Joint Multiple Intent Detection and Slot Labeling for Goal-Oriented Dialog

Rashmi Gangadharaiah
AWS AI, Amazon
rgangad@amazon.com

Balakrishnan Narayanaswamy
AWS AI, Amazon
muralibn@amazon.com

Abstract

Neural network models have recently gained traction for sentence-level intent classification and token-based slot-label identification. In many real-world scenarios, users have multiple intents in the same utterance, and a token-level slot label can belong to more than one intent. We investigate an attention-based neural network model that performs multi-label classification for identifying multiple intents and produces labels for both intents and slot-labels at the token-level. We show state-of-the-art performance for both intent detection and slot-label identification by comparing against strong, recently proposed models. Our model provides a small but statistically significant improvement of 0.2% on the predominantly single-intent ATIS public data set, and 55% intent accuracy improvement on an internal multi-intent dataset.

1 Introduction

In dialog systems, the natural language understanding component (NLU) is responsible for identifying the user's request and creating a semantic frame that succinctly summarizes the user's needs. These semantic frames are typically constructed using *intents* and *slot-labels* (Tür et al., 2010). As the names imply, an intent captures the intention of the user and slot-labels capture any additional information or constraints the user provides. These constraints must be satisfied in order to fulfill the user's request. The example below shows a user's request, *"how is the weather in Dallas ?"*. We need to identify the intent (*"GetWeatherInfo"*) as well as the values for the slot-labels (SL), here, *"City"* (value=*"Dallas"*). It is crucial that intents and slot-labels are identified with high accuracy as an error made by the NLU component propagates through downstream components such as the dialog state tracker, the dialog policy and the natural language generator components, leading to a substantial degradation of the performance of the entire dialog system.

NLU Semantic Frame

how	is	the	weather	in	Dallas	?
↕	↕	↕	↕	↕	↕	↕
SL: O	O	O	O	O	City	O

Intent: GetWeatherInfo

Intent detection has been modeled as a sentence classification task where an intent (y^I) is assigned to the user's utterance. Slot labeling is typically modeled as a sequential labeling problem, where a user's sentence, $x_1, x_2, ...x_N$, is labeled with $y_1^S, y_2^S, ..y_N^S$, and y_i^S is the slot label assigned to the token at position i (x_i). In the example above, the sequence of slot labels would be, *"O O O O O City O"*, where, *"O"* stands for *"Other"*.

Sequential models such as Maximum Entropy Markov models (Toutanova and Manning, 2000; McCallum et al., 2000; Berger et al., 1996) and Conditional Random Fields, CRFs (Lafferty et al., 2001; Jeong and Geunbae Lee, 2008) are popular approaches for slot-labeling while intent prediction is often performed using standard classification approaches such as Support Vector Machines (Cortes and Vapnik, 1995) or logistic regression (Bishop, 2006). More recently, neural network-based models (Mesnil et al., 2015; Kurata et al., 2016; Goo et al., 2018; Liu and Lane, 2016) have been shown to significantly outperform previous approaches. These models are also appealing since a single model is trained end-to-end to perform both intent detection and slot label identification. Jointly modeling intent and slot label identification (Liu and Lane, 2016; Goo et al., 2018) has been shown to significantly outperform other neural network-based approaches. This is intuitive since slot labels could depend on the intent.

Proceedings of NAACL-HLT 2019, pages 564–569
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

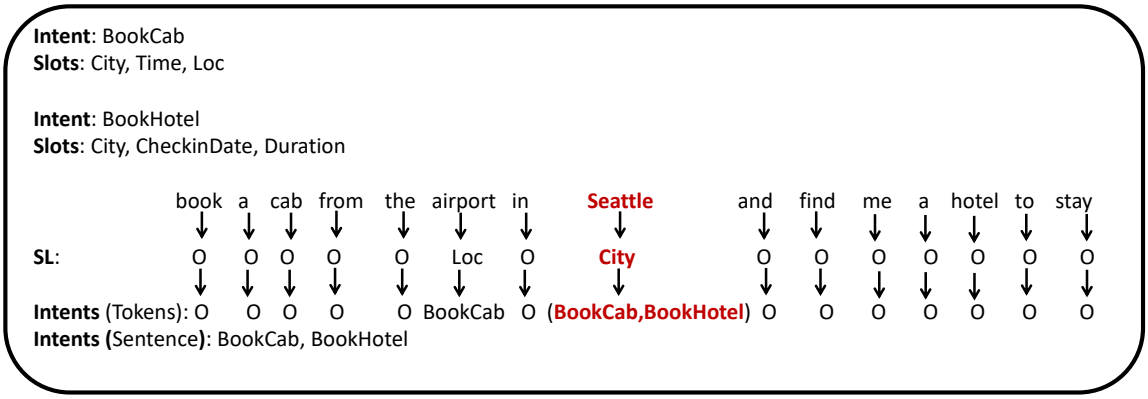

Figure 1: An example showing slot values belonging to multiple intents. Here, *Seattle* belongs to two of the intents in the user's utterance, *BookHotel* and *BookCab*.

Most neural network-based approaches (Mesnil et al., 2015; Kurata et al., 2016; Goo et al., 2018; Liu and Lane, 2016), with the exception of (Xu and Sarikaya, 2013a), predict a single intent for a user's utterance. In real-world scenarios, users indicate multiple intents in the same utterance. For example, a user's utterance such as, *"show me flights from Dallas to New York and the cost"*, clearly has two intents, one for obtaining the price of the flights (*"GetFlightCost"*) and another for the flight information. It is critical to understand and model such scenarios to allow more natural interaction with users. In this paper, we treat the intent detection task as a multi-label classification problem and suggest various neural network models to obtain multiple intents.

Our work is related to Xu et al.,(2013b) and Kim et al.,(2017), where multiple intents are assigned to a user's utterance. Xu et al., (2013b) use log-linear models to achieve this, while we use neural network models. Both Xu et al., (2013b) and Kim et al., (2017) only consider intents and do not handle slot labels. In this paper, we jointly perform multi-label intent classification and slot-label identification.

In contrast with all prior work, we investigate and study the problem of assigning slot labels (or constraints) provided by a user to multiple intents. Consider the example in Figure 1 with two intents in the same domain, *"BookCab"* and *"BookHotel"*. Suppose *"BookCab"* has three possible slot labels, *"City"*, *"Time"* and pick up location (*"Loc"*), and suppose that *"BookHotel"* has slot labels *"City"*, *"CheckinDate"*, and *"Duration"*. Consider a user's utterance, *"book a cab from the airport in Seattle and find me a hotel to stay"*. Here, the user wants to book a cab (*"BookCab"* intent) as well book a hotel (*"BookHotel"*). The slot label *"Seattle"* should be assigned to both intents to accurately capture the user's request. Hence, we study a model that predicts multiple intents both at the token level as well as at the sentence-level.

We model token-level multi-intent classification using Long Short Term Memory (LSTMs) units to capture dependencies that may exist between intents. For example, a user who wants to book a cab is also likely to make a request for a hotel in the same utterance but probably not order food i.e., intents such as *"BookCab"* and *"BookHotel"* are more likely to occur together when compared to *"BookCab"* and *"OrderFood"*. To summarize, the contributions of this paper are:

- We investigate approaches to the problem of multi-intent classification. We perform joint multi-intent classification both at sentence-level and at token-level. We see that,
  - the token-level multi-intents help assign user constraints to the intents.
  - sentence-level multi-intent classification captures dependencies between intents.
- We compare the performance of the approach with recently proposed state-of-the-art approaches and show significant improvement.

The paper is organized as follows. Section 2 describes the proposed approach. Section 3 describes the experimental setup, including, data sets and metrics used to evaluate the approaches followed by the results in Section 3.2. Finally, we conclude and suggest possible future directions and extensions in Section 4.

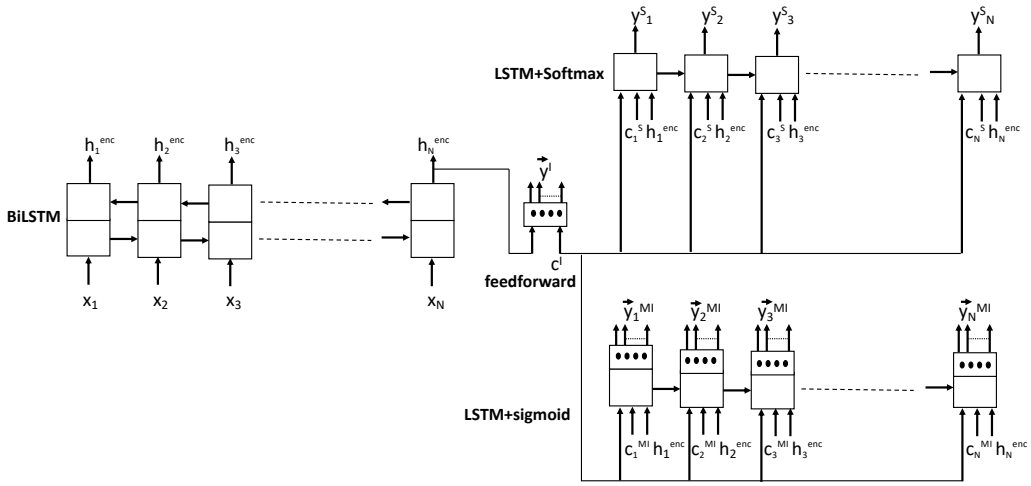

Figure 2: Proposed Approach. A bidirectional LSTM is used for the encoder layer. Multiple intents are predicted both at the sentence level (y^I) and at the token level (y^{MI}). y^I uses a feedforward network. Slot labels (y^S) and token level intent prediction (y^{MI}) both use LSTM layers, which have skip connections to the encoder states.

2 Proposed Approaches

LSTM-based RNN models have become popular for sequential labeling, especially in natural language processing tasks, due to their ability to model long-term dependencies. We extend encoder-decoder architectures from Liu et al., (2016) and Gangadharaiah et al., (2018), which showed superior performance when compared to Convolutional neural network based CRFs (Xu and Sarikaya, 2013a) and other RNN-based architectures (Mesnil et al., 2015; Kurata et al., 2016) for intent detect and slot label identification.

We use a bidirectional LSTM encoder to encode the input word sequence. The encoder hidden state, h_i^{enc}, at each word position is a concatenation of the forward state (fh_i) and backward state (bh_i), $h_i^{enc} = [fh_i, bh_i]$.

For intent detection at the sentence-level, a context vector c^I is computed using the encoder's final hidden state. The vectors, c^I and the final encoder's hidden state vector are sent to a dense layer of sigmoid units to predict the probabilities for every intent. This produces multiple intents ($\vec{y}^I$) as opposed to previous approaches that produce a single intent.

For slot labeling, the decoder also uses LSTMs. At each decoding step i, the decoder state ($h_i^{S,dec}$) is a function of the previous decoder state ($h_{i-1}^{S,dec}$), the previously emitted label (y_{i-1}^S), the encoder's state ($h_i^{S,enc}$), the context vectors, (c_i^S) and c^I, as shown in Figure 2. The context vector c_i^S is a weighted combination of the encoder's states ($h_1^{enc}, h_2^{enc}, ...h_N^{enc}$) with weights, $\alpha_{i,j}^S$, as shown in eqn. 1. g is a feed forward network.

$$c_i^S = \sum_{j=1}^{N} \alpha_{i,j}^S h_j^{enc} \quad (1)$$

$$\alpha_{i,j}^S = \frac{exp(e_{i,j})}{\sum_{k=1}^{N} exp(e_{i,k})}$$

$$e_{i,k} = g(h_{i-1}^{S,dec}, h_k^{enc})$$

The output of the LSTM layer is then sent to a softmax layer to predict the slot labels. We also experimented with a CRF layer as the decoder. In our preliminary experiments, the LSTM decoder was faster to train and also showed better performance when compared to the CRF layer and hence we use LSTMs in the experiments below. We also apply a slot-gated mechanism similar to Goo et al., (2018). The idea is to leverage the intent's context vector for modeling slot-intent relationships, thereby improving the performance of slot labeling. The slot gate is computed as a function of both the slot context vector (c_i^S) and the intent context vector (c^I), where, v and W are both trainable. In Goo et al., (2018), a similar model showed at-par or better performance over Liu et al. (2016) and Tür et. al. (2016). The slot gate gS is defined as,

$$gS = \sum v \cdot tanh(c_i^S + W \cdot c^I) \quad (2)$$

where, gS is used to weight h_i^{enc} and c_i^S to obtain y_i^S, i.e., $h_i^{enc} + c_i^S \cdot gS$ is sent to the feed forward network to compute y_i^S.

Since a slot label can belong to multiple intents, we also perform multi-label intent detection at the token level. We again use an LSTM decoder, where each decoder state, $h_i^{MI,dec}$, is a function of c^I, previous decoder state ($h_{i-1}^{MI,dec}$), the encoder's state (h_i^{enc}) and the context vector (c_i^{MI}), as shown in Figure 2. c_i^{MI} is computed in the same manner as c_i^S. The output of the decoder is sent to a dense layer with sigmoid units. Thus at each word position, we predict multiple intents.

3 Experiments

In all our experiments, we set the hidden vectors to a dimension of 64 and use the adam optimizer with an early stopping strategy. We use a drop-out rate of 0.5 for regularization and the maximum norm for gradient clipping is set to 5. The results are obtained by averaging the performance of the models over 10 runs. To do a fair comparison against existing models, we do not pre-train our word embeddings (Devlin et al., 2018; Pennington et al., 2014; Mikolov et al., 2013), instead we use an embedding layer in the model which is trained along with the rest of the model's parameters.

As done in the NLU community, we report F1 scores for slot labeling. We use F1 scores for intent detection at the token-level and accuracy for sentence-level intent detection.

3.1 Datasets

We use two widely used public datasets, ATIS (Airline Travel Information System) (Tür et al., 2010) and SNIPS [1]. The ATIS dataset contains audio recordings of people requesting flight reservations, with 21 intent types and 120 slot labels. There are 4,478 utterances in the training set, 893 in the test set and 500 utterances in the development set. The SNIPS data was collected from the SNIPS personal voice assistant, with 7 intent types and 72 slot labels. The training set contains 13,084 utterances, the test set contains 700 utterances and the development set also contains 700 utterances. The ATIS dataset contains utterances with multi intents, while the SNIPS is only single intent. In order to demonstrate that our approach does not degrade performance on single intent datasets, we also perform evaluations on the SNIPS dataset.

We also test the performance of the models on an internal dataset. In this dataset, about 52% of examples are multi-intent compared to ATIS which has ≈2% of test examples with multi-intents. The average number of intents per utterance in the internal dataset is 1.6.

[1] https://github.com/snipsco/nlu-benchmark/tree/master/2017-06-custom-intent-engines

3.2 Results

We compare our approach against two of the state of the art approaches that have shown the best performance in previous work. We will use **Model 1** to refer to the model proposed by Liu et al., (2016). **Model 2** refers to the more recent model proposed by Goo et al., (2018). Table **??** shows results obtained by the model investigated in this paper when compared with Model 1 and Model 2.

As mentioned earlier, both Models 1 and 2 only handle single intents per user utterance. For these two models, we insert a # between the multiple intents and treat it as one single intent, i.e., when an example such as, *"please give me a list of all the flights between dallas and baltimore and their cost"*, contains multiple intents, *"atis_flight"* and *"atis_airfare"*, we use *"atis_flight#atis_airfare"* instead. When evaluating the baselines, the ordering of intents does not matter, and so we replace the # with spaces once we have the predictions.

To allow comparison across approaches, both ATIS and SNIPS were modified to include token-level intents as follows. For utterances that had only a single intent, we assigned this intent to all tokens that had a slot label (i.e., to slot labels that do not correspond to O). For utterances that had more than one intent, we assigned all intents to all tokens that had slot labels. After this process, if an utterance had two intents, $intent_1$ and $intent_2$, and if a token i had a slot label, the token would end up with targets of the form,

$$(slot^i, intent_1^i, intent_2^i)$$

The proposed model shows a statistically significant improvement in sentence-level intent prediction (S-level) on ATIS when compared to the two baselines. Any improvement in slot labeling is unclear, since this could be attributed to the architecture changes which involved additional penalty terms on the intent (since we use both token-level and sentence-level intent loss). We also notice that the performance on SNIPS (a single intent dataset) does not degrade. We see a larger performance boost in both token-level (T-level) and sentence-level (S-level) intent detection on the internal dataset due to the large percentage of examples with multi-intents. Wilcoxon signed-rank test

Model	ATIS			SNIPS			Internal Dataset		
	Slot (F1)	Intent (Acc) S-level	Intent (F1) T-level	Slot (F1)	Intent (Acc) S-level	Intent (F1) T-level	Slot F1	Intent (Acc) S-level	Intent (F1) T-level
Model 1	90.16	93.84	N/A	87.24	97.14	N/A	89.28	57.27	N/A
Model 2	93.37	95.18	N/A	**88.23**	96.85	N/A	89.64	57.47	N/A
Proposed approach	**94.22**	**95.39**	95.82	88.03	**97.23**	97.89	**90.94**	**89.41**	**94.54**

Table 1: Performance of the model against Model 1 and Model 2. We report F1 scores for slot labeling. For intent detection, we use F1 scores for intent detection at the token-level (T-level) and accuracies (acc) for sentence-level (S-level) intent detection. N/A: as Models 1 and 2 perform single intent detection only at S-level.

(Wilcoxon, 1945) was used to find statistical significance.

4 Conclusion and Future Work

The paper investigated an approach for multi-intent classification. We perform multi-intent classification both at sentence-level and at token-level. The token-level multi-label classification helped assign common constraints (or slot labels) to multiple intents, improving accuracy. The sentence-level multi-intent classification captured dependencies between intents. We compared the performance of our approach with previously proposed state-of-the-art approaches for single intent classification and showed significant improvements in performance on all the datasets.

As future work, we would like to explore other architectures to directly model dependencies between slot labels and intents. This is useful since only a subset of slot labels occur with certain intents. We will also test the proposed approaches against real-world scenarios to understand their generality across various domains.

References

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Rashmi Gangadharaiah, Balakrishnan Narayanaswamy, and Charles Elkan. 2018. What we need to learn if we want to do and not just talk. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 3 (Industry Papers)*, pages 25–32.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757. Association for Computational Linguistics.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*, INTERSPEECH '16. ISCA.

Minwoo Jeong and G. Geunbae Lee. 2008. Triangular-Chain Conditional Random Fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1287–1302.

Byeongchang Kim, Seonghan Ryu, and Gary Geunbae Lee. 2017. Two-stage Multi-intent Detection for Spoken Language Understanding. *Multimedia Tools Appl.*, 76(9):11377–11390.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2077–2083. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. *CoRR*, abs/1609.01454.

Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 591–598, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding. *Trans. Audio, Speech and Lang. Proc.*, 23(3):530–539.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-speech Tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2010. What is left to be understood in ATIS? In *SLT*, pages 19–24. IEEE.

Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83.

Puyang Xu and Ruhi Sarikaya. 2013a. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *ASRU*, pages 78–83. IEEE.

Puyang Xu and Ruhi Sarikaya. 2013b. Exploiting Shared Information for Multi-intent Natural Language Sentence Classification. In *Interspeech*.