**DI 504 – FOUNDATIONS OF DEEP LEARNING PROJECT PROPOSAL**

**DL BASED CLASSIFICATION OF EMPLOYEE FEEDBACK**

**Baran Özden – 2168334**

1. **PROBLEM**
   1. **Definition**

Organizations regularly conduct employee engagement surveys to gain understanding of employees’ current perspective about their workplaces. One of the most important and common questions in these surveys is “What would you change in your company?” or “What is the thing that affects you the worst in your working life?” and the expected response is usually in free-text format. For companies with thousands of employees, understanding these responses is a very hard task because there are thousands of free-text responses to be analyzed. Usually, Human Resources departments make manual labelling on these responses to analyze the company’s weak points. But it takes tens of labor days. Thanks to advancements in natural language processing, this could also be seen as a problem in the field of NLP.

Responses to employee dissatisfaction questions may be related to a single topic or it could be about several topics. HR experts usually label these responses as a main topic which is the main theme of the response or the most critical part and supporting topics if exist in the response. So, the mapping is usually like {response -> [mainTopic; supportingTopic1; supportingTopic2; …]}. As a starting point the problem could be separated into two problem domains. The first one is multi-class classification of the main topic and the second one is multi-label classification of all the possible topics for a response. For the sake of simplicity, I will start with the first problem and if I have extra time, I will try to deal with the second problem too.

* 1. **Dataset**

The original dataset consists of employee information, responses to different questions, human-annotated labels of responses, and net promoter score. In this study, I am going to use only responses to the dissatisfaction question and the related labels. Data is sampled with respect to domain experts’ opinions on important labels. Also, during sampling some responses are downsampled to ensure balance between labels. The final dataset contains 8800 samples and 11 classes. To ensure privacy, labels will be shared in numerical form rather than text form. Below, some mimicked examples in English are given:

|  |  |
| --- | --- |
| **Response** | **Label** |
| My team leader never appreciates my efforts. | 3 (Manager) |
| I doubt the vision drawn for the company. | 5 (Top Management) |
| Office climate is not set properly. | 7 (Working Environment and Conditions) |

Table : Mimicked Data Example

A screen shot of a graph

Description automatically generated

Figure : Distribution of Number of Words in Responses

Above, distribution of number of words in responses is given. Mean of number of words is 25.32, minimum number is 1 and the maximum number is 658.

1. **LITERATURE REVIEW**

Natural Language Processing with machine learning utilizes both traditional machine learning algorithms such as Naive Bayes and Decision Trees and deep neural network architectures such as Recurrent Neural Networks (LSTM, GRU) and Convolutional Neural Networks mostly with specialized feature engineering techniques like TermFrequency-InverseDocumentFrequency, Bag of Words and, Word Embeddings (Word2Vec, GloVe, fastText). In contrast to these traditional ML and DL approaches, Transformer architectures which is a very popular deep learning architecture are widely used in NLP without a need of explicit feature engineering in downstream tasks. There are many successful transformer models which are pretrained on a huge amount of data and published open source both with their architectures and different checkpoints (model parameters like weights).

The final dataset contains 8800 samples and 11 distinct labels. Since text data is complex in nature (having Turkish texts make it harder too) and we have many labels to map using traditional ML directly could be meaningless. Training a deep neural net from scratch needs so many samples and we do not have. Also, to utilize each sample effectively there is need of a context-aware system. Therefore, I narrowed my search to utilization of context-aware pretrained models which are already holding much information about the structure of language and relation between words in a text. This led me to focus on the papers about text classification with transformers.

I used “text classification”, “multi-class text classification”, “multi-label text classification”, “Turkish text classification”, “employee feedback classification” as keywords through my search. After reviewing some papers, I came up with two papers. Unfortunately, none of them are about employee feedback classification, but I rely on the generalization of the techniques and general language understanding.

The first paper is “Transformers are Short-text Classifiers” written by Fabian Karl & Ansgar Scherp. Through the paper, authors are comparing different architectures and models by their performance on short text classification. They used eight datasets which are R8 (Reuters news with 8 classes), MR (movie-review documents), SearchSnippets (snippets returned by a search engine), Twitter (tweets categorized as negative or positive sentiments), TREC (questions with 6 classes), SST-2 (Stanford Sentiment Treebank with negative and positive sentiments), NICE (goods and services with 45 classes), STOPS (short texts of products and services inspired from   
Amazon and YELP) for both training and benchmarking. Their datasets are like mine in terms of the nature of input and output. They evaluated BagOfWords based multilayer perceptron’s, LSTM’s, CNN’s, Graph NN’s, Graph Convolutional Nets, and BERT-family models. Except R8 and Snippets, BERT-family models performed better in all benchmarks. They made evaluation with respect to model accuracy.

The second paper is “Pretrained Neural Models for Turkish Text Classification” written by Halil İbrahim Okur & Ahmet Sertbaş. This paper is chosen to mention Turkish text classification. Through the paper, authors evaluate the performance of different pretrained models and word vector algorithms on Turkish text classification datasets. They used two datasets which are TTC-3600 (Turkish news collection with 6 classes, balanced) and TRT-Haber dataset (news collected from website of TRT Haber with 11 classes) for benchmarking. They tokenized the datasets and then generated word embeddings by Fasttext, BERTurk, DistilBERTurk, Electra, and Word2Vec. After gathering embeddings, they used SVM as their classifier. They evaluated model performances by Precision, Recall, F1-score, and accuracy. BERTurk, DistiBERTurk, and Electra performed best with very close scores.

1. **SOLUTION STRATEGY**

Through this proposal, I mentioned my problem, and my findings from literature. I plan to follow the data science lifecycle and apply some methods I found through my research. I will start with preprocessing my data. My main strategy will be the use of pretrained transformer models to gather word embeddings to use in my models. After gathering embeddings, I plan to build a baseline with a traditional machine learning model. Then I will try to beat that model by adding classification layers to the pretrained models. I will use basic classification evaluation metrics such as accuracy, precision, recall, and F1-score.