Slot Tagging for Task Oriented Spoken Language Understanding in Human-to-human Conversation Scenarios

Kunho Kim[†], **Rahul Jha**[†], **Kyle Williams**[†], **Alex Marin**[†], **Imed Zitouni**[‡*]
[†]Microsoft Corporation, Redmond, WA, USA
[‡]Google, Mountain View, CA, USA
{kuki, rajh, kywillia, alemari}@microsoft.com, izitouni@google.com

Abstract

Task oriented language understanding (LU) in human-to-machine (H2M) conversations has been extensively studied for personal digital assistants. In this work, we extend the task oriented LU problem to human-to-human (H2H) conversations, focusing on the slot tagging task. Recent advances on LU in H2M conversations have shown accuracy improvements by adding encoded knowledge from different sources. Inspired by this, we explore several variants of a bidirectional LSTM architecture that relies on different knowledge sources, such as Web data, search engine click logs, expert feedback from H2M models, as well as previous utterances in the conversation. We also propose ensemble techniques that aggregate these different knowledge sources into a single model. Experimental evaluation on a four-turn Twitter dataset in the restaurant and music domains shows improvements in the slot tagging F1-score of up to 6.09% compared to existing approaches.

1 Introduction

Spoken Language Understanding (SLU) is the first component in digital assistants geared towards task completion, such as Amazon Alexa or Microsoft Cortana. The input to an SLU component is a natural language utterance from the user and its output is a structured representation that can be used by the downstream dialog components to select the next action. The structured representation used by most standard dialog agents is a semantic frame consisting of domains, intents and slots (Tur and De Mori, 2011). For example, the structured representation of "Find me a cheap Italian restaurant" is the domain **Restaurant**, the intent *find_place*, and slots $[cheap]_{price\_range}$, $[Italian]_{cuisine}$, $[restaurant]_{place\_type}$. Different sub-tasks within SLU have been extensively studied for human-to-machine (H2M) task completion scenarios (Sarikaya et al., 2016).

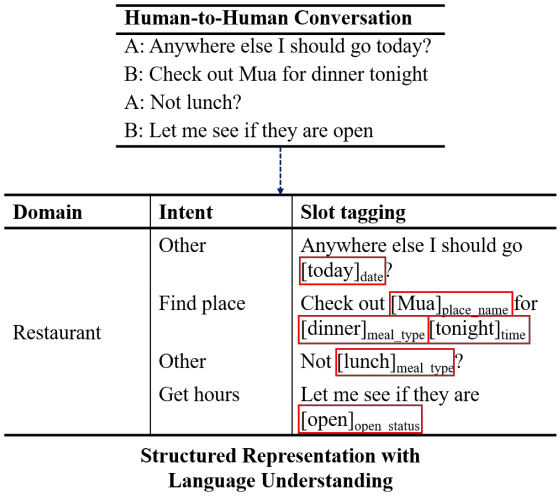

Figure 1: Example of language understanding for task completion on a H2H conversation. In this work, our goal is to identify useful slots (marked with red rectangles).

We extend the task oriented SLU problem to human-to-human (H2H) conversations. A digital assistant can listen to the conversation between two or more humans and provide relevant information or suggest actions based on the structured representation captured with SLU. Figure 1 shows an example of capturing intents and slots expressed implicitly during a conversation between two humans. The digital assistant can show general information about the restaurant Mua, and provide the opening hours based on the captured structured representation. These types of H2H task completion scenarios may allow digital assistants to suggest useful information to users in advance without them needing to explicitly ask questions.

In this paper, we investigate SLU oriented to-

*Work done while the author was at Microsoft Corporation

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 757–767
Hong Kong, China, November 3-4, 2019.

wards task completion for H2H scenarios with a specific focus on solving the **slot tagging** task. Some early conceptual ideas on this problem were presented in DARPA projects on developing cognitive assistants, such as CALO[1] and RADAR[2]. This work can be seen as an effort to formalize the problem and propose a practical framework.

SLU for task completion in H2H conversations is a challenging problem. Firstly, since the problem has not been studied before, there are no existing datasets to use. Therefore, we built a multi-turn dataset for two H2H domains that we found to be prevalent in Twitter conversations: **Music** and **Restaurants**. The dataset is described in more detail in Section 4. Secondly, the task is harder than H2M conversations in several aspects. It is hard to identify the semantics of noisy H2H conversation text with slang and abbreviations, and such conversations have no explicit commands toward the digital assistants requiring the assistant to indirectly infer users intent.

In this work, we introduce a modular architecture with a core bi-directional LSTM network, and additional network components that utilize knowledge from multiple sources including: sentence embeddings to encode semantics and intents of noisy texts with web-data and click logs, H2M based expert feedback, and contextual models relying on previous turns in the conversation. The idea of adding components is inspired from some recent advances in H2M SLU that use additional encoded information (Chen et al., 2016; Su et al., 2018; Kim et al., 2017; Jha et al., 2018). However, these work only considered adding a component from a single knowledge resource. Furthermore, since these additional components bring in information from different perspectives, we also experimented with deep learning based ensemble methods. Our best ensemble method outperforms existing methods by 6.09% for the *Music* domain and 2.62% for the *Restaurant* domain.

In summary, this paper makes the following contributions:

- A practical framework on slot tagging for task oriented SLU on H2H conversations using bidirectional LSTM architecture.
- Extension of the LSTM architecture utilizing knowledge from external sources (e.g. Web data, click logs, H2M expert feedback, and pervious sentences) with deep learning based ensemble methods
- Newly developed dataset for evaluating task oriented LU on H2H conversations

We begin by describing our methods for H2H slot tagging in Section 3. We then describe the data used in our experiments in Section 4 and discuss results in Section 5. This is followed by a review of the related work and conclusion.

2 Related Work

LU involves domain classification, intent classification, and slot tagging (Tur and De Mori, 2011; Sarikaya et al., 2016). Recently various deep neural network (DNN) models have been studied to solve each of these task, such as deep belief network (Sarikaya et al., 2011), deep convex network (Deng et al., 2012), RNN and LSTM (Ravuri and Stolcke, 2015; Mesnil et al., 2015).

Recent advances in LU use additional encoded information to improve DNN based models. There have been some attempts to use data or models from existing domains. One direction is to do transfer learning. Kim et al. (2017) and Jha et al. (2018) utilized previously trained models relevant to the target domain as expert models. They use the output of expert models as additional input to add relevant knowledge while training for the target domain. Goyal et al. (2018) reused low-level features from previously trained models and only retrained high level layers to adapt to a new domain.

There have also been some attempts to use contextual information. Xu and Sarikaya (2014) used past predictions of domains and intents in the previous turn for predicting current utterance. Chen et al. (2016) expanded upon this work by using a set of past utterances utilizing a memory network (Sukhbaatar et al., 2015) with an attention model. Subsequent works attempted to use the order and time information. Bapna et al. (2017) additionally used the chronological order of previous sentences, and Su et al. (2018) used time decaying functions to add temporal information.

Our work trains a sentence embedding that encodes the semantics and intents. DSSM and its variants (Huang et al., 2013; Shen et al., 2014; Palangi et al., 2016) are used for training sentence embedding, which were originally used for finding

[1] https://en.wikipedia.org/wiki/CALO
[2] https://www.cmu.edu/cmnews/extra/030718_darpa.html

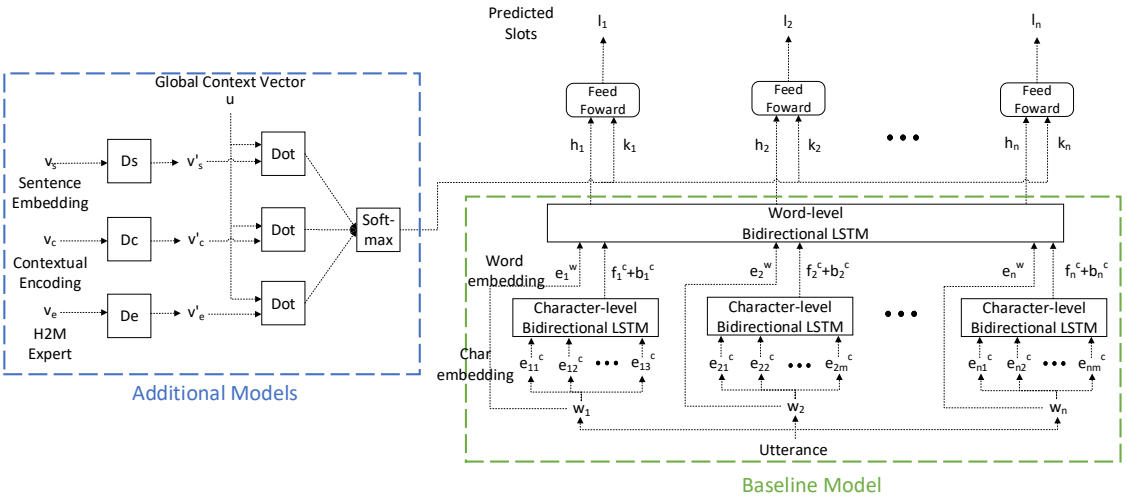

Figure 2: Overview of our slot tagging architecture. Our architecture consists with the core network (Section 3.1) and additional network components utilizing knowledge from multiple sources (Each discussed in Section 3.2.1, 3.2.2, 3.2.3). A network ensembling approach is applied on additional components (Section 3.3), figure shows with the attention mechanism.

relevance between the query and retrieved documents in a search engine. Also there have been attempts to use sentence embeddings similar to our data (Twitter). Dhingra et al. (2016) trained an embedding for predicting hash tags of a tweet using RNNs, Vosoughi et al. (2016) used an encoder-decoder model for sentiment classification.

All of the previous methods have studied LU components for task completion in H2M conversations. On the other hand, prior work on LU on H2H conversations has focused on dialog state detection and tracking for spoken dialog systems. Shi et al. (2017) used CNN model, and later extended multiple channel model for a cross-language scenario (Shi et al., 2016). Jang et al. (2018) used attention mechanism to focus on words with meaningful context, and Su et al. (2018) used a time decay model to incorporate temporal information.

3 Methods

Figure 2 shows the overview of our slot tagging architecture. Our modular architecture is a core LSTM-based network and additional network components that encode knowledge from multiple sources. Slot prediction is done with the final feed forward layer, whose input is the composition of the output of the core network and the additional components. We first describe our core network and then the additional network components, followed by our network ensembling approach.

3.1 Core Network

Our core network is a bidirectional model similar to Lample et al. (2016). The first character-level bidirectional LSTM layer extracts the encoding from a sequence of characters from each word. Each character c is represented with a character embedding $e^c \in \mathbb{R}^{25}$, and the sequence of the embedding is used as the input. The layer outputs

$$f^c = LSTM_{forward}(e^c) \quad (1)$$
$$b^c = LSTM_{backward}(e^c) \quad (2)$$

for each character, where $f^c, b^c \in \mathbb{R}^{25}$.

The second word-level bidirectional LSTM layer extracts the encoding from a sequence of words for each sentence. For each word w_i, the input of the layer is $g_i = f_i^c \oplus b_i^c \oplus e_i^w$ where f_i^c and b_i^c is the output of previous layer, $e_i^w \in \mathbb{R}^{100}$ is the word embedding vector, and $\oplus$ is a concatenation operator of vectors. We use pre-trained GloVe with 2B tweets [3] (Pennington et al., 2014) for the word embedding. The forward and backward word-level LSTM's produce

$$f_i^w = LSTM_{forward}(g_i) \quad (3)$$
$$b_i^w = LSTM_{backward}(g_i) \quad (4)$$

where $f_i^w, b_i^w \in \mathbb{R}^{100}$. Finally, slot l_i is predicted with the last feed forward layer with the input $h_i = f_i^w \oplus b_i^w$.

[3] Downloaded from https://nlp.stanford.edu/projects/glove/

Our model is trained using stochastic gradient descent with Adam optimizer (Kingma and Ba, 2015), with the mini batch size 64 and the learning rate 0.7×10^{-3}. We also apply dropout (Srivastava et al., 2014) on embeddings and other layers to avoid overfitting. The learning rate and dropout ratio were optimized using random search (Bergstra and Bengio, 2012). The core network can be used alone for slot tagging, however we discuss our additional network components in the following sections for improving our architecture.

3.2 Additional Network Components

In this section, we discuss additional network components that encode knowledge from different sources. Encoded vectors are used as additional input to the feed forward layer as shown in Figure 2.

3.2.1 Sentence Embedding for H2H Conversations

Texts from H2H conversations are noisy and contain slang and abbreviations, which can make identifying their semantics challengins. In addition, it can be challenging to infer their intents since there are no explicit commands toward the digital assistants. The upper part of Figure 3 shows part of a conversation from Twitter. The sentence lacks the semantics needed to fully understand "club and country". However, if we follow the URL in the original text, we can get additional information to assist with the understanding. For instance, the figure shows texts found from two sources, **1) web page title of the URL in the tweet** and **2) web search engine queries that lead to the URL in the tweet**. We use web search queries and click logs from a major commercial Web search engines to find queries that lead to clicks on the URL. Using this information, we can infer from the Web page title that the "club and country" referred to in the tweet are Atletico Madrid and Nigeria. Furthermore, the search queries from the search engine logs indicates possible user intents.

In our approach, we encode knowledge found from these two sources based on the URL. In our dataset, we were able to gather 2.35M pairs of tweet text with URL and web search engine queries that lead to the same URL, and 420K pairs of tweet text and web page titles of the URL. We then use this information to train a sentence embedding model that can be used to encode the semantics and implicit intents of each H2H conversation sentence. Our approach is to train a model that projects texts from H2H conversation and texts from each knowledge sources into a same embedding space, keeping the corresponding text pairs close to each other with other non-relevant texts being apart, as shown in Figure 3. The learned embedding model F then can be used to represent any texts from H2H sentences with a vector with semantically similar texts (or similar intents) being projected close to each other in the embedding space. Embeddings are used as additional component of our modular architecture, so that the semantic and intent information can be utilized in our slot tagging model.

We use the deep structured semantic model (DSSM) architecture (Huang et al., 2013) to train the sentence embedding encoder. DSSM uses letter-trigram word hashing, so it is capable of partially matching noisy spoken words so that we can get more robust sentence embeddings for H2H conversations. Let S be the set of sentences from the H2H conversations that have the URL. For each sentence $s \in S$, we find corresponding texts (**web page title of the URL, web search engine queries to the URL**) T_s^+ and randomly choose non-related texts T_s^- from corresponding texts of other sentences (in other words, from different URLs). Like the original DSSM model, each sentence s, $t_s^+ \in T_s^+$, and $t_s^- \in T_s^-$ are initially encoded with letter-trigram word hashing vector x, and used as the input of two consecutive dense layers,

$$x' = W_1 x + b_1 \tag{5}$$

$$y = W_2 x' + b_2 \tag{6}$$

where $x' \in \mathbb{R}^{1000}$ and $y \in \mathbb{R}^{300}$. We train the model to favor choosing $t_s^+ \in T_s^+$ over $t_s^- \in T_s^-$ for each s. So the loss function is defined as minimizing the likelihood,

$$\text{loss} = -\log \prod_{s, T_s^+} P(T_s^+ | s) \tag{7}$$

$$P(T_s^+ | s) = \prod_{s, t_s^+ \in T_s^+} \frac{\exp(\gamma \text{sim}(s, t_s^+))}{\sum_{t \in T} \exp(\gamma \text{sim}(s, t))} \tag{8}$$

$$\text{sim}(s, t_s^+) = \cos(y_s, y_{t_s^+}) \tag{9}$$

where cos is cosine similarity of two encoded vectors. Please refer to the original paper (Huang et al., 2013) for further details. The dropout ratio,

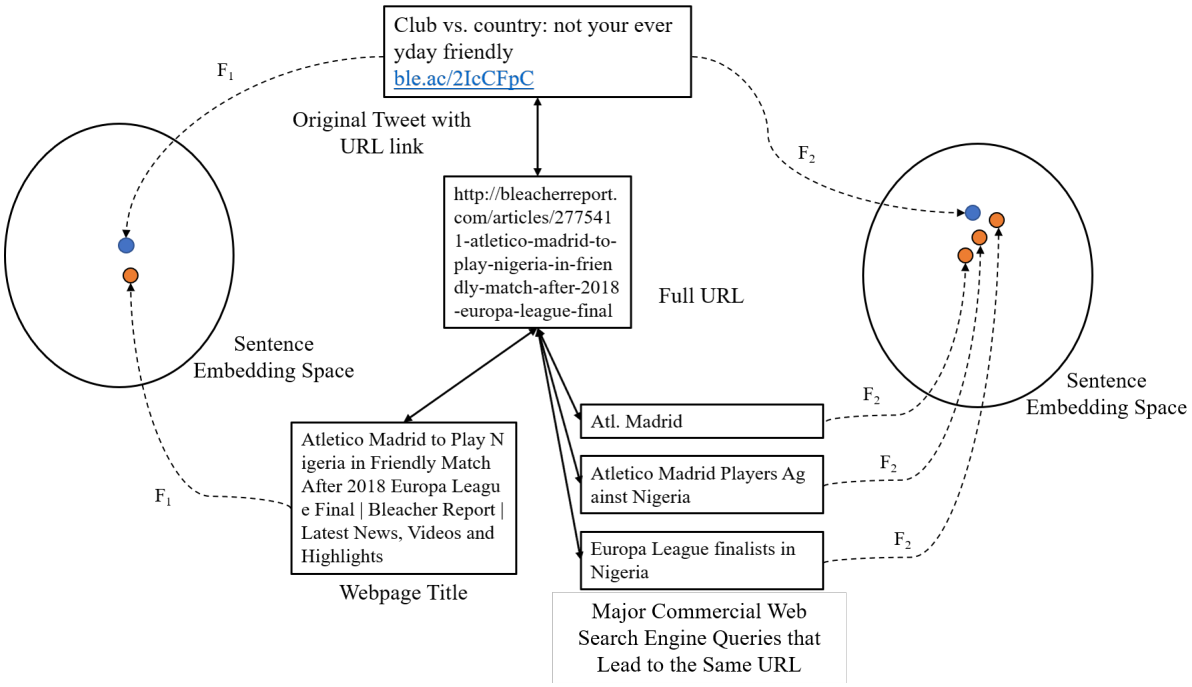

Figure 3: Example of H2H conversation text with URL link and corresponding texts found by following the URL. We use those two sources of corresponding texts to train sentence embedding models. Each model projects the original text and its corresponding texts to a close position in the sentence embedding space, while non-relevant texts are being apart.

learning rate, and γ are selected based on a random search (Bergstra and Bengio, 2012), which are 0.0275, 0.4035×10^{-2}, and 15 respectively. The output of the second dense layer y of trained model is used as the sentence embedding: for each sentence we extract the sentence embedding $v_s \in \mathbb{R}^{300}$.

3.2.2 Contextual Information

Contextual information extracted from previous sentences is known to be useful to improve understanding of human spoken language on other scenarios (Xu and Sarikaya, 2014; Chen et al., 2016; Su et al., 2018). To obtain knowledge from a previous sentence in the conversation, we extract a contextual encoded vector using the memory network (Chen et al., 2016), which uses the weighted sum of the output of word-level bidirectional LSTM h in the core network (Section 3.1) from previous sentences. We did not consider a time decaying model (Su et al., 2018) since our data has a small number of turns.

We tested the model with some variations on 1) number of previous sentences to use and 2) weighting scheme (uniform or with attention). using the implementation from the original paper[4]. From our experiments, the best result was achieved using the previous two sentences with a uniform weight. We use this model to extract the contextual encoded vector $v_c \in \mathbb{R}^{100}$.

3.2.3 Human-to-Machine Expert Feedback

Kim et al. (2017) and Jha et al. (2018) introduced a transfer learning method, which reuses the knowledge from existing trained models on relevant domains (i.e. expert models) to take advantage of previous knowledge to train on a new domain. They extract the output of the expert model and use it as an additional input of feed forward layer for the model on a new domain.

We adopt this idea to take advantage of massive amount of labeled data for H2M conversations. Instead of transferring knowledge from domain to domain, we transfer the knowledge of different tasks within a similar domain. For example, we use **Places (H2M)** domain for the **Restaurant (H2H)** domain, and **Entertainment (H2M)** domain for the **Music (H2H)** domain. We use previously trained slot tagging models on H2M conversations on similar domains as our expert model, which has the same architecture as our core net-

[4] https://github.com/yvchen/ContextualSLU

work (Section 3.1). These H2M models were originally used for the SLU component of a commercial digital assistant. The output of word-level bidirectional LSTM h is then extracted as the encoded vector from H2M expert model $v_e \in \mathbb{R}^{200}$.

3.3 Network Ensemble Approaches

Since additional network components (sentence embedding v_s, contextual information from previous turns of the conversation v_c, and H2M based expert feedback v_e) bring information from different perspectives, we discuss how to compose them into a single vector k with various ensemble approaches.

- **Concatenation**: Here, we simply concatenate all encodings into a single vector,

$$k = v_s \oplus v_c \oplus v_e \tag{10}$$

- **Mean:** We first apply a separate dense layer to each encoded vector to match dimensions and transform into the same latent space, and then take the arithmetic mean of transformed vectors.

$$v'_{\{s,c,e\}} = W_{\{s,c,e\}} + b_{\{s,c,e\}} \tag{11}$$
$$k = \text{mean}(v'_s, v'_c, v'_e) \tag{12}$$

  In the Figure 2, we denote the dense layer applied to each encoded vector $v_{\{s,c,e\}}$ as $D_{\{s,c,e\}}$ for simplicity of representation. Each transformed vector $v'_{\{s,c,e\}} \in \mathbb{R}^{100}$, so $k \in \mathbb{R}^{100}$.

- **Attention:** We apply an attention mechanism to apply different weights on the encoded vectors for each sentence. For our problem, it is not straightforward to define a context vector for each sentence to calculate the importance of each encoded vector; therefore, we adopted the idea of using a global context vector (Yang et al., 2016). The global context vector $u \in \mathbb{R}^{100}$ can be thought as a fixed query of "*finding the informative encoded vector for slot tagging*" used for each sentence. The weight of each encoded vector is calculated with the standard equation of calculating the attention weight, which is the softmax of the dot product of encoding and context vector,

$$w_{\{s,c,e\}} = \frac{\exp(\tanh(v'_{\{s,c,e\}})^\intercal u)}{\sum_{v' \in \{v'_s, v'_c, v'_e\}} \exp(\tanh(v')^\intercal u))} \tag{13}$$

$$k = w_s v'_s + w_c v'_c + w_e v'_e \tag{14}$$

  where $v'_{\{s,c,e\}}$ are same as Equation 11.

The combined single vector k is then aggregated with the output of core network h, $k \oplus h$ is used as the input of the final feed forward layer as shown in Figure 2. The same hyperparameters (mini batch size, learning rate, dropout ratio) and optimizer is used as stated in the baseline model (Section 3.1).

4 Data

Although some datasets with H2H conversations are available (Forsythand and Martell, 2007; Danescu-Niculescu-Mizil and Lee, 2011; Nio et al., 2014; Sordoni et al., 2015; Lowe et al., 2015; Li et al., 2017), they were not feasible to use for experimenting on our task. All datasets excluding the Ubuntu Dialogue (Lowe et al., 2015) were collected without any restrictions on the domain and, as a result, there were insufficient training samples to train a slot tagging model for a specific domain. In addition, the Ubuntu Dialogue dataset (Lowe et al., 2015) focuses on questions related to Ubuntu OS, which is not an attractive domain an intelligent focus that focuses on task completion rather than question answering.

Since there were no existing datasets that were sufficient for our task in H2H conversation, we built our own dataset for the experiments. It was difficult to acquire actual H2H conversations from instant messages due to privacy concerns. Therefore, we chose to use public conversations on Twitter and extracted sequences in which two users engage in a multi-turn conversation. Using this approach, we were able to collect 3.8 million sequences of four-turn conversations using Twitter Firehose.

We focused on two domains for our experiments: **Restaurants** and **Music**. To acquire the dataset for each domain, we first defined a set of key phrases and found the candidate conversations with at least one of those key phrases. Key phrases consisted of the top 100 most frequently used unigrams and bigrams on each relevant domain from the H2M conversation dataset. We used

Restaurant
A: *[lunch]*$_{meal\_type}$?
B: *[lunch]*$_{meal\_type}$ sounds good. Our routine usually involves sitting at *[Nano's]*$_{place\_name}$ with our packed/purchased *[lunches]*$_{meal\_type}$ care to join?
A: great. I'll get something from *[physiol]*$_{place\_name}$ and meet you there at...?
B: I'll be there in *[5-10 mins]*$_{time}$
Music
A: *[quavo]*$_{media\_person}$ got another one
B: I was bout to listen earlier but it said *[feat]*$_{media\_role}$ *[Lil Uzi Vert]*$_{media\_person}$ lol
A: he *[rap]*$_{media\_genre}$ for about two minutes you don't even gotta listen to *[lil uzi]*$_{media\_person}$
B: *[quavo]*$_{media\_person}$ already a legend man

Table 1: Example conversation in each domain of our dataset

the H2M **Places** domain to find the top n-grams for the **Restaurant** domain and the H2M **Entertainment** domain to find top n-grams for the **Music** domain. **Places** includes other type of places besides restaurants (e.g. tour sights), and also **Entertainment** includes other genre (e.g. movies). So we manually replaced unigrams and bigrams that were not music or entertainment related, and also some terms that are too general (e.g. time, call, find). We were able to gather 16K and 22K candidate conversations for the **Restaurant** and **Music** domains, respectively, using the keyphrases.

We randomly sampled 10K conversations for each domain for annotating slots and domain. Annotation was done by managed judges, who had been trained over time for annotating SLU components such as intents, slots and domains. A guideline document was provided with the precise definition and annotated examples of each of the slots and intents. Agreement between judges and manual inspection of samples for quality assurance was done by a linguist trained for managing annotation tasks. We also ensured that judges did not attempt to guess at the underlying intents and slots, and annotate objectively within the context from the text. We only keep the conversations that are labeled relevant to each domain by annotators. Table 1 shows an example conversation from the dataset in each domain, and Table 2 shows the dataset statistics.

Domain	#Conv.	#Words/Conv.	#Slots
Restaurant	6,514	37.64	15[5]
Music	5,582	44.72	20[6]

Table 2: Statistics of our dataset. Each column shows the number of items in the dataset. "Conv." stands for conversations.

5 Experiments

5.1 Experimental Setup

All experiments were done with 10-fold cross validation for the slot tagging task, and we generated training, development, test datasets using 80%, 10%, and 10% of the data. The development dataset is used for hyperparameter tuning with random search (Bergstra and Bengio, 2012) and early stopping. The baseline is set with core network only (Section 3.1). We evaluated the performance of each of the models with precision, recall, and F1. We checked for statistical significance over the baseline at the p-value < 0.05 using the Wilcoxon signed-rank test.

5.2 Evaluation on Adding Sentence Embeddings for H2H Conversations

In this section, we evaluate adding the sentence embeddings into our slot tagging architecture introduced in Section 3.2.1. Table 3 shows the results of adding sentence embeddings, compared with the baseline and existing sentence embedding methods. We extracted two months of recent tweets that had non-twitter domain URLs in the text for our method. Below is the brief description of each method:

- DSSM (Deep Structured Semantic Model) (Huang et al., 2013): Pre-trained DSSM model from the authors , trained with pairs of *(Major commercial web search engine queries, clicked page titles)*.
- Tweet2Vec (Dhingra et al., 2016): The model was originally used to predict hashtags of a

[5]Slots in **Restaurant** domain include absolute_location, amenities, atmosphere, cuisine, date, distance, meal_type, open_status, place_name, place_type, price_range, product, rating, service_provided, time.

[6]Slots in **Music** domain include app_name, media_award, media_category, media_content_rating, media_genre, media_keyword, media_language, media_lyrics, media_nationality, media_person, media_price, media_release_date, media_role, media_source, media_technical_type, media_title, media_type, media_user_rating, radio_call_sign, radio_frequency

Model	Restaurant			Music		
	P	R	F1	P	R	F1
Core Network (Baseline)	71.23	62.68	66.63	**64.33**	44.14	51.61
+ DSSM (Huang et al., 2013)	70.82	61.60	65.88*	62.83	43.76	51.57
+ Tweet2Vec (Dhingra et al., 2016)	70.58	59.99	64.75*	62.67	41.38	49.79*
+ Ours *(Tweets, Web Search Engine Queries)*	**71.73**	62.38	66.70	63.13	44.09	51.86
+ Ours *(Tweets, Web Paage Titles)*	71.19	**63.51**	**67.11***	63.70	**44.44**	**52.32**

Table 3: Comparison of adding sentence embedding component to our architecture. P, R, F1 stands for precision, recall, F1-score (%) respectively. * denotes the F1-score is statistically significant compared to the baseline.

tweet. We use the pre-trained model from the authors, which used 2M tweets for training.

- Ours *(Tweets, Web Search Engine Queries)*: Trained our model with 2.35M pairs of *(Tweet text with shared URL, Web serach engine queries that lead to the shared URL)*. We extracted most frequent queries (up to eight) found from the major commercial web search engine query logs.
- Ours *(Tweets, Web Page Titles)*: Trained our model with 420K pairs of *(Tweet text with shared URL, web page title of URL)*.

The result shows that adding our proposed sentence embedding network improves the slot tagging result compared to the baseline, while other previous methods have a negative effect. This implies that 1) a sentence embedding specifically trained for H2H conversation texts are needed (compared with original DSSM), 2) our idea of embedding semantics and intentions from web data and search engine query logs can help to improve the slot tagging task (compared to the Tweet2Vec). Since our sentence embedding network trained with web page titles gives the most significant improvement, we used this for further evaluation.

5.3 Evaluation on Utilizing Knowledge Sources

We also tested adding contextual information and H2M expert feedback network components to our slot tagging architecture. Contextual information is extracted from previous two sentences with uniform weighting. For the H2M expert model, we used the pretrained model of **Entertainment** domain for the target **Music** domain, and the **Places** domain for the target **Restaurant** domain for H2M LU.

The upper part (rows 2-4) in Table 4 shows the result of adding each. Results show that 1) adding network component from each knowledge source leads to an improvement on at least one of the domain, 2) improvement on each method varies with the domain. Adding sentence embeddings and contextual information led to significant improvements for the **Restaurant** domain while contextual information and H2M expert feedback led to significant improvements for the **Music** domain.

5.4 Evaulation on Network Ensemble Approaches

We also conducted an experiment to include all network components to see if we can improve further by considering multiple knowledge sources together. The result is shown in the lower part (row 5-7) of Table 4 with different ensembling methods introduced in Section 3.3. It shows that any of the ensemble approaches to add all of the network components leads to better results than adding either of them individually.

The result implies that each of the proposed method improves the slot tagging method from different perspectives so all of them can be considered. Also, we see that attention has the best results among ensemble approaches, with 2.62% higher F1 score for the **Restaurant** domain, and 6.09% for the **Music** domain compared to the baseline. This implies the attention model can help to find the best way to ensemble additional components by predicting the importance of each component for each sentence. Especially, we could see a statistically significant improvement on the **Music** domain compared with other methods. We believe this is because the improvement of each network component on the **Music** domain is more obvious compared to the **Restaurant** domain. We would like to test in other domains for the future work.

Model	Restaurant			Music		
	P	R	F1	P	R	F1
Core Network (Baseline)	71.23	62.68	66.63	64.33	44.14	51.61
+ Sentence Embedding	71.19	63.51	67.11*	63.70	44.44	52.32
+ Contextual	72.74	64.75	68.47*	64.42	49.16	55.72*
+ H2M Expert	71.91	61.21	66.63	64.14	44.67	52.64*
+ Ensemble (Concatenation)	**74.09**	64.89	69.21*	**66.41**	50.10	57.07*
+ Ensemble (Mean)	73.20	**65.43**	69.07*	65.18	51.01	57.11*
+ Ensemble (Attention)	74.03	65.11	**69.25***	66.19	**51.29**	**57.70****

Table 4: Comparison on adding additional network components from each knowledge source and network ensemble approaches that adds all components. P, R, F1 stands for precision, recall, F1-score (%) respectively. * denotes the F1 score is statistically significant compared to the baseline. ** denotes the F1 score of ensemble model is also statistically significant compared to the concatenation ensemble model.

6 Conclusion

We studied slot tagging in H2H online text conversations. Starting from a core network with bidirectional LSTM, we proposed to use additional network components and ensemble them to augment useful knowledge from multiple sources (web data, search engine click logs, H2M expert feedback, and previous utterances). Experiments with our four-turn Twitter dataset on **Restaurant** and **Music** domains showed that our method improves up to 6.09%-points higher F1 on slot tagging compared to existing approaches. For future work, we plan to study our model on domain and intent classification, and also on additional domains.

Acknowledgement

We would like to thank Eric Mei and Soumya Batra for their help on data collection and labeling.

References

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. Sequential dialogue context modeling for spoken language understanding. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 103–114.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Yun-Nung Chen, Dilek Hakkani-Tür, Gökhan Tür, Jianfeng Gao, and Li Deng. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *INTERSPEECH*, pages 3245–3249.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics.

Li Deng, Gokhan Tur, Xiaodong He, and Dilek Hakkani-Tr. 2012. Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *IEEE Workshop on Spoken Language Technologies (SLT)*.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *The 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, page 269.

Eric N Forsythand and Craig H Martell. 2007. Lexical and discourse analysis of online chat dialog. In *International Conference on Semantic Computing (ICSC 2007)*, pages 19–26. IEEE.

Anuj Kumar Goyal, Angeliki Metallinou, and Spyros Matsoukas. 2018. Fast and scalable expansion of natural language understanding functionality for intelligent agents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 3, pages 145–152.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management (CIKM)*, pages 2333–2338.

Youngsoo Jang, Jiyeon Han, Byung-Jun Lee, and Kee-Eung Kim. 2018. Cross-language neural dialog state tracker for large ontologies using hierarchical attention. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2072–2082.

Rahul Jha, Alex Marin, Suvamsh Shivaprasad, and Imed Zitouni. 2018. Bag of experts architectures for model reuse in conversational language understanding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 3, pages 153–161.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Domain attention with an ensemble of experts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 643–653.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: Amethod for stochastic optimization. In *Proceedings of the 3rd International Conference of Learning Representations (ICLR)*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 986–995.

Ryan Lowe, Nissan Pow, Iulian V Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, page 285.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura. 2014. Conversation dialog corpora from television and movie scripts. In *2014 17th Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA)*, pages 1–4. IEEE.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):694–707.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Suman Ravuri and Andreas Stolcke. 2015. Recurrent neural network and lstm models for lexical utterance classification. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Ruhi Sarikaya, Paul A Crook, Alex Marin, Minwoo Jeong, Jean-Philippe Robichaud, Asli Celikyilmaz, Young-Bum Kim, Alexandre Rochette, Omar Zia Khan, Xiaohu Liu, et al. 2016. An overview of end-to-end language understanding and dialog management for personal digital assistants. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 391–397.

Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. 2011. Deep belief nets for natural language call-routing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5680–5683.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, pages 101–110.

Hongjie Shi, Takashi Ushio, Mitsuru Endo, Katsuyoshi Yamagami, and Noriaki Horii. 2016. A multichannel convolutional neural network for cross-language dialog state tracking. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 559–564.

Hongjie Shi, Takashi Ushio, Mitsuru Endo, Katsuyoshi Yamagami, and Noriaki Horii. 2017. Convolutional neural networks for multi-topic dialog state tracking. In *Dialogues with Social Robots*, pages 451–463. Springer.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Shang-Yu Su, Pei-Chieh Yuan, and Yun-Nung Chen. 2018. How time matters: Learning time-decay attention for contextual spoken language understanding in dialogues. In *Proceedings of the 2018 Conference of the North American Chapter of the Asso-*

ciation for Computational Linguistics: Human Language Technologies (NAACL-HLT), volume 1, pages 2133–2142.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems (NIPS)*, pages 2440–2448.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. 2016. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM International conference on Research and Development in Information Retrieval (SIGIR)*, pages 1041–1044.

Puyang Xu and Ruhi Sarikaya. 2014. Contextual domain classification in spoken language understanding systems using recurrent neural network. In *IEEE Internation Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 136–140.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1480–1489.