Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 844 – 859

doi:10.55730/1300-0632.4021

Joint intent detection and slot filling for Turkish natural language understanding

Osman BÜYÜK[1,2*]

[1]Department of Electrical and Electronics Engineering, İzmir Demokrasi University, İzmir, Turkiye
[2]Sestek Speech Enabled Software Technologies Inc., Department of Research and Development, İstanbul, Turkiye

Received: 13.03.2023 • **Accepted/Published Online:** 01.09.2023 • **Final Version:** 29.09.2023

Abstract: Intent detection and slot filling are two crucial subtasks of a text-based goal-oriented dialogue system. In a goal-oriented dialogue system, users interact with the system to complete a goal (or to fulfill their intent) and provide the necessary information (slot values) to achieve that goal. Therefore, a user's text input includes information about the user's intent and contains required slot values. Recently, joint models that simultaneously detect the intent and extract the slots are proposed to benefit from the interaction between the two tasks. The proposed methods are usually tested using benchmark data sets in English such as ATIS and SNIPS. Intent detection and slot filling problems are much less studied for the Turkish language mainly due to the lack of publicly available Turkish data sets. In this paper, we translate ATIS in English to Turkish and report intent detection and slot filling accuracies of several different joint models for the translated data set. We publicly share the Turkish ATIS data set to accelerate the research on the tasks. In our experiments, the best performance is obtained with the state-of-the-art bidirectional encoder representations from a transformers (BERT) based model. The BERT model is trained using a combination of intent detection and slot filling losses to jointly optimize a single model for both tasks. We achieved 96.54% intent detection accuracy and 91.56% slot filling F1 for the Turkish language. These accuracies significantly improve (7% absolute in slot filling F1) previously reported results for the same tasks in Turkish. On the other hand, we observe that the accuracy in Turkish is still slightly lower compared to the accuracy in English counterparts. This observation indicates that there is still room for improvement in the results for Turkish.

Key words: Intent detection, slot filling, natural language understanding, goal-oriented dialogue systems

1. Introduction

Recent advances in communication technologies have led to the deployment of automatic dialogue systems for natural interaction between humans and computers. The COVID-19 pandemic has also accelerated this progress since the number of online activities have been dramatically increased due to the restrictions in the pandemic. An automatic dialogue system enables the users to ask for information, to conduct transactions or to complete a reservation [1]. A goal-oriented dialogue system is usually described for a specific domain and has a fixed set of entities that a user might inquire about [2].

A goal-oriented dialogue system can be decomposed into three modules: i) a natural language understanding (NLU) module to convey the information in the user input, ii) a dialogue management (DM) module to decide on the next action, iii) a natural language generation (NLG) module to generate a response [3]. The

*Correspondence: osman.buyuk@idu.edu.tr

natural language understanding (NLU) module is a vital component of a goal-oriented dialogue system which detects the intent of the user and also extracts the relevant information from the text input. For example, a user's text input to an automatic airline travel reservation system might be "I want to go from Istanbul to New York on Wednesday". The NLU module classifies the intent of the user as "flight" and extracts the relevant information such as "Istanbul" (departure city name) "New York" (arrival city name) and Wednesday (day of the departure) in the sentence. The extracted information is used to fill slots for the successful completion of the requested reservation. These two subtasks of an NLU module are referred to as intent detection and slot filling, respectively.

Intent detection is a text classification problem in which the user's input text is classified as one of the predefined intent classes. In the slot filling, a slot label is assigned to each word in the input. The slot filling can be viewed as a sequence-to-sequence mapping problem in which the length of the input and output sequences are the same. The two subtasks of the NLU module can be modeled independently [4]. However, recently joint models are proposed that take the user input and perform the intent detection and slot filling simultaneously [4–10]. Thanks to the joint modeling, the two tasks can be performed with a single model in one pass. Additionally, the joint models consider the correlation between the two tasks [10] and might lead to performance improvement. On the other hand, they require a data set in which both intent and slot labels should be available.

Several different approaches are proposed for intent detection, slot filling and joint modeling of the tasks. The proposed methods usually follow the developments in the text classification field. With the advances in deep neural networks (DNNs), conventional DNNs are first applied for the tasks [6–8]. More recently, state-of-the-art transformer based architectures are also investigated [9, 10]. The approaches are usually tested using the benchmark data sets in English such as ATIS [11] and SNIPS [12]. In [6], a recurrent neural network - long-short term memory (RNN-LSTM) architecture is proposed for joint modeling of slot filling and intent detection. In the method, a slot label is predicted for each word in the input utterance. Moreover, an additional 'EOS' is added to the end of the utterance to infer the intent label. In [7], an encoder-decoder RNN framework with attention is proposed for the joint modeling. In the study, several different strategies are investigated to utilize the alignment information. In the slot filling problem, the input and output sequences are aligned word by word, while the original attention mechanism may not provide the exact alignment. In [8], a focus mechanism is introduced in RNN encoder-decoder framework in which the decoder only considers the aligned hidden states of the encoder block. In [4], a fast nonautoregressive model is used. The proposed method uses attention and feed-forward neural network blocks to perform the tasks simultaneously. The decoding is performed in two-passes to fix the errors in the first pass. In [10], a cointeractive transformer framework is proposed to consider the cross-impact between the tasks in both directions, that is from intent to slot and from slot to intent. Other approaches include the use of capsule neural networks [13] and graph LSTM [14] for the joint modeling.

Recently, new language representations such as BERT (bidirectional encoder representations from transformers) [15], Roberta (a robustly optimized BERT) [16], GPT (generative pretrained transformer) [17], XLnet [18] have been released based on the transformer architecture. The language representations are first pretrained on large-scale unlabeled corpora and then fine-tuned to the downstream task using a relatively small amount of task-specific labeled data. This pretraining and fine-tuning procedure has created state-of-the-art performances for a wide range of natural language processing (NLP) tasks. In [9], BERT is used for joint intent detection and slot filling. In the study, intent detection is performed using the vector representation of 'CLS' token which is added to the input sequence to indicate the beginning of an utterance. Vector representations of other

tokens are fed into a soft-max layer to classify over the slot labels. The network is fine-tuned end-to-end with a combined loss of intent detection and slot filling tasks.

Although intent detection and slot filling have been extensively studied for English, much less efforts have been devoted to evaluate intent detection [19–23] and slot filling [20, 24] performances for the Turkish language. This can be mainly attributed to the lack of publicly available Turkish data sets to compare the experiments performed in different institutes. Only recently, ATIS data set is translated to Turkish, and intent detection and slot filling performances are reported using a multi-language BERT model [20]. In the study, two separate models are trained for the two tasks without joint modeling. The same Turkish data set is used to compare conditional random fields (CRF) with several different pretrained transformer language representations for slot filling in [24]. To the best of our knowledge, no result has been reported for the joint modeling of intent detection and slot filling for Turkish, yet.

The data set in [20] is made available at https://github.com/avaapm/TurkishNamedEntityRecognition publicly. However, the translated data set contains some alignment problems between the words and slot labels which significantly degrade the performance [24]. Moreover, intent labels are not available in the publicly released version. The original ATIS is also translated to Turkish for named entity recognition (NER) in https://github.com/UniversalDependencies/UD_Turkish-Atis. In this version, only NER annotations are available without slot and intent labels. Therefore, we retranslate ATIS to Turkish using a semiautomatic translation procedure. In the procedure, we use the Google web interface to automatically translate utterances and perform manual corrections to fix the alignment problems and translation errors. The data set is shared publicly at our github repository[1]. We compare the performances of several different joint models in our Turkish ATIS data set. In the experiments for the Turkish ATIS, we achieved 96.54% intent detection accuracy and 91.56% slot filling F1 which are significantly higher than the previously published results for similar tasks and data sets in Turkish. The best performance is obtained with a BERT-based model when the model is fine-tuned using a combination of intent detection and slot filling losses to jointly optimize a single BERT model for both tasks.

The main contributions of our paper can be summarized as follows: i) we think that the Turkish ATIS data set may serve as a benchmark test set for future intent detection and slot filling studies in Turkish, ii) we present the state-of-the-art intent detection and slot filling performances in the Turkish ATIS data set which can help to evaluate the performance of newly proposed methods.

The remainder of our paper is organized as follows. In Section 2, we provide the details of the ATIS data set and describe our translation procedure. In Section 3, we provide a brief summary of the joint intent detection and slot filling methods used in our experiments. Experimental results are provided in Section 4. We conclude our paper with a summary of our key findings.

2. Data

2.1. ATIS data set

ATIS (Airline Travel Information Systems) is a widely used data set in NLU research. The data set contains audio recordings of users when making flight reservations. The data set consists of 26 intent and 129 slot labels. We use the same train, validation and test partitions as in [10] with 4478/500/893 utterances, respectively. After splitting the data set into train/validation/test subsets, we realized that there exist some labels which do

[1] https://github.com/obu80/Turkish_ATIS

not appear in the training set. These labels (5 intent and 10 slot labels) are removed from the data set and the experiments are conducted with 21 intent and 119 slot labels. Moreover, the validation and test sentences which contain any of these labels are removed from the sets. As a result, 2 validation and 11 test sentences are taken out and the experiments are performed with 498 validation and 882 test utterances for English.

Inside–outside–beginning (IOB) format is used to tag the slot labels in the original ATIS. In the IOB format, B-, I- and O- abbreviations are used to indicate the beginning, inside and outside of a slot, respectively. An example sentence from the data set is shown with its slot and intent labels in Table 1.

Table 1. An example sentence from the ATIS data set with its slot and intent labels.

Sentence	I	want	to	fly	from	Baltimore	to	Dallas	round	trip
Slots	O	O	O	O	O	B-fromloc.city	O	B-toloc.city	B-round_trip	I-round_trip
Intent	flight									

2.2. Turkish ATIS data set

We use Google translate web API to translate the sentences in the original ATIS to Turkish. After the translation, the number of words may increase/decrease and the order of the words may also change. In order to align the translated words with the correct slot labels, we employ the following semiautomatic procedure:

- We first list all slot values in the original ATIS. This results in a list of approximately 700-word phrases in English.
- We use Google web API to translate the phrases in the list to Turkish and prepare a look-up table for the original English and translated Turkish slot values.
- When we translate a sentence to Turkish:
  - We determine the slot values in the original English sentence and use the look-up table to find the corresponding Turkish slot values.
  - We check the Turkish slot values in the translated sentence.
    - If we find the Turkish slot values in the translated sentence, we do not perform any manual corrections. We assign the slot labels in the original English to the Turkish slots.
    - If we cannot find the Turkish slot values in the translated sentence, the sentence is manually checked.

With this procedure, approximately 80% of the sentences are automatically translated. For the remaining sentences, manual intervention is needed. During the manual intervention, we fix the alignment problems between the words and slot labels and/or correct translation errors if they exist. Sometimes we need to add multiple pronunciations to the words in the translation look up table since some words are translated differently by Google API when the slot value is provided with its context in the sentence. For example, the slot phrase 'one stop' in the look-up table has the following three alternatives 'bir durak', 'tek durak' and 'bir durağı' and the phrase 'noon' has four alternatives 'öğle vakti', 'öğlen', 'öğleden' and 'öğle'. The translation of the example English sentence in Table 1 is provided in Table 2.2.

Table 2. An example sentence from the Turkish ATIS data set with its slot and intent labels.

Sentence	Baltimore'dan	Dallas'a	gidiş	dönüş	uçmak	istiyorum
Slots	B-fromloc.city	B-toloc.city	B-round_trip	I-round_trip	O	O
Intent	flight					

In Table 2.2, we observe that the proper nouns 'Baltimore' and 'Dallas' are left untranslated in the Turkish sentence. This is the major limitation of the translation procedure. These untranslated words are usually much less presented in any Turkish text corpora. Therefore, their representations are obtained using an insufficient amount of training data. As we will see in Section 4, these untranslated proper nouns are one of the main factors that degrade the performance in the Turkish experiments.

The Turkish ATIS data set contains 26 intent labels as in the original set. However, we need to add extra 'I-' labels to the original list of slot labels after the translation. For example, the original slot value 'earliest' with a single word can be translated to Turkish with two words as 'en erken'. This might require adding a new slot label (I-arrive_time.period_mod) to the label set for the second word. As a result, there are 137 slot labels in the Turkish version. The data set is divided into the same train/validation/test splits as in the original English with 4478/500/893 utterances, respectively. Some statistics from the Turkish data set is provided in Table 3. Statistics in the second, third and fourth columns are only computed for the training samples. In order to present the lexical diversity of the Turkish ATIS, we provide type-token ratio (TTR) metric in the fourth column. TTR is computed by dividing the total number of unique words in the data set by the total number of words. TTR for the English ATIS is 0.0171. Relatively higher TTR in the Turkish might be attributed to the agglutinative nature of the Turkish language. For example, the number of unique words in the Turkish ATIS is approximately two times more than in the English version. As observed in the second column, the data set has an imbalanced number of samples per class. Additionally, we can observe from the third column that there are approximately 8 words on average in each utterance.

Table 3. Statistics from the Turkish ATIS data set.

	# of classes	Mean/std. dev. of # of samples per class	Mean/std. dev. of # of words per sample	Type-token ratio(TTR)
Slots	137	140.27/485.50	8.22/3.25	0.0477
Intent	26	176.92/648.55		

Similar to the original data set, there exist some labels which do not t appear in the training set after the split. These labels are removed from the data set. The number of removed slot and intent labels are 13 and 5, respectively. The experiments are performed with 124 slots and 21 intent labels for Turkish. Additionally, the validation and test sentences which contain any of these labels are not used in the experiments. As a result, 3 validation and 14 test sentences are removed from the sets and the experiments are performed with 497 validation and 879 test sentences.

We perform the experiment in Table 4 in order to evaluate the quality of translation. For the experiment, 100 sentences from the validation set are translated by a human translator[2]. The translator is a native Turkish speaker who speaks English as a second language. We compare the automatic translation outputs with the

[2]The translator graduated from the American Culture and Literature Department of Bilkent University, Turkey and works as an English teacher and translator.

reference human translations using rouge metric [25]. Rouge (recall-oriented understudy for gisting evaluation) is a set of metrics used for evaluating automatic summarization and machine translation applications in NLP. Rouge-N measures the number of matching 'n-grams' between the automatic translation hypothesis and the human reference. We use the huggingface implementation to compute the rouge metric[3]. We provide the mean of the precision, recall and F1 values for the rouge-1, rouge-2 and rougeL in Table 4. As observed in the table, the rouge-2 score is approximately 0.5. This indicates that almost half of bigrams in the automatic translation also appear in the reference. In Table 5, we give some example sentences from the validation set together with their translations. In the table, the human translations are indicated with 'Ref-trans' label. 'Auto-trans' translations are obtained with the semiautomatic procedure described in this section. As seen in the table, the quality of the automatic translations is close to the human's references.

Table 4. Precision, recall and F1 rouge scores of automatic translation outputs.

	Precision	Recall	F1
Rouge-1	0.740	0.738	0.735
Rouge-2	0.516	0.520	0.515
RougeL	0.671	0.670	0.667

Table 5. Some example sentences from the validation set together with their semiautomatic and human translations.

Original	i'm interested in a flight from pittsburgh to atlanta
Ref-trans	pittsburgh'dan atlanta'ya olan uçuşla ilgileniyorum
Auto-trans	pittsburgh'dan atlanta'ya bir uçuşla ilgileniyorum
Original	what does the abbreviation co mean
Ref-trans	co kısaltmasının anlamı nedir
Auto-trans	co kısaltması ne anlama geliyor
Original	show me the least expensive flight leaving miami on sunday after 12 o'clock noon and arriving cleveland
Ref-trans	pazar günü öğle 12'den sonra miami'den kalkan ve cleveland'a varan en ucuz uçuşu göster
Auto-Trans	bana pazar günü öğlen 12'den sonra miami'den kalkan ve cleveland'a varan en ucuz uçuşu göster
Original	show flights from denver to oakland arriving between 12 and 1 o'clock
Ref-trans	denver'dan oakland'a 12 ile 1 arası varacak olan uçuşları göster
Auto-trans	denver'dan oakland'a saat 12 1 arasında gelen uçuşları göster
Original	round trip fares from denver to philadelphia less than 1000 dollars
Ref-trans	denver'dan philadelphia'ya 1000 dolardan az olan gidiş dönüş sefer ücretleri
Auto-trans	denver'dan philadelphia'ya gidiş dönüş 1000 doların altında

3. Methodology

Many different models have been proposed to jointly model the intent detection and slot filling tasks in the literature. The English ATIS data set has also accelerated the research on the topic. RNN architecture is used

[3]https://github.com/huggingface/datasets/blob/main/metrics/rouge

in the initial DNN-based models [6–8]. More recently, state-of-the-art transformer based architectures are also applied [9, 10]. In this paper, we compare the performances of 7 different joint models which are highly cited in the literature. The first 5 models (BiRNN, Att-BiRNN, Att-BiRNN-Crf, Focus-BiRNN and Slot-Refine) are frequently used as a baseline in the recent studies for the English ATIS. The last two models (Inter-Trans and BERT-joint) present the state-of-the-art in the field. We use the same Glove word embeddings as the initial word embedding vectors when needed.

We prefer to run each model with its default parameters in general. On the other hand, some of the hyperparameters are tuned using the validation set of the ATIS data set. The implementation details of each model and the model parameters are described in the following subsections to enhance the reproducibility of our experiments.

3.1. Glove word embeddings

We feed the same Glove word embeddings [28] as initial word vectors to some of the joint approaches. The GloVe model is trained using a large text corpus on the nonzero entries of a global word-to-word cooccurrence matrix. The occurrence matrix calculates how frequently a word cooccurs with another [28]. For English, we use the uncased version of Glove in https://nlp.stanford.edu/projects/glove. The embeddings are 300 dimensional and trained from Wikipedia 2014 and Gigaword 5 corpora. The model has 400K vocabulary with approximately 1GB file size. For Turkish, we use the uncased version of Glove in https://github.com/inzva/Turkish-GloVe. The embeddings are 300 dimensional and are trained from 21 GB of text corpora. The model has 253K vocabulary with a file size of approximately 700MB.

3.2. Bi-directional RNN-LSTM (BiRNN)

In unidirectional RNNs, the information flows in one direction from left-to-right to capture the contextual information. On the other hand, bidirectional RNNs (BiRNNs) process the input in both directions so that the output layer can get information from past (backwards) and future (forward) states simultaneously. BiRNN includes an additional RNN layer which processes the input sequence backward (right-to-left). The output is obtained using a combination of the backward and forward layers.

In [6], a bidirectional RNN-LSTM network is used for joint modeling of intent detection and slot filling. In the method, an additional 'EOS' token is added to the end of each input utterance. An intent label is predicted for the 'EOS' token. Additionally, a slot label is predicted for each word in the utterance. We implement the BiRNN method using the implementation in https://github.com/yvchen/JointSLU. We use mini-batch stochastic gradient descent with a batch size of 10 examples and adagrad optimization as used in [6]. The hidden vector size and learning rate are set to 50 and 0.01, respectively. The input words are represented with 1-hot word vectors since this input representation gives competitive results compared to the other contextual word embeddings in [6]. We perform 100 epochs of training in the BiRNN method.

3.3. Bidirectional RNN with attention (Att-BiRNN)

In [7], an encoder-decoder framework with RNNs is used for the joint modeling. For slot filling, the final hidden state of an encoder bi-directional RNN is fed into a decoder which has unidirectional RNNs. In order to benefit from the alignment information between the words and slot labels, the hidden encoder state of each word is also fed into the corresponding decoder block. For intent detection, an additional decoder is used. The intent

detection decoder shares the same encoder with the slot filling. We implement Att-BiRNN method using the implementation in https://github.com/sz128/slot_filling_and_intent_detection_of_SLU.

We use one layer of LSTM as used in [7] in the experiments. The network parameters are uniformly initialized between [-0.2,0.2]. The hidden vector size and batch size are set to 100 and 64, respectively. We use a fixed learning of 0.01. The Glove embeddings in Section 3.1 are fed into the network as the initial vectors. We perform 100 epochs of training in the Att-BiRNN method.

3.4. Bidirectional RNN with conditional random fields (Att-BiRNN-Crf)

Slot label predictions are dependent on predictions for surrounding words. It has been shown that structured prediction models such as conditional random fields (CRF) [27] can be used to improve the slot filling performance [9, 26]. In CRFs, the predictions are modeled as a graphical model, which represents the presence of dependencies between the predictions. CRF assigns a transition score for each transition pattern between adjacent tags to ensure the best tag sequence of the utterance from all possible ones.

In this paper, we investigate the use of CRFs on top of the Att-BiRNN method as implemented in https://github.com/sz128/slot_filling_and_intent_detection_of_SLU. Similar to Att-BiRNN, we use the Glove embeddings as the initial word vectors. The other model parameters are kept the same with the Att-BiRNN method. The number of training epochs is 100.

3.5. Bi-directional RNN with focus (Focus-BiRNN)

In [8], a focus mechanism is proposed in the RNN-based encoder-decoder framework to fully utilize the alignment information. In the proposed method, the decoder only focuses on the aligned hidden states of the encoder. This method is referred to as Focus-BiRNN in this paper and is implemented using the same repository in Section 3.3. The model parameters and initial word embeddings are kept the same with the Att-BiRNN method. The number of training epochs is set to 100.

3.6. A two-pass nonautoregressive model (Slot-Refine)

In [4], a nonautoregressive model which consists of multi-head attention and feed-forward neural network blocks is proposed for the joint modeling. Since the proposed method is nonautoregressive it may produce uncoordinated slots (such as the beginning of a slot label (e.g., B-departure_date_day) is followed by inside of another label (e.g., I-to_cityname)). In order to handle such inconsistencies, slot detection is performed in two passes. The output of the first pass is fed into the second pass as the input.

We implement the Slot-Refine using its open-source code in https://github.com/moore3930/SlotRefine. The Glove word embeddings are used as the initial word vectors in the experiments. The batch size and learning rate are set to 32 and 0.0025, respectively. The number of transformer layers, attention heads, and hidden sizes are 2, 8, and 32. We perform 50 epochs of training for the Slot-Refine method and report the best performance metrics.

3.7. Cointeractive transformer model (Inter-Trans)

Recently in [10], a cointeractive transformer model is proposed to simultaneously consider the bidirectional connection between the intent detection and slot filling tasks in a unified framework. The proposed approach

has a cointeractive attention layer which allows the flow of information in both directions, from slot to intent and from intent to slot. This method is referred to as Inter-Trans.

We implement the Inter-Trans method using its implementation in https://github.com/kangbrilliant/DCA-Net. The GloVe vectors are used as the initial word embeddings in the experiments. The hidden units of the shared encoder and the cointeractive module are set to 128 as in the original paper [10]. The number of cointeractive modules is 2. Batch size and learning rate are 32 and 0.001, respectively. Adam optimization is used in the training [10] for 100 epochs.

3.8. Joint modeling with BERT (BERT-joint)

We also implement a BERT-based joint model similar to [9]. In the model, 'CLS' token's final embedding is passed through a single layer feed-forward neural network classifier to obtain the intent prediction scores. An intent loss L_I is computed using the intent scores. Similarly, a single layer classifier is added on top of the BERT model to perform the classification over the slot labels for each word token. A slot loss L_S is computed using the output of the slot classifier. We use cross entropy loss to compute both L_I and L_S. Then, a single BERT model is fine-tuned with the task-specific classifiers in an end-to-end fashion using a combined loss in Equation (1):

$$L_C = \alpha L_I + \beta L_S \tag{1}$$

Here, α and β are weights with $\alpha + \beta = 1$. In the experiments, we set $\alpha = 0.5$ and $\beta = 0.5$ since this combination yields a balanced performance for the two tasks.

The overall block diagram of the BERT-joint method is shown in Figure 1. As shown in the figure, the input to the BERT model is the word tokens (TOKEN_i) together with the special tokens indicating the beginning (BOS) and end (EOS) of utterance. The input tokens are passed through the BERT model to obtain the contextual token representations (REP_i) for each token. These token representations are processed by a dense layer to predict the intent (LAB_Intent) and slot (LAB_Sloti) labels.

In the BERT-joint model, we use BERT-base-uncased in https://huggingface.co/bert-base-uncased for English as the pretrained language representation. The file size of this model is 440MB. We use BERTurk-base-uncased in https://huggingface.co/dbmdz/bert-base-turkish-uncased for the experiments in Turkish. The file size of BERTurk-base-uncased (BERTurkish) is 445MB. We perform 10 epochs of training for both languages.

4. Experimental results

4.1. Evaluation metrics

In this study, we use sentence accuracy (Sent_Acc), intent detection accuracy (Intent_Acc), intent detection F1 (Intent_F1) and slot filling F1 (Slot_F1) metrics to evaluate the performance of the joint methods. The sentence accuracy shows the percentage of the sentences for which both slots and intent labels are correctly predicted. Intent detection accuracy represents the percentage of correctly predicted intents. We also report macro averaged intent detection F1 (Intent_F1) in order to observe the influence of data imbalance on the performance. The Intent_Acc and Intent_F1 are computed using sklearn library, https://scikit-learn.org/. In order to compute Slot_F1, we use huggingface's sequence labeling evaluation metric (seqeval) in https://huggingface.co/spaces/evaluate-metric/seqeval. The seqeval is used to evaluate the performance of chunking tasks such as named-entity recognition, part-of-speech tagging and semantic role labeling. It supports IOB format and can be used for measuring the performance of a system that has processed the CoNLL-2000

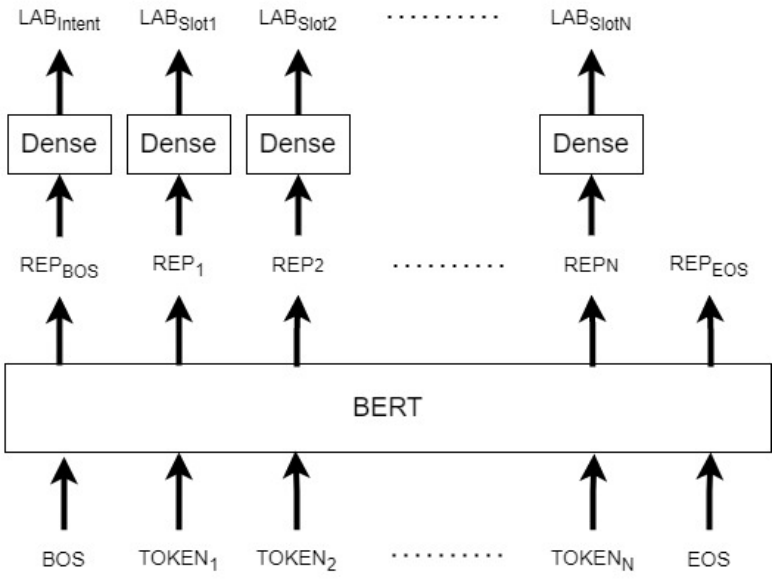

Figure 1. Block diagram of the BERT-Joint method

shared task data. We report the average F1 score in Table 6 and Table 7 which is the harmonic mean of the precision and recall.

4.2. Experimental results

In the experiments, we first determine the number of training epochs required to obtain the best validation performance for each method. We set the number of training epochs after some informal trials. In the informal trials, we monitor the improvement in sentence accuracy metric in the validation set. We stop the training when the validation sentence accuracy remains constant. As a result, 100 epochs are performed for BiRNN, Att-BiRNN, Att-BiRNN-Crf, Focus-BiRNN and Inter-Trans methods. Fifty epochs are needed for the Slot-Refine. We perform 10 epochs of training for the BERT-joint.

In Table 7, we provide the results for the Turkish ATIS data set using all seven joint models. We also present the results for the original ATIS in Table 6 in order to compare the results in English to Turkish. The experiments in Table 6 and Table 7 are run with the same parameters. In the first four columns of the tables, we provide the evaluation metrics for the validation set. The following four columns represent the results of the test set. When computing the metrics, the sentence accuracy is monitored for each epoch in the validation set and the epoch that gives the best validation sentence accuracy is saved as the best epoch. We report the validation and test Slot_F1, Intent_Acc, Intent_F1 and Sent_Acc for the best epoch. In order to obtain more reliable results, we repeat each experiment with a different random seed for 5 times. In the tables, the mean and standard deviation of the runs are presented.

For the English ATIS data set, we obtain the best performances with the cointeractive transformer network (Inter-Trans) and BERT-joint model. The performances of the methods are comparable for the test and validation sets except the Sent_Acc metric. In the Sent_Acc metric, the cointeractive transformer outperforms the BERT-joint model. For the Turkish ATIS data set, we generally obtain the best results with the BERT-

joint method. Only in the Sent_Acc metric in the test set, the cointeractive transformer yields slightly better performance. The better performance of BERT-joint might be attributed to the fine-tuning of all parameters of a pretrained BERT model to the target domain using task-specific training samples. The fine-tuning procedure might yield better embeddings for the tokens in the target domain, especially for the untranslated proper nouns in English such as location names Baltimore and Dallas in Table 2.2.

When we compare Table 6 and Table 7, we observe that there is still a performance gap between the results in English and Turkish. The same observation is also made in the previous works [22, 24]. This result can be partly attributed to the errors during the translation of the original English data set to Turkish. Despite manual corrections, there could still be some minor problems in the Turkish data set. On the other hand, the pretrained models in Turkish are usually trained using smaller text corpora compared to the English counterparts. We think that the performance gap is also due to the differences between the pretrained transformer/word embedding models in English and Turkish. In the future, more efforts could be made to develop more representative models for the Turkish language.

Table 6. Results for the original English ATIS data set.

	Validation Set				Test Set			
	Slot_F1	Intent_Acc	Intent_F1	Sent_Acc	Slot_F1	Intent_Acc	Intent_F1	Sent_Acc
BiRNN	95.86± 0.23	95.86± 0.67	67.07± 6.45	85.54± 0.33	93.44± 0.26	93.31± 0.81	64.95± 4.90	78.41± 0.45
Att-BiRNN	95.05± 0.16	95.98± 0.53	69.08± 7.79	83.57± 0.46	93.10± 0.28	95.37± 0.89	77.93± 4.74	81.15± 1.31
Att-BiRNN-Crf	96.01± 0.11	96.26± 0.32	74.59± 6.28	85.26± 0.32	94.29± 0.18	95.64± 0.89	78.96± 5.80	82.83± 1.06
Focus-BiRNN	94.98± 0.35	96.50± 0.27	70.10± 6.78	83.57± 0.60	93.24± 0.18	95.80± 0.27	77.89± 2.83	81.02± 0.56
Slot-Refine	97.42± 0.22	97.83± 0.40	84.67± 8.32	89.71± 0.61	95.05± 0.18	97.52± 0.19	84.32± 2.48	85.21± 0.16
Inter-Trans	**98.46± 0.22**	97.79± 0.25	**91.55± 2.69**	**94.01± 0.38**	**96.55± 0.09**	**97.98± 0.13**	91.28± 1.06	**90.86± 0.29**
BERT-joint	97.74± 0.13	**98.15± 0.15**	90.43± 2.82	91.60± 0.23	95.43± 0.10	**97.95± 0.30**	**91.61± 0.99**	87.68± 0.34

In order to further analyze the slot detection errors in the Turkish ATIS, we provide the reference and recognized slots for two example sentences in Table 8 and Table 9. In the tables, we use the BERT-joint model to obtain the recognition results. As observed in Table 8, the slot label 'B-a_date.d_num (beginning arrival date day number)' for the first word 'altı' is confused with the slot label 'B-d_date.day_num (beginning departure date day number)'. This error might be attributed to the fact that the language model may not be able to resolve the relatively long-term relation between the last word 'varır (arrives)' and the first word 'altı (six)' in the sentence. Additionally, 'B-toloc.p_name (beginning to location airport name)' and 'I-toloc.p_name (inside to location airport name)' labels for the words 'love field'e' are confused with 'B-toloc.c_name (beginning to location city name)' and 'I-toloc.c_name (inside to location city name)'. We think that this might be due to the fact that the untranslated proper noun in English (love field'e) might be inadequately presented in the training corpora of the pretrained word embedding/language model for Turkish. When we further analyze the result, we realize that the proper noun (love field's) also does not appear in the training set of the ATIS data set. Therefore, the pretrained BERT model parameters are not fine-tuned with this phrase. All the slots in

Table 7. Results for the Turkish ATIS data set.

	Validation Set				Test Set			
	Slot_F1	Intent_Acc	Intent_F1	Sent_Acc	Slot_F1	Intent_Acc	Intent_F1	Sent_Acc
BiRNN	86.71± 0.65	92.15± 0.87	60.68± 6.39	67.80± 0.52	83.86± 0.28	90.23± 0.85	55.47± 3.57	58.54± 0.55
Att-BiRNN	83.05± 0.42	93.84± 0.63	68.92± 5.05	60.28± 0.94	81.41± 0.47	91.94± 0.83	62.51± 2.95	56.79± 0.97
Att-BiRNN-Crf	86.72± 0.46	93.64± 0.41	69.58± 4.12	67.04± 0.66	84.85± 0.50	92.46± 0.76	63.76± 2.16	63.09± 0.88
Focus-BiRNN	85.57± 0.82	92.71± 0.34	55.30± 3.49	64.26± 1.50	84.01± 0.53	90.48± 1.07	53.92± 3.34	60.86± 1.17
Slot-Refine	91.22± 0.36	94.20± 0.51	64.82± 6.34	74.32± 0.78	88.06± 0.27	93.03± 0.41	69.48± 3.70	66.93± 0.65
Inter-Trans	93.05± 0.21	95.33± 0.34	83.08± 1.21	**80.52± 0.49**	90.20± 0.24	95.51± 0.41	82.36± 2.02	**77.24± 0.58**
BERT-joint	**93.80± 0.25**	**95.81± 0.19**	**83.15± 2.80**	80.20± 0.74	**91.56± 0.29**	**96.54± 0.18**	**87.44± 0.63**	76.90± 0.43

this utterance are correctly recognized in the English experiments. In the second example in Table 9, 'I-f_mod (inside flight mode)' label for the word 'gelen' is confused with the outside label 'O'. The phrase 'earliest arriving' which appears in the original English sentence is translated to Turkish as 'en erken gelen'. Although the first two slots are correctly recognized for the first two words (en erken), the model makes an error in the last word. When we further analyze the results, we realize that a similar error was made in the experiments for the English ATIS.

Table 8. An example sentence from the Turkish ATIS test set with the reference and recognized slot labels. The recognition output is obtained using the BERT-joint method.

Sent	altı	haziranda	love	field'e	hangi	uçaklar	varır
Ref	B-a_date.d_num	B-a_date.m_name	B-toloc.p_name	I-toloc.p_name	O	O	O
Rec	B-d_date.d_num	B-a_date.m_name	B-toloc.c_name	I-toloc.c_name	O	O	O

Table 9. An example sentence from the Turkish ATIS test set with the reference and recognized slot labels. The recognition output is obtained using the BERT-joint method.

Sent	boston'dan	washington	dc'ye	en	erken	gelen	uçuş	nedir
Ref	B-fromloc.c_name	B-toloc.c_name	B-toloc.s_code	B-f_mod	I-f_mod	I-f_mod	O	O
Rec	B-fromloc.c_name	B-toloc.c_name	B-toloc.s_code	B-f_mod	I-f_mod	O	O	O

4.3. Further experiments for Turkish

We run more experiments in order to gain more insight into the results for the Turkish ATIS data set. In Table 6 and Table 7, it is observed that the macro averaged intent F1 scores are much lower when compared to the intent detection accuracy. In macro average F1 computation, the label imbalance is not taken into account. The metric computed by calculating F1 for each label and taking their unweighted mean. Therefore, the large

performance gap between the intent macro F1 and accuracy might be attributed to low performance for the labels which have few samples in the test set. In order to validate this observation, we provide Table 10 in which F1 scores for individual intent labels are provided with the number of train and test samples for each label. As observed in the table, the 3 labels ('city', 'quantity', 'flight+airfare') with a F1 score of less than 70% have only 21 support samples in total 879 test utterances. When we further analyze the results in the low-performance labels, we observe that they are usually confused with the majority intent label 'flight'. For example, 'city' label with 6 support samples is confused with the 'flight' 3 times. Similarly, 'flight+airfare' with 12 samples is confused with the 'flight' 5 times.

Table 10. Detailed intent detection results for the Turkish ATIS data set. The results are obtained using the best-performing BERT-joint method.

Label	F1	# of train samples	# of test samples	Label	F1	# of train samples	# of test samples
flight	98.03	3389	627	city	28.57	33	6
airfare	88.40	404	48	flight+airfare	67.13	19	12
ground_service	98.87	231	35	distance	100	17	10
airline	99.48	141	38	airport	100	17	17
abbreviation	96.96	130	32	ground_fare	85.12	16	7
aircraft	84.21	71	9	flight_no	100	15	8
flight_time	100	45	1	capacity	100	15	21
quantity	54.54	41	3	meal	97.77	6	5

In the second set of experiments, we use the BERT as a feature extractor and freeze its parameters during fine-tuning. The token embeddings of the BERT model are fed into the intent detection and slot filling classifiers similar to the BERT-joint model. Different from the BERT-joint, the model is not fine-tuned end-to-end but only the classifier parameters are trained using the loss function in Equation (1). We obtain 70.77% Intent_Acc and 58.41% Slot_F1 when the parameters of the BERT are not adapted to the target domain. From these results, it is clear that the fine-tuning of the entire model is crucial in the BERT-joint method to obtain high performance. However, it should also be noted that fine-tuning all parameters of a relatively large BERT model results in a longer training duration.

We obtained 96.54% intent detection accuracy and 91.56% slot filling F1 for the Turkish ATIS data set. We compare the performance in our data set to the previously published version of ATIS in [20] in Table 11. As mentioned earlier, we could not find the intent labels in the previous version so the comparison is based on only slot filling performance. We use the BERTurkish as the pretrained transformer model as in Table 7 and report the performance using the same Slot_F1 and Sent_Acc metrics. Sent_Acc shows the percentage of the sentences for which all slot labels are correctly recognized. We ran the experiment 5 times in the previous version of ATIS with a different random seed and reported mean and standard deviation of the metrics in Table 11. As observed in the table, the performance in our version is significantly better than the performance in the previous version. This result might be attributed to the alignment problems between the words and slots in the previous version. Some of the alignment problems are also mentioned in [24]. In [24] and [20], 85,07% and 88.0% slot filling F1 are reported using the ATIS version in [20], respectively. In [20], the best intent detection F1 is 89.9%. From these results, it can be concluded that we report significantly improved intent detection and slot filling performances for the Turkish language compared to the previous studies.

Table 11. Comparison of the slot filling performance on our ATIS data set to the previous version in [20].

	Slot_F1	Sent_Acc
Turkish ATIS (ours)	91.56±0.29	76.90±0.43
Previous Turkish ATIS ([20])	84.46±0.14	63.03±0.54

5. Conclusions

In this paper, we translate the English ATIS data set to Turkish and present our intent detection and slot filling performances for the Turkish language using seven different joint models. In the joint modeling, the intent detection and slot filling are performed simultaneously to utilize the synergy between the tasks. To the best of our knowledge, this is the first study that applies joint modeling for Turkish. In the experiments, we obtained the best performance in Turkish using the BERT-joint method with 96.54% intent detection accuracy and 91.56% slot filling F1. The performance in Turkish is still slightly lower than the English counterparts. This result might be attributed to the following reasons; i) translation/alignment errors in the automatic translation procedure, ii) untranslated proper nouns in English which are usually less presented in the training corpora of the Turkish pretrained models, iii) agglutinative nature of the Turkish language which results in larger vocabulary size, iv) more robust pretrained models in English since they are trained using larger training data. On the other hand, our results are significantly better than the previously published results for similar tasks and data sets in Turkish. We shared our Turkish data set in our github repository for further experimentation. We think that our results together with the publicly available data set will serve as a strong baseline for intent detection and slot filling studies in the Turkish language. We hope that this will accelerate further research on new methods for both tasks. Our work might potentially have an impact on researchers developing practical goal-oriented dialogue systems for Turkish.

The ATIS data set is severely imbalanced although it is extensively used as a benchmark test set in the literature. This constitutes the major shortcoming of the data set. In the future, more balanced data sets containing diverse intent/slot classes might be collected and made publicly available for the Turkish language.

In this paper, the machine learning models are trained using a supervised data set in which the slot words and intent classes are carefully labeled by human annotators. The human annotation process is time-consuming and expensive. It is usually difficult to find annotated data sets in a commercial setting. For example a client company for a chat-bot application might provide the raw conversations between its user and human agent in order to automate its dialogue system. In this scenario, the intent classes and slot words should be directly extracted from the raw text. In the future, we plan to tackle the problem of extracting valuable information from raw text using semiautomatic methods.

Acknowledgment

O.B. did the experiments, interpreted the results, and wrote the paper.

References

[1] Bayatlı S,Kurnaz S,Ali A, Washington JN, Tyers FM. Unsupervised weighting of transfer rules in rule-based machine translation using maximum-entropy approach. Journal of Information Science and Engineering 2020; 36 (2).

[2] Vukovic R, Heck M, Ruppik BM, Van-Niekerk C, Zibrowius M et al. Dialogue term extraction using transfer learning and topological data analysis. In: The 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2022), Heriot-Watt University, Edinburgh, UK; 2022. pp. 564-581.

[3] Casanueva I, Budzianowski P, Su PH, Mrksic N, Wen TH et al. A benchmarking environment for reinforcement learning based task oriented dialogue management. arXiv preprint arXiv:1711.11023 2017.

[4] Wu D, Ding L, Lu F, Xie J. SlotRefine: A fast non-autoregressive model for joint intent detection and slot filling. arXiv preprint arXiv:2010.02693 2020.

[5] Weld H, Huang X, Long S, Poon J, Han SC. A survey of joint intent detection and slot filling models in natural language understanding. ACM Computing Surveys 2002;55 (8):1-38

[6] Hakkani-Tur D, Tur G, Celikyilmaz A, Yun-Nung C, Gao J et al. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In: Interspeech 2016, pp. 715-719.

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

[8] Zhu S, Yu K. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017), New Orleans, LA, USA; 2017, pp. 5675-5679.

[9] Chen Q, Zhuo Z, Wang W. Bert for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909, 2019.

[10] Qin L, Liu T, Che W, Kang B, Zhao S et al. A co-interactive transformer for joint slot filling and intent detection. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021), Toronto, Ontario, Canada; 2021, pp. 8193-8197.

[11] Hemphill CT, Godfrey JJ, Doddington GR. The ATIS spoken language systems pilot corpus. In Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania; 1990.

[12] Coucke A, Saade A, Ball A, Bluche T, Caulier A et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv:1805.10190 2018.

[13] Zhang C, Li Y, Du N, Fan W, Yu P. Joint slot filling and intent detection via capsule neural networks. In: ACM Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5259–5267, Florence, Italy.

[14] Zhang L, Ma D, Zhang X, Yan X, Wang H. Graph LSTM with context-gated mechanism for spoken language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 9539-9546), 2020.

[15] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 2018.

[16] Liu Y, Ott M, Goyal N, Du J, Joshi M el al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 2019.

[17] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD et al. Language models are few-shot learners. Advances in neural information processing systems 2020; 33: 1877-1901.

[18] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R et al. XLnet: Generalized autoregressive pretraining for language understanding. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS 2019), Vancouver, Canada; 2019, 517, pp. 5753–5763.

[19] Dundar EB, Kilic OF, Cekic T, Manav Y, Deniz O. Large scale intent detection in Turkish short sentences with contextual Word embeddings. In: 12th International Joint Conference on Knowledge Discovery Knowledge Engineering and Knowledge Management (KDIR) 2020; pp. 187-192.

[20] Sahinuc F, Yucesoy V, Koc A. Intent classification and slot filling for Turkish dialogue systems. In: IEEE 28th Signal Processing and Communications Applications Conference (SIU); 2020.

[21] Yilmaz EH, Toraman C. Intent classification based on deep learning language model in Turkish dialog systems. In: IEEE 29th Signal Processing and Communications Applications Conference (SIU); 2021.

[22] Buyuk O, Erden M, Arslan LM. Leveraging the information in in-domain datasets for transformer-based intent detection. In: IEEE Innovations in Intelligent Systems and Applications Conference (ASYU); 2021.

[23] Buyuk O, Erden M, Arslan LM. Mitigating data imbalance problem in transformer-based intent detection. Avrupa Bilim ve Teknoloji Dergisi 2021; 32: 445-450.

[24] Ozcelik O, Yilmaz EH, Şahinuc F, Toraman C. Slot filling for voice assistants. In: IEEE 30th Signal Processing and Communications Applications Conference (SIU) 2022.

[25] Lin CY. Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. 2004. pp. 74-81.

[26] P Xu, Sarikaya R. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) 2013; 78–83.

[27] Raymond C, Riccardi G. Generative and discriminative algorithms for spoken language understanding. In: 8th Annual Conference of the International Speech Communication Association (Interspeech); 2007.

[28] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014;1532-1543.