Large Scale Intent Detection in Turkish Short Sentences with Contextual Word Embeddings

Enes Burak Dündar, Osman Fatih Kılıç, Tolga Çekiç, Yusufcan Manav and Onur Deniz
Natural Language Processing Department, Yapi Kredi Technology, Istanbul, Turkey

Keywords: Intent Detection, Text Classification, Chatbot, Language Modeling.

Abstract: We have developed a large-scale intent detection method for our Turkish conversation system in banking domain to understand the problems of our customers. Recent advancements in natural language processing(NLP) have allowed machines to understand the words in a context by using their low dimensional vector representations a.k.a. contextual word embeddings. Thus, we have decided to use two language model architectures that provide contextual embeddings: ELMo and BERT. We trained ELMo on Turkish corpora while we used a pretrained Turkish BERT model. To evaluate these models on an intent classification task, we have collected and annotated 6453 customer messages in 148 intents. Furthermore, another Turkish document classification dataset named Kemik News are used for comparing our method with the state-of-the-art models. Experimental results have shown that using contextual word embeddings boost Turkish document classification performance on various tasks. Moreover, converting Turkish characters to English counterparts results in a slightly better performance. Lastly, an experiment is conducted to find out which BERT layer is more effective to use for intent classification task.

1 INTRODUCTION

Intent detection is a problem where messages are assigned to a set of topics. In order to solve this problem, messages are first converted to the features. These features are then fed to a classifier. Intent detection can be considered as a text classification task. Moreover, this concept is widely utilized in conversational systems. In conversational systems, there are two sides: an agent and a user. Therefore, It is highly important to have a robust intent detection method for making users feel they are talking to people not a machine.

We have developed a large scale intent detection method for a Turkish chatbot in banking domain. A chatbot should be able to understand user messages so that users can express their problems in many different ways. The messages they write have different characteristics. They are generally short and about more than one topics. Therefore, it becomes a bit harder to understand messages that do not have too much distinctive information.

In (Dündar et al., 2018), a hybrid method is proposed to understand what bank customers want to tell about their problems in Turkish. The hybrid method contains two approaches: a character based string similarity and a cosine similarity between word vectors. These similarity metrics are used for selecting the most similar question among a set for a user question if their similarity score is greater than a predefined threshold value.

In (Shridhar et al., 2019), authors have proposed a subword semantic hashing approach for representing texts for intent detection. Thereby, it deals with the out-of-vocabulary words since they are not handled by word level language models such as Word2Vec(Mikolov et al., 2013) and GloVe(Pennington et al., 2014). Also, the authors mention that it deals with spelling errors by looking words at subword level. They have conducted experiments on small datasets: The Chatbot Corpus, The AskUbuntu Corpus, and The Web Applications Corpus. They have 206, 190, and 100 samples and 2, 5 and 8 intents respectively.

In (Liu and Lane, 2016), an attention-based RNN model has been proposed in order to solve intent detection and slot filling jointly. They also investigated alignment-based RNN models since the slot filling task requires an alignment between inputs and outputs. The authors have used a spoken language understanding dataset which contains approximately 6k utterances for 18 different intents. Their proposed

Dündar, E., Kılıç, O., Çekiç, T., Manav, Y. and Deniz, O.
Large Scale Intent Detection in Turkish Short Sentences with Contextual Word Embeddings.
DOI: 10.5220/0010108301870192
In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2020) - Volume 1: KDIR*, pages 187-192
ISBN: 978-989-758-474-9

method shows state-of-the-art performance for this dataset.

In the last decade, many language models have been developed. Context independent language models such as Word2Vec(Mikolov et al., 2013), GloVe(Pennington et al., 2014), and fastText(Bojanowski et al., 2017) have shown promising results on various tasks such as machine translation, named entity recognition, and so on. Word2vec generate word representations by looking its neighbouring words. Therefore, it is too focused on local diversity, and can miss relations in abstract level. On the other hand, GloVe consider global co-occurences by representing words. So, it differs from Word2Vec in this way. Unlike these two methods, fastText is capable of handling subword information. Therefore, it is also convenient to use for out-of-vocabulary words. In (Dündar and Alpaydın, 2019), it is shown that fastText is better for representing syntactic information in Turkish, a morphologically rich language.

Thanks to advancements of computation power, it has become possible to train much larger models like ELMo (Peters et al., 2018), BERT(Kenton et al., 2018), and so on whereas context information is handled. Turkish is a low-resource but morphologically rich language compared to English. And, the number of studies are also less. Therefore, we have also investigated Turkish text classification methods in literature to understand the language specific problems.

In (Dündar and Alpaydın, 2019), language models: word2vec, fastText, and ELMo are trained on Turkish corpora. Authors have conducted experiments on document classification and word analogy tasks. They have analyzed word embedding dimensionality and corpora effect on word representations by learning different language models. In (Sen and Erdogan, 2014), word2vec is trained on Turkish and tested on analogy tasks. Authors investigated the effect of word embedding spaces in different sizes on these tasks.

In (Ayata et al., 2017), authors propose a sentiment classification model for Turkish tweets about different topics. They have trained a word embedding model in order to vectorizer these tweets. Then, two classifiers, Support Vector Machine and Random Forests, are used to decide their sentiments.

In (Sogancioglu et al.,), a two-stage classification method is proposed to multi-label classification of texts on banking domain. At the first stage, a binary classifier is used to determine whether or not a message is about daily language. If it is not about daily language, then it is classified by the second stage method. That study does not utilize any language models but traditional methods to vectorize texts like bag-of-words. The number of intents is 9 which is too few compared to our dataset.

In this study, we collected an intent detection dataset on Turkish banking domain. Furthermore, we have trained ELMo language model on Turkish corpora. Also, we used a HuggingFace's Transformer library(Wolf et al., 2019) to conduct experiments on BERTurk, a Turkish BERT model(Schweter, 2020).

Finally, a classifier model is trained on features obtained by these language models to assign a message to a intent. We have shown that it becomes possible to understand intents of short texts with large scale intent categories even without using too complicated classifiers.

The rest of the paper is organized as follows. In Section 2, our classifier models are presented. In Section 3, experimental results are shown. Also contextual word embedding models are discussed and compared with each other. In Section 4, we summarize the study, evaluate the results, and discuss about future works.

2 METHODOLOGY

2.1 Corpus

We have created a huge Turkish corpus by combining 3 different corpora: Boğaziçi Web Corpus(Sak et al., 2008), Turkish Wikipedia dump, and webchat messages in banking domain. Boğaziçi web corpus contains nearly half a billion words. It is created by crawling three newspapers and several web pages. Additionally, Turkish Wikipedia dump contains more than 50 million words. Our own webchat corpus contains approximately 10 million dialogues between customers and customer representatives. Total number of messages in these dialogues is more than 100 million. Thereby, we have created a corpus consisting of nearly 650 million words.

2.2 Contextual Word Embeddings

As mentioned in Section 1, Word2vec and GloVe does not consider subword information but FastText. Moreover, they are not capable of handling contextual information when representing words. Therefore, they fail on representations of polysemous words. On the other hand, ELMo and BERT which are mentioned in Section 2.2.1 and 2.2.2 can deal with this issue.

2.2.1 ELMo

A bidirectional language model, named ELMo(Peters et al., 2018), has been developed for handling contextual information in sequences. It has created a huge impact on NLP community since using context for creating word embeddings has never been applied so well before. ELMo has shown that it can effectively create quite good representations for the same word when it is used for different contexts. Contrary to token based language models, ELMo is capable of handling words in character level. Thereby, it can create representations for words event if they are not seen during training.

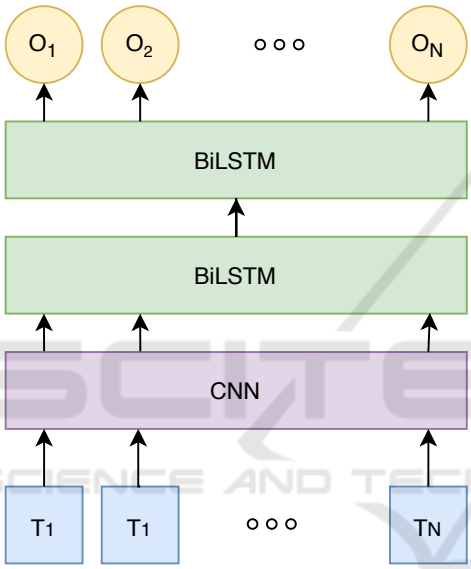

Figure 1: ELMo architecture used in this study.

ELMo model has 3 layers: a convolutional neural network(CNN) and two bidirectional long short term memory(biLSTM) layers. At first, CNN layer is used to create context independent representations for words in character level. Then, they are fed to biLSTM layers so that it can obtain word representations by considering the contextual information. Finally, it predicts the target words. We used the original ELMo architecture mentioned above in our study, and it is shown in Figure 1. T_n represents words. O_n represents the outputs of the last biLSTM layer.

2.2.2 BERT

BERT is a transformer based language model trained in unsupervised way (Kenton et al., 2018). The general architecture of this model is presented in Figure 2. T_n represents sub-words. O_n represents the outputs of the last transformer layer. The task used to pretrain the language model is to predict masked words in sequences fed to itself by combining features from left and right contexts. BERT has an unsupervised tokenizer that means tokenizer is learnt by a corpus before training. It can also split words into small pieces. Thereby, the number of tokens of a text is greater than the number of words.

Special tokens [CLS] and [SEP] are added into sequences. [CLS] is used to denote the beginning of a sequence while [SEP] is put at the end of a sequence. Also, the model learns whether or not the next sentence is correct. Thus, it can create a better representation for sequences. Authors have showed that it outperforms ELMo on several tasks such as question answering, sentence classification, and so on.

Compared to ELMo, BERT uses multi-head attention mechanism. That means it is not stateful. Therefore, it requires more computational power to train on large datasets.

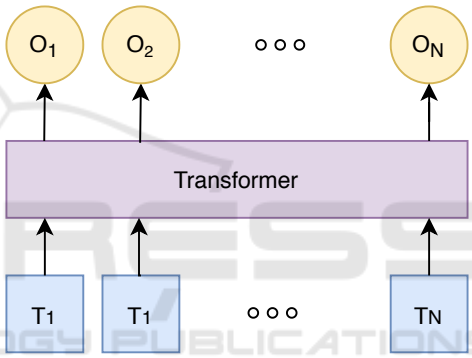

Figure 2: BERT architecture used in this study.

3 EXPERIMENTS

We have utilized 2 GPUs(GeForce GTX 1080 Ti) for training language models and classifiers on a Linux machine. Keras(Chollet et al., 2015) with Tensorflow(Abadi et al., 2015) backend, Pytorch(Paszke et al., 2019) and HuggingFace(Wolf et al., 2019) libraries are utilized while developing models. During experiments, we have normalized words by lowercasing and converting Turkish characters to English ones. Experiment using normalized texts are denoted in "Type" column as "norm". Detailed explanations about models are given in Section 3.2.

3.1 Datasets

3.1.1 Intent Sentences

When we deployed our chatbot platform, we had a dataset of predefined intent messages. Some of these

messages are manually written or collected from webchat conversations mentioned in Section 2.1. To increase our dataset, incoming messages have been periodically analyzed and annotated. If there are lots of messages related to a specific intent which is not covered in our predefined intents, then a new intent is created with these messages. Since labels are not balanced, we subsample 20-50 messages for each intent. After all, the dataset contains 6453 messages for 148 intents assigned by annotators.

3.1.2 Kemik News

Kemik News is one of the most popular datasets created in Turkish. It contains approximately 20k documents for 7 classes. Documents are not distributed to these classes equally. Therefore, we splitted the dataset so that label distributions are similar among train,validation, and test sets.

3.2 Models

In our experiments, we used two language models: ELMo and BERT and a traditional method named TF-IDF to extract features of texts. ELMo is trained on Turkish corpora mentioned in Section 2.1. Therefore, it is suitable to use with preprocessed texts mentioned in the beginning of this section. In the experiments, 3 metrics have been determined: precision(P), recall(R), and F1 score.

For ELMo model, we have used the original architecture developed by Allen NLP(Peters et al., 2018). It consists of one context-independent and two context-dependent layers. In all experiments, we concatenated features obtained from these layers. Each layer represents words in 512 dimensional space, concatenated features are 1536.

Two different classifiers are put on these features. In the first one, they are fed to a bidirectional GRU layer followed by a GlobalMaxPooling and a dense layer with softmax activation. The second classifier model does not have a bidirectional GRU layer. Therefore, it can be considered as a linear classifier. In Figure 4, the classifier model architecture is shown. E represents token embeddings.

For BERT experiments, BERTurk(Schweter, 2020), a public HuggingFace model is utilized. It is trained on a corpus(44,04,976,662 tokens) which is a much larger than the corpus mentioned in Section 2.1. The first token feature of a sequence at the last layer is selected for representing a text. It is a special token, [CLS] which is used in BERT. Then, its features are fed to a dense layer with softmax activation. In Figure 4, the classifier model architecture is shown. E_n represents token embeddings. O_n represents the

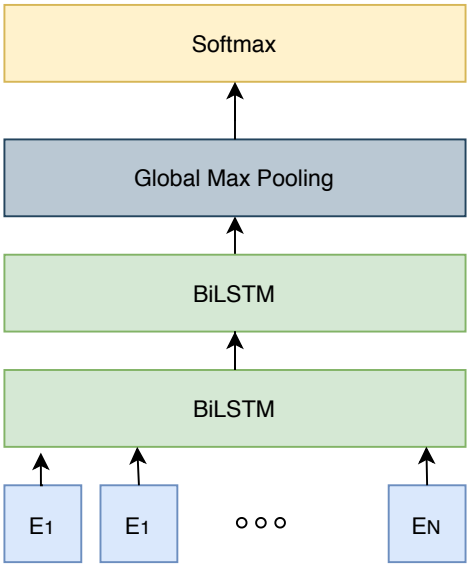

Figure 3: ELMo classifier architecture for experiments.

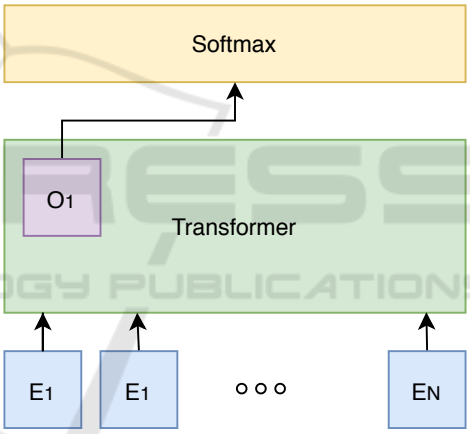

Figure 4: BERT classifier architecture for experiments.

outputs of the last transformer layer. O_1 which is the output representation of special token([CLS]) is fed to a linear classifier.

3.2.1 Intent Detection

Intent detection results are represented in Table 1. In the models column, a multinomial naive bayes and a support vectors machine classifiers are used for baseline. They are fed with TF-IDF features. In the type column, "N" denotes whether or not texts are normalized. ELMo model is trained with a normalized corpus. Therefore, the classifier on top of it is fed with normalized texts. Moreover, "F" denotes when weights of BERT model is frozen. And, "L" in the type column denotes features from the last ELMO layer is used. When we look at the re-

sults of MNB and SVM, normalizing text shows better performance. Using a linear classifier on top of the ELMo model, named ELMo+L, reduces its performance. Concatenating representations of the last three ELMo layers results in better scores. Also, it can be clearly seen that BERT results are better than ELMo. Furthermore, freezing BERT weights results in a better performance.

Table 1: Intent classification results.

Models	Type	P	R	F1
MNB		0.6847	0.6746	0.6437
MNB	N	0.7107	0.6963	0.6674
SVM		0.7291	0.7190	0.7046
SVM	N	0.7547	0.7386	0.7230
ELMo	N	0.9019	0.8905	0.8866
ELMo	N+L	0.8963	0.8832	0.8799
ELMo+L	N	0.8769	0.8471	0.8441
ELMo+L	N+L	0.8625	0.8450	0.8399
BERT		0.9084	0.8988	0.8941
BERT	F	0.9133	0.9028	0.8995

3.2.2 Text Classification

The same models and feature types in intent detection are also used for this task. The results of Kemik News classification task are shown in Table 2. The best result is obtained by ELMo model even if it is trained with much smaller corpus contrast to the one used for training BERT. Documents used in these experiments are much longer. This property of the dataset can cause this performance difference between ELMo and BERT since bidirectional GRU layers are not used in BERT. Also, the number of classes may also affect it. Similar to intent detection task, normalizing texts has resulted in a better performance. In (Dündar and Alpaydın, 2019), authors have conducted experiments on Kemik News dataset by using accuracy metric. Their best result has 94.44% accuracy with a model which represents documents by averaging their word embedding vectors.

3.2.3 Contributions of BERT Layers

We used the original BERT model consisting of 12 layers. Each layer contains information at a different level. Some of them may contain more abstract information while the other ones do not. Therefore, we have conducted an experiment to measure which layer is more convenient for our task. BERT Layer performances in terms of precision, recall and F1 score on intent detection task are shown in Table 3.

We find out the last layer is the best one. Although there seems to be a correlation between layer levels and performance, layer 5 scores are better than layer 6. In all our experiments, we have used BERT features extracted by this layer. Also, the maximum sequence length is set to 256. During the experiment, BERT layers are not frozen, and pooled outputs of each layer representations are fed to a linear classifier.

Table 2: Kemik News classification results.

Models	Type	P	R	F1
MNB		0.7460	0.6348	0.5690
MNB	N	0.7454	0.6349	0.5691
SVM		0.8995	0.8979	0.8919
SVM	N	0.9021	0.9007	0.8949
ELMo	N	0.9567	0.9568	0.9567
ELMo	N+L	0.9562	0.9562	0.9562
ELMo+L	N	0.9395	0.9391	0.9388
ELMo+L	N+L	0.9399	0.9400	0.9399
BERT		0.9550	0.9551	0.9549
BERT	F	0.9540	0.9542	0.9539

Table 3: Which BERT layer is more useful for intent classification?

Layer	P	R	F1
1	0.7528	0.7479	0.7269
2	0.8363	0.8161	0.8081
3	0.8593	0.8409	0.8335
4	0.8754	0.8605	0.8535
5	0.8873	0.8729	0.8681
6	0.8853	0.8719	0.8657
7	0.8865	0.875	0.8704
8	0.8896	0.8781	0.8724
9	0.8969	0.8843	0.8802
10	0.9001	0.8895	0.8853
11	0.9052	0.8926	0.8892
12	0.9084	0.8988	0.8941

4 CONCLUSIONS

In this study, we have shown that using contextual word embeddings cause the performance of the text classification tasks to improve. Experiments have shown that it is not possible to differentiate performances of BERT and ELMo on Turkish text classification tasks in terms of precision, recall and F1 score. On the other hand, document lengths play an important role for selecting the appropriate model architecture since we have shown that longer documents are classified better with a model having RNN structure. In the future, the recent language models such as ALBERT(Lan et al., 2019), RoBERTa(Liu et al., 2019), XLNet(Yang et al., 2019) and ELECTRA(Clark et al.,

2020) can be applied to this study. Moreover, uncertainty analysis can be applied to large scale intent detection task since there are too many categories that are difficult to distinguish from each other.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ayata, D., Saraçlar, M., and Özgür, A. (2017). Turkish tweet sentiment analysis with word embedding and machine learning. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

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.

Chollet, F. et al. (2015). Keras. https://keras.io.

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

Dündar, E. B. and Alpaydın, E. (2019). Learning word representations with deep neural networks for turkish. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

Dündar, E. B., Çekiç, T., Deniz, O., and Arslan, S. (2018). A hybrid approach to question-answering for a banking chatbot on turkish: Extending keywords with embedding vectors. In *KDIR*, pages 169–175.

Kenton, J. D. M.-W. C., Kristina, L., and Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Liu, B. and Lane, I. (2016). Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Sak, H., Güngör, T., and Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *International Conference on Natural Language Processing*, pages 417–427. Springer.

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

Sen, M. U. and Erdogan, H. (2014). Learning word representations for turkish. In *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, pages 1742–1745. IEEE.

Shridhar, K., Dash, A., Sahu, A., Pihlgren, G. G., Alonso, P., Pondenkandath, V., Kovács, G., Simistira, F., and Liwicki, M. (2019). Subword semantic hashing for intent classification on small datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE.

Sogancioglu, G., Köroğlu, B. A., and Agin, O. Multilabel topic classification of turkish sentences using cascaded approach for dialog management system.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.