deep learning architectures such as LSTM [2] and CNN [3] to capture context and semantics in text, while recent advancements have shown substantial performance improvements by leveraging transformer-based models, such as BERT [4]. Emotion detection has been explored further using dialogue-specific transformer models, such as DialogueBERT [5], which are fine-tuned on conversational data to capture turn-based emotional flows more effectively.

There are also other approaches, i.e., Knowledge-Enriched Transformer (KET) [6] and interactive double states emotion cell model (IDS-ECM) [7]. KET enriches text understanding by pulling in external knowledge to interpret subtle, culturally dependent, or implicit emotional cues, which enhances emotion detection in contexts where sentiment might otherwise be ambiguous. IDS-ECM addresses the interactive nature of dialogue by modeling the dual emotional states of both the speaker and the listener, making it highly effective for understanding how emotions develop and influence each other within a conversation. Both models are powerful extensions of standard NLP models and well-suited for emotion detection tasks in dialogue systems.

In this project we focus on lexicon and machine learning models [10]. Lexicon-based models rely on predefined sets of words and rules to interpret language, often using dictionaries or sentiment lists, making them more rigid but interpretable. In contrast, machine learning models learn patterns from data, allowing them to handle complex and varied language but often functioning as ”black boxes” with less transparency. Overall, lexicon models are simpler and faster, while ML models are generally more accurate and adaptable.

This project targets to elucidate NLP techniques analyzing data structure and content in order to detect emotional and sentimental nuances in dialogue. By creating scripts and building models tailored to detect emotional and sentimental cues, we aim to gain insights into the conversational characteristics of emotion and assess the effectiveness of various computational approaches in identifying these features. Specifically, the analysis will involve pre-processing the DailyDialog dataset, exploring the distribution of emotions, sentiments and dialogue acts, and implementing scripts and models to classify these labels.

II. Methodology

A. Dialogue statistics analysis

All dialogue data and political debate data were formatted into a data table where rows are dialogues and columns utterances. These tables were analyzed, after special characters were removed and utterances were tokenized. The goal was to calculate the vocabulary size, number of utterances, average number of tokens per utterances, average number of pronouns per utterance, average number of agreement wording per utterance, average number of negation wording per utterance and the average number of person/organization named-entities

per utterance. Goals were met and the results are presented in section III.

Vocabulary size was counted by iterating over each token in each utterance in each dialogue. If a token had not been seen before, it was added to a list and the vocabulary size was incremented. If the token was in the seen tokens list, nothing happened.

Counting utterances was accomplished by iterating over the dialogues.

Average number of tokens per utterance was calculated by counting the number of tokens and then dividing it by the number of utterances.

Number of pronouns per utterance was calculated iterating over each token in each utterance in each dialogue, and using the NLTK part of speech tagger to see if the token was a pronoun or not. The average was then calculated by dividing the number of pronouns by the number of utterances.

Custom lists of agreement and negation wording were used to calculate the number of agreement and negation wording by iterating over each token and checking if it is in the agreement/negation word list. Used agreement and negation words can be found in table II-A. The average was calculated by dividing with the number of utterances.

Number of person/organization named-entities were calculated by using the spacy named-entity tagger. The number of named-entities were calculated using three different pre-trained english spacy models. The small model ”en_core_web_sm”, medium model ”en_core_web_md” and the large model ”en_core_web_lg”. The code iterates over all utterances and makes entity tags for them. It then iterates over all entity tags in that utterance and counts the ”ORG” and ”PERSON” labels. The average was calculated by dividing the number of named-entities with the number of utterances.

Agreement words	Negation words
yes	no
ok	not
sure	don’t
okay	can’t
agreed	neither
agree	

TABLE II
AGREEMENT AND NEGATION WORDS USED IN THE ANALYSIS

B. Emotion analysis using WNAffect

The goal of the emotion analysis is to generate emotions for each utterance in DailyDialogue database and compare results from WNAffect and emotion tags in DailyDialog. For validating results, accuracy and precision and recall of the results is calculated using sklearn.metrics package where measures can be calculated for multi-class models. For generating emotions WNAffect Python package was used. WNAffect uses WordNet Domains 3.2 for resource which provides set of emotional words organized as tree structure [9]. WNAffect takes word and part-of-speech tag as input and gives none or more emotion for each word of utterance.

C. Sentiment analysis with Vader sentiment analyzer

Vader sentiment analyser is sentiment analysing software which is tuned especially sentiments expressed in the social media. Vader sentiment tool is based on rule-based and lexicon models. Scoring of the analysis is based on positive (pos), neutral (neu), and negative (neg) sentiments from which the compound score is calculated. Typical threshold values from compound scores are positive sentiment = compound score $>=$ 0.05, neutral sentiment = -0.05 $>$ compound score $>$ 0.05, and negative sentiment = compound score $<=$ -0.05. [11]

The scores are calculated for each utterance in DailyDialog dataset. The results of the Vader sentiment analysis are used in the correlation study between the sentiment and emotion states.

D. Correlation study between the sentiment and emotion states

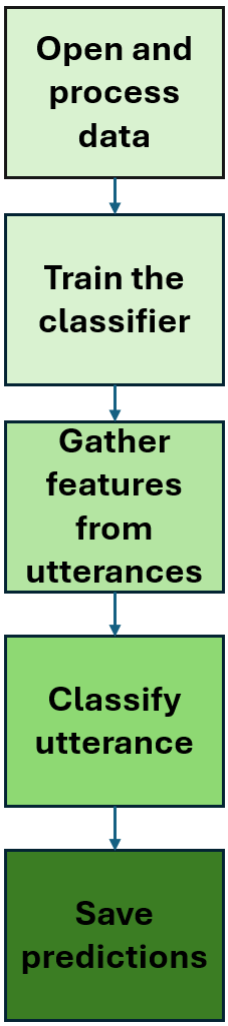
Goal of the study is to calculate correlation between sentiments found with Vader sentiment analyzer and emotions found with WNAffect. The correlation is calculated using compound score thresholds from Vader sentiment analysis, and base emotion found from WNAffect. Then scores are assigned for each utterance as 0 when there is no match between Vader sentiment and base emotion from WNAffect, 0.5 when there is partial match, and 1 if sentiments are fully matched.

Base emotion of the WNAffect results can be obtained by moving upwards from the emotion tree. For example, if moving one level up can be done with emotion.get_level(emotion.level-1) where “emotion” is emotion object returned by WNAffect. Base emotions of the WNAffects are neutral emotion, ambiguous emotion, positive emotion and negative emotion. In this case, both neutral emotion and ambiguous emotion are considered as neural emotion. Because WNAffect gives emotions for words basis, each utterance can have zero or more emotions. For that reason, list of base emotions values are constructed and for each emotion in utterance number is given where -1 is negative emotion, 0 is neutral or ambiguous emotion, and 1 is positive emotion.

Correlation of the emotions are calculated based on base emotion list from WNAffect and compound score from Vader sentiment analyzer. For example, if base emotion list of a utterance is [1,-1,1] (positive, negative, positive) and compound score is 0.6, then it is considered as partial match and correlation score is set to 0.5.

E. Predicting the dialogue acts in each utterance

We use the NLTK nps chat corpus to train a Naive Bayes classifier, that will be used to predict the dialogue acts for the daily dialogue data. The method to training this model can be found on the NLTK organization book chapter 6, section 2.2. The code implementation opens the specified dialogue data, trains the model and iterates over each utterance, predicting dialogue acts for them. The dialogue acts are then saved to a json file. This process is demonstrated in figure II-E



The accuracy of the naive Bayes classifier is tested with the Dailydialog dialogue act labels. The labeled dialogue acts are placed in just four categories. 1= inform, 2 = question, 3 = directive and 4 = commissive. The NLTK model predicts 15 different dialogue acts. Thus, the 15 dialogue acts are placed in the four Dailydialog categories. Some of the 15 dialogue acts do not match any of the labeled dialogue acts. Those are ignored. Then the accuracy of the model is calculated.

F. Dialogue act and emotion/sentiment correlation

The project details for this task were not met. The current code implementation calculates the most associated emotion and the average compound sentiment for each dialogue act. The Pearson correlation calculation isn't computing the correct correlation.

1) Most associated emotion and average compound sentiment: The first part of the code requires the dialogue emotion, sentiment and dialogue act json files. All of these are structured in the same way. They are lists of lists, where first element is the dialogue and second is the emotion/sentiment/dialogue act for the utterance in that place of the dialogue. The current files include all of these json files for the whole dataset, and they are all ordered the same way as the original data is.

The code block iterates over all three lists simultaneously. Most of the emotion data is empty, thus the execution checks if the emotion for that utterance is empty. If not, the emotion is saved in a dictionary for the specific dialogue act that was predicted for that utterance, and if the emotion is already in the dictionary, its value is incremented by one.

On the same iteration, the compound sentiment for that utterance is gathered and added to a list that is specific for the dialogue act found in that utterance. Each time this is done, a variable in the same list is incremented by one, telling how many compound sentiments were added together. This is done for the sake of calculating the average.

Lastly, for each dialogue act, the code finds the emotion with the highest value and calculates the average compound sentiment by dividing each dialogue acts compound sentiment by how many compound sentiments were added together.

2) Correlations: The correlation calculation takes the upper emotion json file instead of the regular emotions file, and the same dialogue act and sentiment files. The upper emotions are numerical.

The json files are made into pandas dataframes, which are then manipulated to be fully numerical. The dialogue act strings are made into numerical values. Emotion = 1, yAnswer = 2, Continuer = 3, and so on until 15. Empty element is 0. The sentiment dataframe is made to only have the compound sentiments as elements, empty elements are 0. The upper emotions dataframe has otherwise numerical values, but empty values were made to be 0.

The correlation is calculated between each dataframe column. For no dividing by zero error to occur, each column needs to have some variance. Thus each column in each dataframe is iterated over and checked if its standard deviation is zero. If it is, it is added to a list of columns to be removed. After this process, all zero deviation columns are dropped.

Lastly, the code calculates the column-wise correlations between the dialogue act dataframe and the emotion and sentiment dataframes. This ends up giving 30 correlations for both cases. The problem however, is that this is not what was specified in the project tasks. The idea was to find how some specific dialogue acts correlate better with specific emotion and sentiment. These column-wise correlations with all dialogue data tell nothing. The task specification was misunderstood and there was not enough time to correct it, after the problem was found.

G. Machine learning model for emotion predictions

The goal of the study is to use simple machine learning model for predicting emotions for each utterance in DailyDialog dataset. Emotions tags which are "no emotion", "anger", "disgust", "fear", "happiness", "sadness", and "surprise" was used as classes for the model. Different attributes was used for creating features for utterances, for example, sentiment from Vader sentiments analyser, dialogue act labels from Daily dialog dataset, number of pronouns, number of negation terms and tf-idf vectors. Four different machine learning models Multinomial Naive Bayes, Random forest classifer, Ridge classifier and Linear support vector classifier with basic settings were compared in the study.

H. Other implementations for emotion, sentiment and dialogue act analysis

As briefly mentioned in introduction, emotion detection in dialogue systems has a wide variety of models implemented and studied. We also looked into dialogue acts, which categorizes the functional role of an utterance (e.g., a question or a statement) and helps in contextualizing dialogue flows. This body of research highlights diverse approaches to understanding conversational nuances, often leveraging the dataset's rich annotations for multi-faceted emotional and intent recognition. Among these studies, we identified several key works that propose innovative methods, such as knowledge-enriched and state-based models, which enhance emotion and sentiment detection in dialogue. These studies not only advance our understanding of conversational dynamics but also serve as promising frameworks for further exploration in our project. Here are some suggestions of how we could utilize these as a hybridization/extension to our work:

KET uses external knowledge to help the model understand context, especially when dialogues involve entities or concepts that require additional background information. We could identify or build a knowledge base that includes concepts and entities frequently mentioned in the DailyDialog dataset. Common sources could be ConceptNet to include human-relevant relationships, or DBpedia/Wikipedia for more detailed information on specific entities (like names, places, events). After this knowledge base addition, for each dialogue in DailyDialog we could extract relevant entities and keywords. Query the knowledge base to fetch related concepts, synonyms, or associations. Then represent the external knowledge in a format that can be appended to the input sequence or incorporated into the attention mechanism of the transformer (e.g., by adding knowledge as embeddings or token sequences). To train the model, we could modify the architecture to integrate knowledge. In the KET setup, this often involves dual-input mechanisms where one input is the dialogue sequence, and the other is the external knowledge representation. With these extensions our model could improve emotion classification accuracy by understanding emotions in a broader context. For instance, if a dialogue references a cultural or situational cue (like "I'm stuck in traffic"), KET could add the use of external knowledge to understand this context and infer frustration or impatience.

We also found another promising study which integrates linguistic politeness cues [8], expressed through polite phrases, into conversational analysis, alongside emotional states and dialogue acts, with the aim of uncovering correlations between these cues. As DailyDialog is pre-annotated with emotion and dialogue act labels, we could enhance the dataset by using a politeness analyzer to annotate the utterances with politeness values. Another possibility is to use a pre-trained politeness analyzer model which could be adapted to the DailyDialog dataset to provide politeness scores without extensive re-training. After either of the options has been done, we could analyze correlations between politeness and dialogue acts (e.g., requests, commands, apologies) to see if politeness patterns differ between these types. Also we could track how politeness changes with emotional tone: does frustration often lead to less polite language, or do certain dialogue settings maintain politeness regardless of emotion? By implementing politeness detection alongside emotion and dialogue act analysis, our project could offer a more holistic understanding of conversational nuances.

III. Results and discussions

A. Dialogue statistics

The entire dataset includes 10 topics. Thus, based on data from table III-A-II, the average number of utterances per topic is 11609,5 and the average vocabulary size for a topic is 2627,3. The political debate topic, table III-A-III, has 1611 utterances, just 13,9% of the average, but it also has a vocabulary size of 1567, which is 59,6% of the average. The average number of tokens per utterance in the politics topic is also just two thirds of what the average over all topics is. This indicates a pattern in talking about politics. A pattern where speakers are economic with words and take fewer turns to speak, but still speak about broad topics. The fact that both number of utterances and average tokens per utterance are smaller than average, yet the vocabulary size is still almost 60% of the average shows that political dialogues invoke broader terminology than the average topic.

The politics topic has around the same average number of negation words per utterance than the whole dataset, but it has 37,5% more agreement words than in the whole dataset. We could have ended up with a dataset that has very agreeable speakers due to the small number of utterances in the politics topic, which would lead to a higher number of agreement words with this set of speakers. Or, it is because politics is not as divisive of a topic as we think, and it only seems divisive online.

Vocabulary size	26273
Number of utterances	116095
Average number of tokens per utterance	9,506
Average number of pronouns per utterance	0,911
Average number of agreement words per utterance	0,024
Average number of negation words per utterance	0,049

TABLE III
STATISTICS FOR THE WHOLE DATASET

Vocabulary size	1567
Number of utterances	1611
Average number of tokens per utterance	6,849
Average number of pronouns per utterance	0,729
Average number of agreement words per utterance	0,033
Average number of negation words per utterance	0,045

TABLE IV
STATISTICS FOR THE POLITICAL DEBATE TOPIC

In tables III-A-IV and III-A-V we see the results of the entity tag calculations. Three different spacy models were used, but they didn't make a significant change in the outcome in either tests.

The politics topic has roughly half the average number of person/organization named-entities per utterance than all topics combined has. This can also be due to the small dataset,

but it also can be an indicator that the speakers are talking more about ideas and problems/solutions than individuals and organizations. At first glance this does not seem right, since much of politics is centered around politicians, who are individuals, and political parties, which are organizations.

Used spacy model	Average number of person/organization named-entities per utterance
Small	0,030
Medium	0,027
Large	0,026

TABLE V
ENTITY TAG RESULTS FOR THE WHOLE DATASET

Used spacy model	Average number of person/organization named-entities per utterance
Small	0,015
Medium	0,012
Large	0,014

TABLE VI
ENTITY TAG RESULTS FOR THE POLITICAL DEBATE TOPIC

B. Emotion analysis using WNAffect

Implementation of the emotion analysis consists of functions for generating emotions for each utterance and saving all emotions from each utterance found into the single file. In the DailyDialogues database there is seven emotion tags used for utterances which are “no emotion”, “anger”, “disgust”, “fear”, “happiness”, “sadness”, and “surprise”. In the validation phase, each utterance emotion found with WNAffect is compared with tag in the DailyDialog dataset. Utterances were preprocessed with nltk word tokenizer which divide each utterance to tokens. Also, nltk pos_tag function were used for finding part-of-speech tag for each word in the utterance.

Two approaches were used for comparing tags in DailyDialog and received emotions from WNAffect. In the first approach (A1), received WNAffect emotions were compared directly to DailyDialog tag where one match in received emotions were considered as match. In the second approach (A2), received WNAffect emotions were expanded with all upper level emotions found for each emotions in utterances. Upper level emotions were searched up to level 4 in the WNAffect tree. Results of the two approaches used is presented in table VII.

A1	
Accuracy	0.64
Precision	0.018
Recall	0.12
A2	
Accuracy	0.64
Precision	0.058
Recall	0.2

TABLE VII
ACCURACY, PRECISION AND RECALL FOR APPROACHES A1 AND A2

From the results in table VII, it can be observed that precision of predictions in both approaches are quite low. Although using also upper level emotions for evaluating matches seems to improve precision and recall rate, results suggest that WNAffect does not predict emotions similarly than tags suggest in the DailyDialog dataset. Although, there can be more different approaches than presented and it can be subjective and case depended on which kind of approach is appropriate.

C. Sentiment analysis with Vader sentiment analyzer

Figure 1 shows results of the Vader sentiment analyzer. Results suggest that sentiment of the utterances in DailyDialog dataset contains large number of neutral utterances. Results can correlate to the emotion tags from DailyDialog because 83 % of tags is ”no emotion” which can be though also as neutral sentiment.

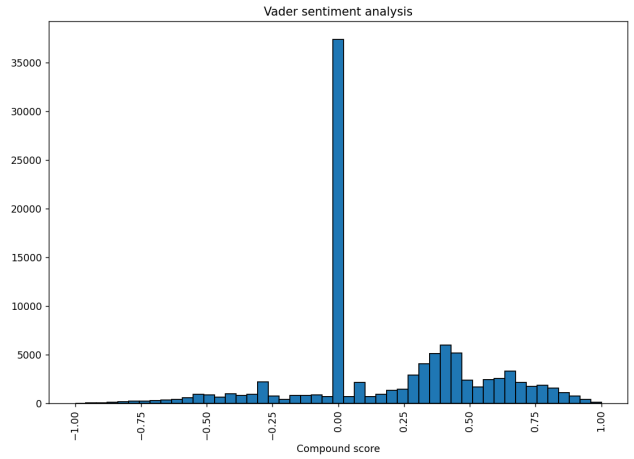


Fig. 1. Vader sentiment analysis compound score distribution

D. Correlation study between the sentiment and emotion states

Results of comparing sentiments extracted with Vader sentiment analyzer and WNAffect are shown in the figure 2. Initial results on figure 2 suggests that majority of the predicted sentiments are different between Vader sentiments and WNAffect. Because WNAffect results with no emotion was also taken in account at results in figure 2, the results can be misleading because ’none’ valued results is considered, in this case, always incompatibility. When excluding ’none’ values from compatibility index, results suggest that majority of Vader sentiments matches sentiments extracted with WNAffect shown in figure 3.

E. Generating dialogue acts for utterances

As was specified in the section II-E, the dialogue act predictions were made with a Naive Bayes classifier trained on the NLTK nps chat corpus. The accuracy of the classifier on the NLTK nps corpus is 0,667.

The NLTk classifier dialogue acts are matched into the Dailydialog dialogue act categories as described in table VIII. Dialogue acts that do not fit into any of the four categories are ignored. After iterating over all utterances in all dialogues and checking whether the labels match, we end up with the

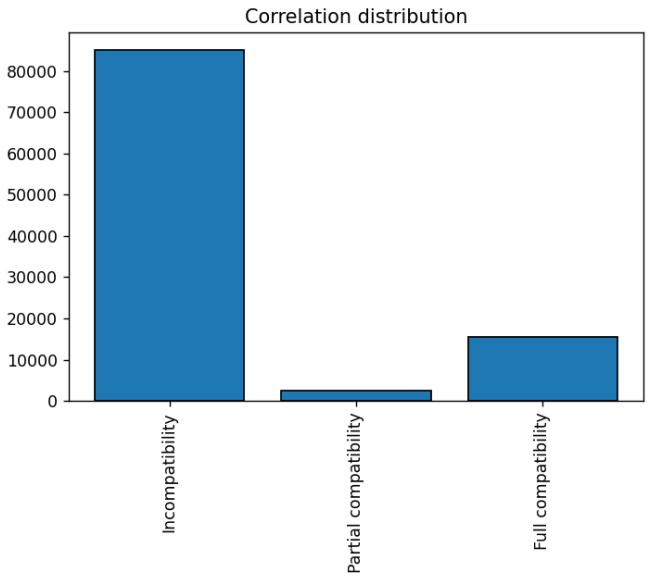


Fig. 2. Compatibility index distribution between Vader sentiments and WNAffect

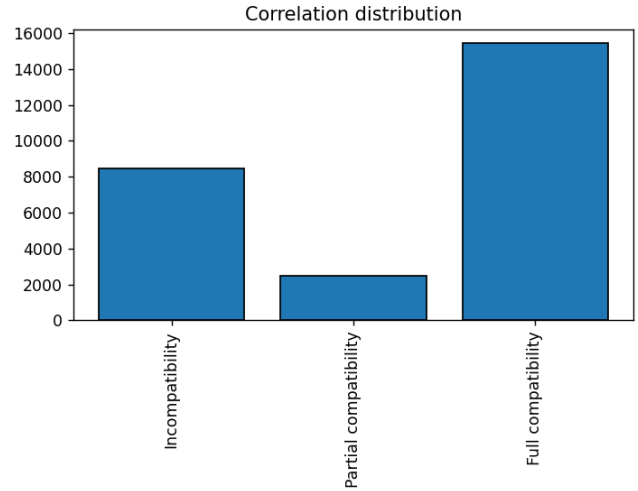


Fig. 3. Compatibility index distribution between Vader sentiments and WNAffect with 'none' results removed

accuracy of 0,507 in the whole dataset. There was 51262 correct and 49889 incorrect predictions. The result is predictable, since the accuracy of the model on a new dataset is lower than what it is on the set it was trained on. The efficiency of the model is thus not the best on this dataset, but it is not much lower than the accuracy on the set it was trained on.

F. Dialogue act correlations with emotions and sentiments

As mentioned in the II-F2 section, the results in table III-F-VII are not what this task aimed to find. The correlations that the program calculates do not make sense in what we are trying to do, hence I have not included them here.

The emotion distribution is dominated by stupefaction and benevolence, hence why most the most associated emotions are mainly benevolence and stupefaction. Why this is, however, is hard to say. The average compound sentiment does not vary much. All average compound sentiments for dialogue acts where an emotion was found, are positive and under 0,5, i.e. slightly positive, to a varying degree. This makes sense with benevolence, as it is a positive emotion. Stupefaction however can go both ways.

NLTK model dialogue act	Closest Dailydialog dialogue act category
Emotion	No fit
yAnswer	1
Continuer	3
whQuestion	2
System	No fit
Accept	4
Clarify	1
Emphasis	1
nAnswer	1
Greet	3
Statement	4
Reject	3
Bye	1
Other	No fit
ynQuestion	2

TABLE VIII
NLTK CLASSIFIER DIALOGUE ACTS CLASSIFIED INTO THE DAILYDIALOG DIALOGUE ACT CATEGORIES

Dailydialog dialogue act category	Category number
Inform	1
Question	2
Directive	3
Commissive	4

TABLE IX
DAILYDIALOG DIALOGUE ACT CATEGORIES

Dialogue act	Most associated emotion	Average compound sentiment
Emotion	No emotions found	-0,197
yAnswer	Stupefaction	0,413
Continuer	Stupefaction	0,149
WhQuestion	Benevolence	0,103
System	No emotions found	0,067
Accept	Benevolence	0,360
Clarify	Stupefaction	0,246
Emphasis	Stupefaction	0,281
nAnswer	Stupefaction	0,076
Greet	Benevolence	0,019
Statement	Benevolence	0,199
Reject	Stupefaction	0,122
Bye	Benevolence	0,172
Other	Stupefaction	0,214

TABLE X
MOST ASSOCIATED EMOTION AND THE AVERAGE COMPOUND SENTIMENT FOR DIFFERENT DIALOGUE ACTS, CALCULATED FROM THE DAILYDIALOG DATASET

G. Machine learning model for emotion predictions

Result of the machine learning study contains accuracy, precision and recall scores for each machine learning models used which is shown in table XI. Implementation of study includes creating feature vectors for sentiment, pronoun count, negation count, and dialogue act label and saving them in to single file. When dataset is loaded also tf-idf vectors is created and added existing feature vectors. Before training the classifiers, the data was also balances so that 65000 instances of utterances with "no emotion" tags were removed from

dataset. Balancing was done because initial testing showed that balancing the data before training the models gives better results. After balancing, vectorized data and labels was splitted to train and test sets with test set size of 20 %.

Overall, results seems quite similar between machine learning models but there is distinct differences in the results. For example, results of the Multinomial Naive Bayes classifier shows that classifier does not predict classes which have low number of labels in the data and tend to classify to labels which have higher occurrence. Random forest classifier seems to give best precision scores. Although, recall and accuracy of the SVM model was marginally better than random forest classifier. The confusion matrices of four models are presented at figures 4 - 7.

Multinomial Naive Bayes	
Accuracy	0.72
Precision	0.21
Recall	0.23
Random forest classifier	
Accuracy	0.77
Precision	0.75
Recall	0.42
Ridge classifier	
Accuracy	0.76
Precision	0.68
Recall	0.39
Linear support vector classifier	
Accuracy	0.78
Precision	0.62
Recall	0.44

TABLE XI
ACCURACY, PRECISION AND RECALL MACHINE LEARNING MODEL PREDICTIONS

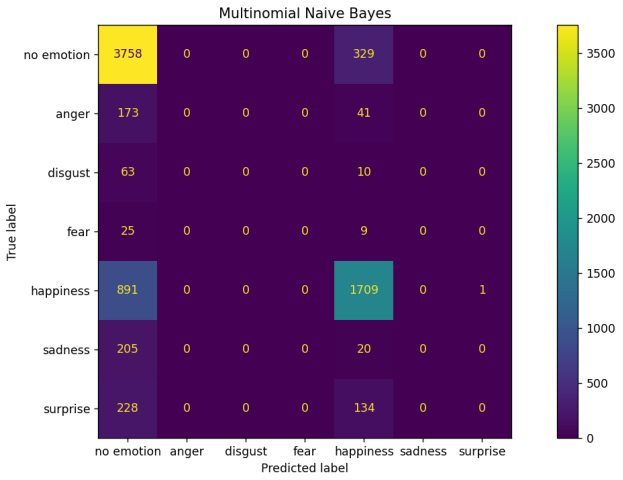


Fig. 4. Confusion matrix, Multinomial Naive bayes predictions

IV. Overall discussions

Overall, results seem to point, that using machine learning model for analysing emotion can be better solution than lexical based models. For example, precision which shows how much true positive results are in all positive predictions, seems to be quite low with WNAffect analyser. Although in this

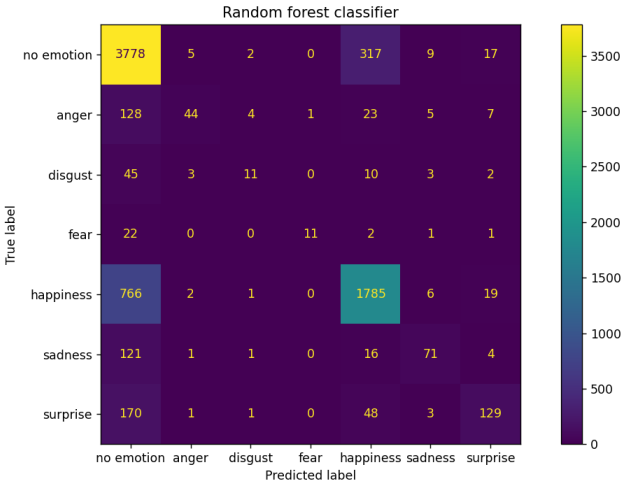


Fig. 5. Confusion matrix, Random forest predictions

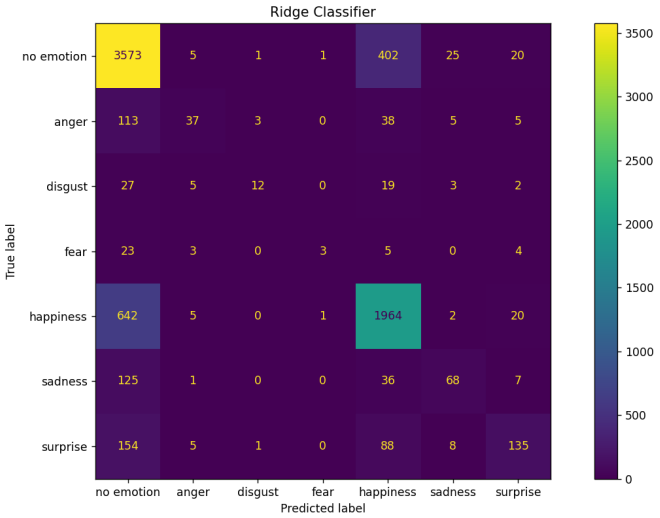


Fig. 6. Confusion matrix, Ridge classifier predictions

case, results cannot be compared directly, because data was balanced heavily when using machine learning models. Also, WNAffect can give several predictions which is not part of the tag set in DailyDialog dataset which leads to different kind of approaches of preprocessing the emotions of WNAffect. Different approaches can be expanded further but initial results with two approaches studied suggest that there is difference between approaches but both approaches are still unpractical in general.

It was recognized that the naive Bayes classifier trained on the NLTK nps chat had low accuracy with regards to predicting dialogue acts for the utterances in the Dailydialog dataset. Finding a more accurate model for predicting dialogue acts for these topics is of interest.

Vader sentiment analyser predicts numerical value which should correlate the sentiment of the utterance. When comparing sentiments in the WNAffect and results of Vader sentiment majority of results seems to be fully compatible which suggest

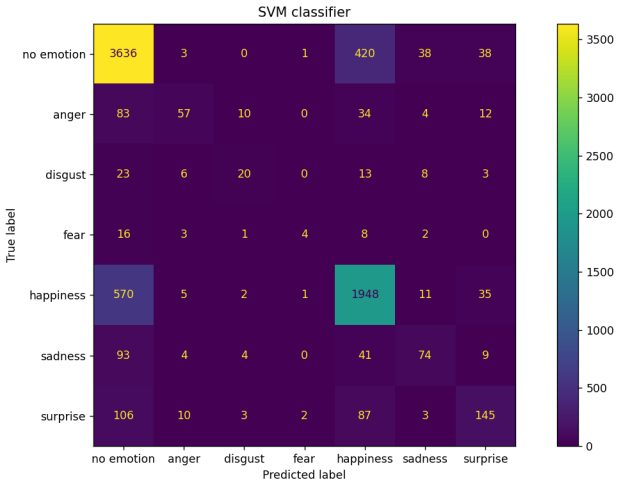


Fig. 7. Confusion matrix, SVM classifier predictions

that prediction sentiments of the utterances can be practical. Additionally, emotions in the DailyDialog tags can be possible to change to base sentiments which are positive, neutral, and negative sentiments. This model could have been used further to analyse also the sentiment accuracy which can reveal more about how different models can predict sentiment.

Even though we didn't manage to achieve the goal of recognizing correlations between dialogue acts and emotion/sentiment, there has been promising studies for this kind of behaviour [8]. It would have been extremely interesting to find out if we managed to get similar results where politeness, or some other kind of behaviour, would be highly correlative to emotions and dialogue acts.

V. Conclusions

In conclusion, this project successfully accomplished the majority of its intended tasks, demonstrating effective planning and execution throughout. The results achieved were largely in line with the initial objectives, showcasing significant progress and valuable insights. However, while most outcomes met expectations, there were areas where results could have been further optimized, suggesting room for future refinement. These experiences have provided a deeper understanding of the challenges and opportunities within the project scope, enabling continued improvement and development. In practice this could mean managing to successfully implement the correlation recognition to emotion/sentiment and dialogue acts. Additionally this could be taken further with implementation of behaviour detection which could make the results so much more interesting.

References

[1] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In IJC-NLP, volume 1, pages 986–995.

[2] Sepp Hochreiter and J¨urgen Schmidhuber. 1997. Long short-term memory. Neural Computation, 9(8):1735–1780.

[3] Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[5] Z. Zhang, T. Guo and M. Chen, "Dialoguebert: A self-supervised learning based dialogue pre-training encoder", CIKM, pp. 3647-3651, 2021.

[6] P. Zhong1, D. Wang, C. Miao, "Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations", arXiv:1909.10681, 2019.

[7] D. Li, Y. Li, S. Wang, "Interactive double states emotion cell model for textual dialogue emotion prediction" Knowl. Based Syst. (2020)

[8] C. Bothe and S. Wermter. 2022. Conversational analysis of daily dialog data using polite emotional dialogue acts. In Proceedings of the 13th Language Resources and Evaluation Conference. 2395–2400.

[9] Clemtoy. publication date unknown. WNAffect. Searched 9.11.2024 at: https://github.com/clemtoy/WNAffect

[10] N. M. Hakak, M. Mohd, M. Kirmani and M. Mohd, "Emotion analysis: A survey", Proc. Int. Conf. Comput. Commun. Electron. (Comptelix), pp. 397-402, Jul. 2017.

[11] cjhutto. publication date unknown. vaderSentiment. Searched 9.11.2024 at: https://github.com/cjhutto/vaderSentiment