Classification of Questionnaires with Open-Ended Questions

Miraç Tuğcu, Tolga Çekiç, Begüm Çıtamak Erdinç, Seher Can Akay and Onur Deniz
Natural Language Processing Department, Yapı Kredi Teknoloji, Istanbul, Turkey

Keywords: QA Classification, Data-Centric AI, Clustering, Language Models, Deep Learning, NLP, BERT.

Abstract: Questionnaires with open-ended questions are used across industries to collect insights from respondents. The answers to these questions may lead to labelling errors because of the complex questions. However, to handle this noise in the data, manual labour might not be feasible due to low-resource scenarios. Here, we propose an end-to-end solution to handle questionnaire-style data as a text classification problem. In order to mitigate labelling errors, we use a data-centric approach to group inconsistent examples from the banking customer questionnaire dataset in Turkish. For the model architecture, BiLSTM is preferred to capture long-term dependencies between contextualized word embeddings of BERT. We achieved significant results on the binary questionnaire classification task. We obtained results up to 81.9% recall and 79.8% F1 score with the clustering method to clean the dataset and presented the results of how it impacts overall model performance on both the original and clean versions of the data.

1 INTRODUCTION

Classification is one of the core tasks that has largely been studied in machine learning and by extension, text classification is a common area of research for natural language processing (NLP). Text classification can be broadly defined as categorizing a text of arbitrary length and composition into two or more predefined classes. It has been used for sentiment analysis (Tejwani, 2014), spam detection (Bhowmick and Hazarika, 2016), intent classification (Larson and Leach, 2022), and so on. In earlier research, sparse tf-idf vectors have been used with methods such as Support Vector Machines to classify various types of textual data. With the introduction of dense vectorial representations of tokens in text such as word2vec (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017) which is shown to retain semantic information of words successfully, deep learning based text classification models have gradually surpassed success of earlier models. Recent advances in contextual embeddings by transformer networks have further improved on the shortcomings of word vectors namely the semantic relation between words far apart in a text. Considering these advances, we have approached the problem of analyzing questionnaire data as a purely text classification problem. Rather than trying to extract information from each answer one by one, by introducing the questionnaire as complete text to a contextual embedding model, we trained a classification model.

Classifying multiple open-ended questions & answers requires an understanding of different aspects of creative responses in contrast to close-ended questions. This nature of open-ended questions allows elaboration from respondents, thus making them important in questionnaires and surveys. On the other hand, human analysis of text responses is time-consuming and domain knowledge is needed for non-trivial questions. Surveys are widely used in education, research, and industry domains to get feedback or information from a targeted group of people. Nevertheless, open-ended questions may cause noisy data while increasing the variability of answers. As a result, cleaning the data or annotation can be a cumbersome process even for domain experts.

In this work, we approach classifying questionnaires as a pure text classification problem and introduce our results. We also apply a data-centric approach to reduce the labelling error in the dataset. Details of the literature review are mentioned in Section 2. The preparation process and properties of the Turkish customer questionnaire dataset on the banking domain are explained in Section 3. The proposed model architecture is described in Section 4. The clustering approach to mitigate the noise in the dataset is detailed in Section 5. The experiment setup and results of the mentioned methods are discussed in Section 6.

Tuğcu, M., Çekiç, T., Erdinç, B., Akay, S. and Deniz, O.
Classification of Questionnaires with Open-Ended Questions.
DOI: 10.5220/0012233200003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 1: KDIR*, pages 413-420
ISBN: 978-989-758-671-2; ISSN: 2184-3228

Finally, the conclusions we reached and the details of our future research are shared in Section 7.

2 RELATED WORK

Contemporary methods of text classification with BERT (Devlin et al., 2019) involve using a classifier layer that can leverage from transfer learning and fine-tuning BERT to create robust models for a specific task.

A recent method of multi-class sentiment classification proves that using a simple model architecture of dropout layer (Srivastava et al., 2014) and softmax classifier layer is able to produce satisfying results (Munikar et al., 2019). The same architecture is applied alongside our architecture for questionnaire classification to observe if BERT with a simple classifier network can capture the bidirectional dependencies of question-answer pairs in a text and be robust against labelling errors. FakeBERT (Kaliyar et al., 2021) uses BERT embeddings to perform binary classification for fake news classification by using a CNN layer network as a classifier and comparing results of using GloVe (Pennington et al., 2014) embeddings which are context-independent and unidirectional. For our problem, multiple question-answer pairs could be dependent on each other. Hence, we choose BiLSTM (Graves and Schmidhuber, 2005) model in our architecture to capture sequential dependencies from BERT embeddings which are contextualized and bidirectional. To the best of our knowledge, this is the first work to approach the open-ended questionnaire classification problem as a text classification problem.

3 QUESTIONNAIRE DATASET

In order to create a dataset, Turkish customer questionnaire data in the banking domain is collected from Yapı Kredi. There are different question categories for customer types - Turkish citizens, foreign customers, underage customers, and so on. In this work, questionnaire data of the Turkish citizens' category is used instead of others due to its large proportion compared to other categories. The raw data was in the format of email texts, and answers to the questions were in a separate reply email. Thus, the first challenge of creating the dataset was parsing the question-answer pairs from emails to a structured format.

3.1 Data Parsing

A rule-based parser is developed to extract question-answer pairs from the reply patterns of respondents.

These patterns are about where answers are located in a reply because the questions are sent in a default format in the first email. Two of the most frequent reply patterns are either appending answers next to questions in the reply section or copying the questions to add answers next to them. So we defined rules to check questions in replies and answers next to or beneath them. Specific keyword and length controls are also used to ensure there are no mistakes in the parsing process.

This extraction approach helped to cover $\sim$80% of the email data. We experimented with using mail contents in raw format, but this approach helped us generalize the data and save it in a semi-structured format.

The task is a binary classification problem and the classes are either an issue found with answers or not. The surveyors decide if there is an issue when an unexpected answer arrives to a question.

The data was self-labelled because the surveyors of questionnaires sent a different reply email with extra questions if they decided there was any issue with the answers.

After parsing emails, the dataset is set for binary classification. Given an arbitrary question text q and a corresponding answer text a for a question-answer pair $p = (q,a)$, an example from the dataset includes a series of question-answer pairs $\{p_1, ..., p_n\}$. The task is to predict the class $y \in \{no\_issue, issue\_found\}$ for each example. The dataset has 19006 questionnaire examples with a total of 186092 question-answer pairs.

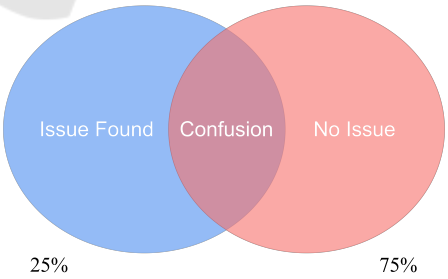

Figure 1: Venn diagram for class confusion and percentages of class distribution in the dataset. Inconsistent samples were detected empirically, but it is impossible to find out how many of them are there.

3.2 Data Inconsistency

There were two reasons for inconsistency in our data, misleading email replies and missing features. Misleading replies are caused by the puzzling nature of the open-ended questions. Even domain experts we consulted have trouble identifying if there is an issue with the answers in this situation. Thus, label errors in data occur due to confusion of labels because the similar questionnaires overlap in different labels, as shown in Figure 1. There are also scenarios where another channel other than emails (calls, short messages) is used to decide if there is an issue with the customer. However, it is unlikely to detect this problem by only using emails due to not having any knowledge if another communication channel is used.

With a sufficiently large dataset, state-of-the-art deep learning models are able to iron out inconsistencies. Because the data was not large enough, in order to tackle the inconsistency problem, we focused on a data-centric approach to handle noisy data.

3.3 Data Preprocessing

After the questionnaire dataset is created, the order of question-answers is shuffled for each example in the dataset to apply regularization and reduce overfitting during the training. Punctuation characters are removed, and lowercase characters are used because of the improper usage of punctuation and uppercase characters in replies. A special token `[SEP]` is added after each question-answer pair when tokenizing to separate question-answer pairs in the input. For empty answers or any answer with a length shorter than one non-whitespace character `[UNK]` token is used. These two special tokens were already in the dictionary of the pre-trained BERT model's tokenizer that is used. The details of the model will be further explained in Section 4.

4 MODEL ARCHITECTURE

Sequence representations of concatenated and tokenized question-answer pairs are used for binary classification to find if there is an issue or not with the given questionnaire, depending on the answers to the questions. The architecture of the model is shown in Figure 2. The model is expected to generalize what an issue could be without further auxiliary features about the issue itself. To achieve this, representations of an attention-based model like BERT (Devlin et al., 2019) are used for classifying, and pre-trained model is further explained in Subsection 4.1. The classifier layer is explained in Subsection 4.2

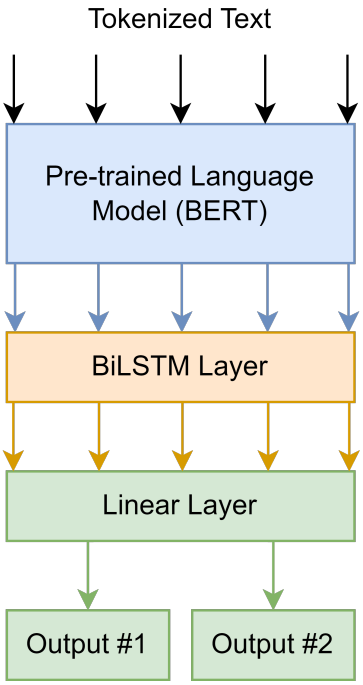

Figure 2: Model architecture for binary classification of questionnaires.

4.1 The Pre-Trained Language Model

Using contextualized word embeddings instead of static word embeddings is a must to represent the context. Different question-answer pairs in the input and answers may be related to each other. Therefore, using question-answer pairs in a text requires representing the context between word tokens across the question-answer pairs. To achieve this, the BERT model is chosen. The word embeddings of the BERT model are able to represent the context between question-answer pairs by using a bidirectional self-attention mechanism. Thus, the word embeddings can represent the context for both the left and right ordering of tokens and capture different dependencies from both sides for each token. The BERT model we used in this work is the BERTurk model (Schweter, 2020), which is a model trained in Turkish corpora. The version with a 32K dictionary size and base architecture with a hidden size of 768 is used. Additional pre-training of BERT model on the domain of the task can improve the text classification performance (Sun et al., 2019). Therefore, the BERTurk model is pre-trained on Masked Language Modelling task with banking documents and old questionnaire emails to further adapt it to the banking domain.

4.2 Classifier Head

During the experiments, using a BiLSTM layer before the linear layer performed better at generalizing the questionnaire data compared to using only a linear neural network inside the classifier layer. While increasing the complexity of the model, the BiLSTM layer also helps to capture higher-level representations of the BERT model by modelling contextual information for both directions. BiLSTM is able to enhance the word embeddings of BERT by using sequential dependencies. For regularization, we also applied a dropout layer before and after the BiLSTM during the training phase. Input and output sizes of the BiLSTM layer are the same as BERT's hidden size (i.e., 768). The input of the linear layer is the concatenated output of BiLSTM's last step. The output size of the linear layer is the number of class sizes which is two. None of the BERT's layers are frozen during the training. Therefore, BERT is finetuned when training the classifier layer, which helps to adapt higher-level representations of BERT to the task it is used.

4.3 Loss Function

The cross-entropy loss function (L_{CE}) is used to calculate the loss at the end of the training pipeline. There is no softmax layer for the outputs in the model architecture, but loss function L_{CE} applies the softmax function internally, as shown in Equation 1 where N is the number of classes (i.e., output neurons) and x_{target} is the value of the target output neuron.

$$L_{CE} = l(x, x_{target}) = -\log\left(\frac{\exp(x_{target})}{\sum_j^N exp(x_j)}\right) \quad (1)$$

L_{CE} is used instead of using a binary cross-entropy (BCE) loss from a sigmoid output to train the neural network. While it is common and more efficient to use BCE loss for a binary classification problem, we observed using the loss function L_{CE} contributes more to the balance problem of classes in the dataset, as shown in Table 1. The output of the loss functions L_{CE} and BCE must be the same if the inputs to the functions are also the same for binary classification. Thus, the outcome of the experiment differs due to the increased complexity of the neural network by using L_{CE} with two output neurons. Also, using L_{CE} enables the model to be used for non-binary classification tasks as well in case of need.

Correctly predicting all the customers with issues is the main problem we are trying to optimize in this classification; hence the significant increase of the recall score observed for Class 1 (as shown in Figure 1) contributes toward the desired solution.

5 CLUSTERING APPROACH

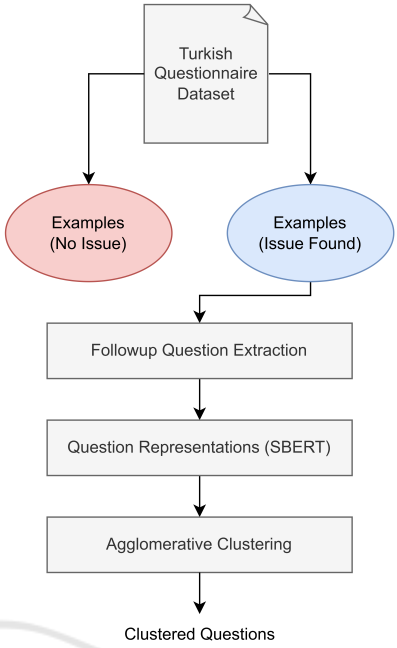

Figure 3: Clustering pipeline to obtain follow-up question clusters.

The dataset is created without annotation, which causes noisy labels due to the aforementioned data properties in Subsection 3.2. This situation is similar to weak supervision noise because email replies are used directly to get labels rather than manual annotation, which is time-consuming and not always feasible. A BERT-based classifier is not robust to weak supervision noise and it can significantly degrade its performance (Zhu et al., 2022). For this reason, a clustering-based approach is used to solve the data inconsistency problem that is related to noise to reduce manual review time and boost text classifier performance.

The questionnaires with any issues in the training dataset are clustered using follow-up questions in 3 steps as shown in Figure 3. Respondents are asked follow-up questions if a surveyor decides there is an issue with the replies. However, some of these questions are mistakenly asked or unrelated to the answers to the questionnaire. Also, there are differences when deciding if there is an issue with a puzzling questionnaire because there is more than one surveyor who is responsible for reviewing questionnaires.

Follow-up questions of questionnaires from the issue-found class are collected. Contextual word embeddings of BERT can be used in a way that semantically similar text embeddings are grouped close to each other in vector space. While this property of

Table 1: Classification results of two different loss function setups of the same model architecture at Figure 2. The model with BCE has only one output activated by a Sigmoid function and uses the Binary Cross Entropy loss function.

Loss Function	Class	Prec.	Recall	F1
BCE	0	0.885	**0.921**	0.903
	1	**0.753**	0.668	0.708
CE	0	**0.904**	0.903	**0.904**
	1	0.733	**0.732**	**0.733**

computed word representations is useful for classification and other downstream tasks in NLP, it also provides the basis for clustering. Thus, questions are represented in vector space by using the Sentence-BERT (Reimers and Gurevych, 2019) framework by using the BERTurk model with mean pooling.

The questions are clustered after reducing the dimensionality of their representations in vector space by using the principal component analysis technique (Jolliffe, 1986). A hierarchical clustering algorithm, agglomerative clustering, is used. The algorithm recursively groups the pair of vectors to find clusters until all representations are assigned to a cluster. We used the algorithm with a distance threshold without specifying a cluster size. Cosine distance is used for obtaining clusters that contextually represent the questions.

The clusters that included the mentioned type of follow-up questions are flagged after the clustering step. We assumed that the inconsistent examples that are not clustered can be found by predicting the cluster of their follow-up questions and checking if the clusters are flagged. However, this approach can only be used if follow-up questions exist for a questionnaire. Thus, inconsistencies in questionnaires without any issue could not be detected with this approach. To handle this problem, the Cleanlab framework (Northcutt et al., 2021) is used to find labelling errors in data. However, there was no empirical improvement in the model due to the aforementioned ambiguous answers in our dataset which can even be puzzling for a human expert. Thus, the proposed methods might still struggle with the inconsistencies even though they are reduced by the clustering approach and some noisy labels could be overlooked.

6 EXPERIMENTS & RESULTS

6.1 Experiment Setup

Experiments are performed to observe the results of handling label errors with a clustering approach on the dataset. The only dataset used in the experiments is the Turkish questionnaire dataset on the banking domain from Yapı Kredi, mentioned in Section 3. The setting of the dataset we used involves using only open-ended questions and their answers from a respondent to classify ambiguous issues via binary classification. Because of the unique properties of the problem and the dataset we used, there is not an available open benchmark dataset for our work.

From the point of view of model architecture, two output neurons are used. However, it is more common to use BCE loss function to calculate loss by using the output of a single neuron after a Sigmoid activation function. We observed using the cross entropy loss function L_{CE} like a multi-class classification problem setting improved the overall performance and recall value for Class 1, as known as the issue-found class, as shown in Table 1. Normally, there is no difference when using either loss function in a binary setting except for the efficiency of using BCE. However, the added complexity of using two output neurons instead of one aids model parameters to converge better for obtaining higher recall scores. Due to recall being more important than other evaluation metrics for this work, the model architecture with the cross-entropy loss L_{CE} is used. Evaluation metrics used in experiments are the common metrics for binary classification, such as precision, recall, F1 score, and AUC score. Macro average evaluation metrics are used due to the dataset being imbalanced. A cleaned test dataset could not be prepared due to manual force not being feasible because of ambiguity. As a result, the original test data is used. Also, the test data that is cleaned by the clustering approach is used to show how the results of the trained models differ for both test datasets.

Various language models are utilized for classification in a fine-tuning setting by using a linear layer as a classifier layer to benchmark different language models on our dataset. mBERT, DistilBERT (Sanh et al., 2019), ELECTRA (Clark et al., 2020), ConvBERT (Jiang et al., 2020), and BERTurk (Schweter, 2020) models from the BERTurk repository are used for this experiment. A parameter-free classification method that uses a compressor (Jiang et al., 2023) is chosen to compare its result with pre-trained language models. This approach is denoted as `gzip` with respect to the compression application that is used. `gzip` utilizes the k-nearest neighbors algorithm where $k = 3$. The pre-trained language model that is used in the model architecture of this work is pre-trained in the banking domain before fine-tuning for the classification task as mentioned in Subsection 4.1. This model will be denoted as `BERT` in this section for convenience. For this experiment only, the models

are trained in the dataset where the examples with empty answers are removed. Sometimes empty answers might be a reason for asking follow-up questions and a question that has no answer can have a linkage with other questions. Thus, the dataset where examples with empty answers are removed is easier to classify compared to the original dataset.

Two different model architectures are chosen to experiment clustering approach. The first one is a BERT with a classifier head that has only one linear network layer (i.e., output layer) denoted as `BERT+L`. The other model is the proposed model architecture where a BiLSTM layer is used as a hidden layer before the output layer and denoted by `+ BiLSTM`. The models trained on the cleaned version of train data are marked with an asterisk character on tables and the following subsection.

6.2 Classifying Results

Table 2: Classification results of using different pre-trained language models and a parameter-free approach.

Model	Acc.	Macro Avg.		
		Prec.	Recall	F1
ConvBERT	0.784	0.777	0.758	0.764
ELECTRA	0.782	0.773	0.758	0.764
DistilBERT	0.755	0.741	0.740	0.741
mBERT	0.784	0.775	**0.763**	0.768
gzip	0.672	0.656	0.614	0.612
BERTurk	0.782	0.775	0.755	0.762
BERT	0.791	0.791	0.760	**0.769**
+ BiLSTM	**0.792**	**0.793**	0.760	**0.769**

Results of using different pre-trained languages have similar results except for the DistilBERT where there is a minor difference with other models, as shown in Table 2. This is expected due to the smaller parameter size of the DistilBERT model. There is a significant difference between the parameter-free approach `gzip` and pre-trained language models. This is anticipated due to the complexity of the task, yet this approach is proven to be successful on less complex text-classification tasks (Jiang et al., 2023) and shows promising results with regard to having no training phase and GPU force. The best result is yielded by the models that use the BERT model that is pre-trained in the banking domain. Removing the examples with empty answers from the dataset helps models to perform slightly better at classification compared to results in Table 3.

The `+ BiLSTM*` slightly improves the recall value, as shown in Table 3. Results of AUC scores of each model especially show the classification abilities of the models. While BERT models without a BiLSTM layer in their classifier heads show poorer results, cleaning the data increases the classification ability on the original test data, as shown in Figure 4. The same models are also tested on cleaned test data. The `BERT+L` model has higher metric scores compared to the other models except for recall as shown in Table 4 on cleaned test data. However, the ROC curve analysis of the `BERT+L` model shows that the model fails to differentiate the classes due to the AUC score being under the value of 0.5. It can be deduced that `+ BiLSTM*` outperforms other models by generalizing the given data better and more confident than other models to decide whether there is an issue with a questionnaire.

Table 3: Model results on the original test data. Models with * are trained in cleaned train data.

Model	Acc.	Macro Avg.		
		Prec.	Recall	F1
BERT+L	**0.743**	0.741	0.718	0.723
BERT+L*	0.737	0.732	0.714	0.719
+ BiLSTM	**0.743**	**0.744**	0.715	0.721
+ BiLSTM*	0.737	0.729	**0.728**	**0.728**

Table 4: Model results on the cleaned test data.

Model	Acc.	Macro Avg.		
		Prec.	Recall	F1
BERT+L	**0.857**	**0.825**	0.792	**0.806**
BERT+L*	0.846	0.807	0.789	0.797
+ BiLSTM	0.852	0.815	0.796	0.804
+ BiLSTM*	0.833	0.786	**0.819**	0.798

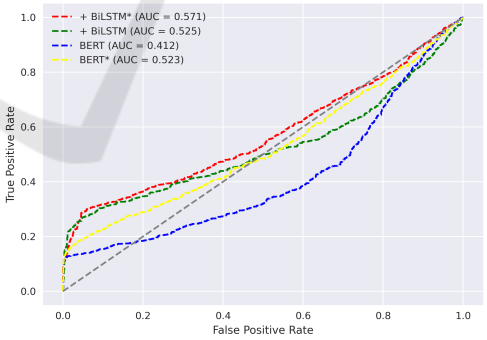

Figure 4: Area under the ROC Curve score of the models on original test data.

7 CONCLUSIONS & FUTURE WORK

By collecting Turkish questionnaire data from respondents' emails and extracting question-answer pairs, we created a customer questionnaire dataset to train a model on the text classification setting. A novel ap-

proach is proposed for questionnaire classification by using concatenated question-answers pairs to perform text classification rather than separately analysing the pairs. A pre-trained BERT is used in the training to get contextual and bidirectional word embeddings to capture the correlation between the pairs in the whole text. A BiLSTM layer on top of BERT is used to represent the sequential dependencies of word embeddings for further improvement. We utilized a data-centric approach, using clustering to group inconsistent data to mitigate the effects of noise caused by open-ended questions that provide deeper insights into a questionnaire and affect the annotation process. The model architecture we proposed for questionnaire classification performed better than a simple text classification architecture. Also, we have observed meaningful improvement in the classification performance with models trained on the data where the clustering approach is applied.

The proposed novel approach for classification can be used in a dataset in a similar setting that has multiple question-answer pairs with the task of classifying these pairs as a single unit and not as separate parts. The method we used doesn't involve any domain-centric or language-centric technique, thus one can assume the methods are applicable to similar data in other contexts or languages. Our work focuses on Turkish data in the banking domain due to not having any public data available. However, the results prove the classification is successful in a noisy dataset that is labelled without supervision.

For future research, we intend to experiment with semi-supervised methods like self-learning to lessen the impact of incorrect labels. This will help us to cover the examples in our dataset that our approach could not affect. We also believe data-centric approaches will improve NLP applications, especially for low-resource languages like Turkish. And using a data-centric approach to handle inconsistent data will further help in situations where manual labour is not feasible. For further work, we aim to develop our method using Explainable AI approaches to understand which question-answer pair mostly contributed to the outcome.

REFERENCES

Bhowmick, A. and Hazarika, S. M. (2016). Machine learning for e-mail spam filtering: Review,techniques and trends.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Jiang, Z., Yang, M., Tsirlin, M., Tang, R., Dai, Y., and Lin, J. (2023). "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828.

Jiang, Z.-H., Yu, W., Zhou, D., Chen, Y., Feng, J., and Yan, S. (2020). Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer-Verlag, Berlin; New York.

Kaliyar, R. K., Goswami, A., and Narang, P. (2021). Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, 80(8):11765–11788.

Larson, S. and Leach, K. (2022). A survey of intent classification and slot-filling datasets for task-oriented dialog.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Munikar, M., Shakya, S., and Shrestha, A. (2019). Fine-grained sentiment classification using bert. *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, 1:1–5.

Northcutt, C., Jiang, L., and Chuang, I. (2021). Confident learning: Estimating uncertainty in dataset labels. *J. Artif. Int. Res.*, 70:1373–1411.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schweter, S. (2020). Berturk - bert models for turkish.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In Sun, M., Huang, X., Ji, H., Liu, Z., and Liu, Y., editors, *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Tejwani, R. (2014). Sentiment analysis: A survey.

Zhu, D., Hedderich, M. A., Zhai, F., Adelani, D. I., and Klakow, D. (2022). Is bert robust to label noise? a study on learning with noisy labels in text classification. *Insights 2022*, page 62.