A Scalable Neural Shortlisting-Reranking Approach for Large-Scale Domain Classification in Natural Language Understanding

Young-Bum Kim **Dongchan Kim** **Joo-Kyung Kim** **Ruhi Sarikaya**
Amazon Alexa
{youngbum, dongchan, jookyk, rsarikay}@amazon.com

Abstract

Intelligent personal digital assistants (IPDAs), a popular real-life application with spoken language understanding capabilities, can cover potentially thousands of overlapping domains for natural language understanding, and the task of finding the best domain to handle an utterance becomes a challenging problem on a large scale. In this paper, we propose a set of efficient and scalable neural shortlisting-reranking models for large-scale domain classification in IPDAs. The shortlisting stage focuses on efficiently trimming all domains down to a list of k-best candidate domains, and the reranking stage performs a list-wise reranking of the initial k-best domains with additional contextual information. We show the effectiveness of our approach with extensive experiments on 1,500 IPDA domains.

1 Introduction

Natural language understanding (NLU) is a core component of intelligent personal digital assistants (IPDAs) such as Amazon Alexa, Google Assistant, Apple Siri, and Microsoft Cortana (Sarikaya, 2017). A well-established approach in current real-time systems is to classify an utterance into a domain, followed by domain-specific intent classification and slot sequence tagging (Tur and de Mori, 2011). A domain is typically defined in terms of a specific application or a functionality such as weather, calendar and music, which narrows down the scope of NLU for a given utterance. A domain can also be defined as a collection of relevant intents; assuming an utterance belongs to the calendar domain, possible intents could be to create a meeting or cancel one, and possible extracted slots could be people names, meeting title and date from the utterance. Traditional IPDAs cover only tens of domains that share a common schema. The schema is designed to separate out the domains in an effort to minimize language ambiguity. A shared schema, while addressing domain ambiguity, becomes a bottleneck as new domains and intents are added to cover new scenarios. Redefining the domain, intent and slot boundaries requires relabeling of the underlying data, which is very costly and time-consuming. On the other hand, when thousands of domains evolve independently without a shared schema, finding the most relevant domain to handle an utterance among thousands of overlapping domains emerges as a key challenge.

The difficulty of solving this problem at scale has led to stopgap solutions, such as requiring an utterance to explicitly mention a domain name and restricting the expression to be in a predefined form as in “Ask `ALLRECIPES`, how can I bake an apple pie?” However, such solutions lead to an unintuitive and unnatural way of conversing and create interaction friction for the end users. For the example utterance, a more natural way of saying it is simply, “How can I bake an apple pie?” but the most relevant domain to handle it now becomes ambiguous. There could be a number of candidate domains and even multiple overlapping recipe-related domains that could handle it.

In this paper, we propose efficient and scalable shortlisting-reranking neural models in two steps for effective large-scale domain classification in IPDAs. The first step uses light-weight BiLSTM models that leverage only the character and word-level information to efficiently find the k-best list of most likely domains. The second step uses rich contextual information later in the pipeline and applies another BiLSTM model to a list-wise ranking task to further rerank the k-best domains to find the most relevant one. We show the effectiveness of our approach for large-scale domain classification with an extensive set of experiments on 1,500 IPDA domains.

Proceedings of NAACL-HLT 2018, pages 16–24
New Orleans, Louisiana, June 1 - 6, 2018. ©2017 Association for Computational Linguistics

2 Related Work

Reranking approaches attempt to improve upon an initial ranking by considering additional contextual information. Initial model outputs are trimmed down to a subset of most likely candidates, and each candidate is combined with additional features to form a hypothesis to be re-scored. Reranking has been applied to various natural language processing tasks, including machine translation (Shen et al., 2004), parsing (Collins and Koo, 2005), sentence boundary detection (Roark et al., 2006), named entity recognition (Nguyen et al., 2010), and supertagging (Chen et al., 2002).

In the context of NLU or SLU systems, Morbini et al. (2012) showed a reranking approach using k-best lists from multiple automatic speech recognition (ASR) engines to improve response category classification for virtual museum guides. Basili et al. (2013) showed that reranking multiple ASR candidates by analyzing their syntactic properties can improve spoken command understanding in human-robot interaction, but with more focus on ASR improvement. Xu and Sarikaya (2014) showed that multi-turn contextual information and recurrent neural networks can improve domain classification in a multi-domain and multi-turn NLU system. There have been many other pieces of prior work on improving NLU systems with pre-training (Kim et al., 2015b; Celikyilmaz et al., 2016; Kim et al., 2017e), multi-task learning (Zhang and Wang, 2016; Liu and Lane, 2016; Kim et al., 2017b), transfer learning (El-Kahky et al., 2014; Kim et al., 2015a,c; Chen et al., 2016a; Yang et al., 2017), domain adaptation (Kim et al., 2016; Jaech et al., 2016; Liu and Lane, 2017; Kim et al., 2017d,c) and contextual signals (Bhargava et al., 2013; Chen et al., 2016b; Hori et al., 2016; Kim et al., 2017a).

To our knowledge, the work by Robichaud et al. (2014); Crook et al. (2015); Khan et al. (2015) is most closely related to this paper. Their approach is to first run a complete pass of all 3 NLU models of binary domain classification, multi-class intent classification, and sequence tagging of slots across all domains. Then, a hypothesis is formed per domain using the semantic information provided by the domain-intent-slot outputs as well as many other contextual and cross-hypothesis features such as the presence of a slot tagging type in any other hypotheses. Reranking the hypotheses with Gradient Boosted Decision Trees (Friedman, 2001; Burges et al., 2011) has been shown to improve domain classification performance compared to using only domain classifiers without reranking.

Their approach however suffers from the following two limitations. First, it requires running all domain-intent-slot models in parallel across all domains. Their work considers only 8 or 9 distinct domains, and the approach has serious practical scaling issues when the number of domains scales to thousands. Second, contextual information, especially cross-hypothesis features, that is crucial for reranking is manually designed at the feature level with a sparse representation.

Our work in this paper addresses both of these limitations with a scalable and efficient two-step shortlisting-reranking approach, which has a neural ranking model capturing cross-hypothesis features automatically. To our knowledge, this work is the first in the literature on large-scale domain classification for a real IPDA production system with a scale of thousands of domains. Our LSTM-based list-wise ranking approach also makes a novel contribution to the existing literature in the context of IPDA and NLU systems. In this work, we limit our scope to first-turn utterances and leave multi-turn conversations for future work.

3 Shortlisting-Reranking Architecture

Our shortlisting-reranking models process an incoming utterance as follows. (1) *Shortlister* performs a naive, fast ranking of all domains to find the k-best list using only the character and word-level information. The goal here is to achieve high domain recall with maximal efficiency and minimal information and latency. (2) For each domain in the k-best list, we prepare a hypothesis per domain with additional contextual information, including domain-intent-slot semantic analysis, user preferences, and domain index of popularity and quality. (3) A second ranker called *Hypotheses Reranker (HypRank)* performs a list-wise ranking of the k hypotheses to improve on the initial naive ranking and find the best hypothesis, thus domain, to handle the utterance.

Figure 1 illustrates the steps with an example utterance, "*play michael jackson*." Based on character and word features, *shortlister* returns the k-best list in the order of `CLASSIC MUSIC, POP MUSIC, and VIDEO` domains. `CLASSIC MUSIC` outputs `PlayTune` intent, but without any slots, low domain popularity, and no usage history for the user, its ranking is adjusted to be last. `POP MUSIC` outputs `PlayMusic` intent and `Singer` slot for "*michael jackson*", and with frequent user usage history, it is determined to be the best domain to handle the utterance.

In our architecture, key focus is on efficiency and scalability. Running full domain-intent-slot semantic analysis for thousands of domains imposes a significant computational burden in terms of memory footprint, latency and number of machines, and it is impractical in real-time systems. For the same reason, this work only uses contextual information in the reranking stage, and the utility of including it in the shortlisting stage is left for future work.

4 Shortlister

Shortlister consists of three layers: an orthography-sensitive character and word embedding layer, a BiLSTM layer that makes a vector representation for the words in a given utterance, and an output layer for domain classification. Figure 2 shows the overall shortlister architecture.

Embedding layer In order to capture character-level patterns, we construct an orthography-sensitive word embedding layer (Ling et al., 2015; Ballesteros et al., 2015). Let $\mathcal{C}$, $\mathcal{W}$, and $\oplus$ denote the set of characters, the set of words, and the vector concatenation operator, respectively. We represent an LSTM as a mapping $\phi : \mathbb{R}^d \times \mathbb{R}^{d'} \rightarrow \mathbb{R}^{d'}$ that takes an input vector x and a state vector h to output a new state vector $h' = \phi(x, h)$[1]. The model parameters associated with this layer are:

$$\begin{aligned}
\textbf{Char embedding: } & e_c \in \mathbb{R}^{25} \text{ for each } c \in \mathcal{C} \\
\textbf{Char LSTMs: } & \phi_f^{\mathcal{C}}, \phi_b^{\mathcal{C}} : \mathbb{R}^{25} \times \mathbb{R}^{25} \rightarrow \mathbb{R}^{25} \\
\textbf{Word embedding: } & e_w \in \mathbb{R}^{100} \text{ for each } w \in \mathcal{W}
\end{aligned}$$

Let $(w_1, \ldots, w_m)$ denote a word sequence where word $w_i \in \mathcal{W}$ has character $w_i(j) \in \mathcal{C}$ at position j. This layer computes an orthography-sensitive word representation $v_i \in \mathbb{R}^{150}$ as:[2]

$$\begin{aligned}
f_j^{\mathcal{C}} &= \phi_f^{\mathcal{C}}\left(e_{w_i(j)}, f_{j-1}^{\mathcal{C}}\right) && \forall j = 1 \ldots |w_i| \\
b_j^{\mathcal{C}} &= \phi_b^{\mathcal{C}}\left(e_{w_i(j)}, b_{j+1}^{\mathcal{C}}\right) && \forall j = |w_i| \ldots 1 \\
v_i &= f_{|w_i|}^{\mathcal{C}} \oplus b_1^{\mathcal{C}} \oplus e_{w_i}
\end{aligned}$$

BiLSTM layer We utilize a BiLSTM to encode the word vector sequence $(v_1, \ldots, v_m)$. The BiLSTM outputs are generated as:

$$\begin{aligned}
f_i^{\mathcal{W}} &= \phi_f^{\mathcal{W}}\left(v_i, f_{i-1}^{\mathcal{W}}\right) && \forall i = 1 \ldots m \\
b_i^{\mathcal{W}} &= \phi_b^{\mathcal{W}}\left(v_i, b_{i+1}^{\mathcal{W}}\right) && \forall i = m \ldots 1
\end{aligned}$$

where $\phi_f^{\mathcal{W}}, \phi_f^{\mathcal{W}} : \mathbb{R}^{150} \times \mathbb{R}^{100} \rightarrow \mathbb{R}^{100}$ are the forward LSTM and the backward LSTM, respectively. An utterance representation $h \in \mathbb{R}^{200}$ is induced by concatenating the outputs of the both LSTMs as:

$$h = f_m^{\mathcal{W}} \oplus b_1^{\mathcal{W}}$$

Output layer We map the word LSTM output h to a n-dimensional output vector with a linear transformation. Then, we take a softmax function either over the entire domains ($softmax_a$) or over two classes (in-domain or out-of-domain) for each domain ($softmax_b$).

$softmax_a$ is used to set the sum of the confidence scores over the entire domains to be 1. We can obtain the outputs as:

$$o = softmax\left(W \cdot h + b\right)$$

where W and b are parameters for a linear transformation.

For training, we use cross-entropy loss, which is formulated as follows:

$$\mathcal{L}_a = -\sum_{i=1}^{n} l_i \log o_i \tag{1}$$

where l is a n-dimensional one-hot vector whose element corresponding to the position of the ground-truth hypothesis is set to 1.

$softmax_b$ is used to set the confidence score for each domain to be between 0 and 1. While $softmax_a$ tends to highlight only the ground-truth domain while suppressing all the rest, $softmax_b$ is designed to produce a more balanced confidence score per domain independent of other domains. When using $softmax_b$, we obtain a 2-dimensional output vector for each domain as follows:

$$o^i = softmax\left(W^i \cdot h + b^i\right)$$

where W^i is a 2 by 200 matrix and b^i is a 2-dimensional vector; o_1^i and o_2^i denote the in-domain probability and the out-of-domain probability, respectively. The loss function is formulated as follows:

$$\mathcal{L}_b = -\sum_{i=1}^{n} \left\{ l_i \log o_1^i + \frac{1 - l_i}{n - 1} \log o_2^i \right\} \tag{2}$$

[1] We omit cell variable notations for simple LSTM formulations.

[2] We randomly initialize state vectors such as $f_0^{\mathcal{C}}$ and $b_{|w_i|+1}^{\mathcal{C}}$.

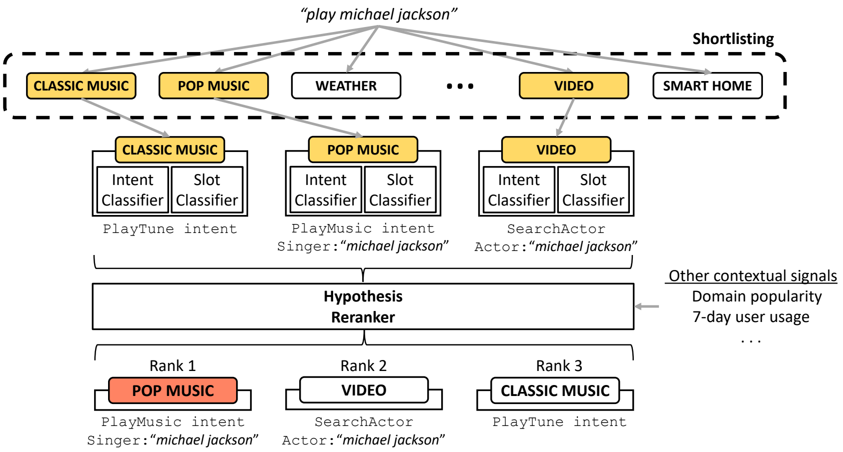

Figure 1: A high-level flow of our two-step shortlisting-reranking approach given an utterance to an IPDA.

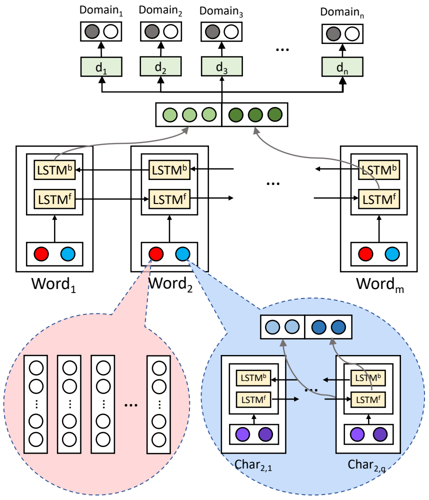

Figure 2: The architecture of our neural shortlisting model that uses character and word-level information of a given utterance.

where we divide the second term by $n-1$ so that o_1^i and o_2^i are balanced in terms of the ratio of the training examples for a domain to those for other domains.

5 Hypotheses Reranker (HypRank)

Hypotheses Reranker (HypRank) comprises of two components: hypothesis representation and a BiLSTM model for reranking a list of hypotheses. We use the term reranking since we improve upon the initial ranking from Shortlister's k-best list. In our problem context, a hypothesis is formed per domain with additional semantic and contextual information, and selecting the highest-scored hypothesis means selecting the domain represented in that hypothesis for final domain classification.

HypRank, illustrated in Figure 3, is a list-wise ranking approach in that it considers the entire list of hypotheses before giving a reranking score for each hypothesis. While previous work manually encoded cross-hypothesis information at the feature level (Robichaud et al., 2014; Crook et al., 2015; Khan et al., 2015), our approach is to let a BiLSTM layer automatically capture that information and learn appropriate representations at the model level. In addition to giving detail of useful contextual signals for IPDAs, we also introduce the use of pre-trained domain, intent and slot embeddings in this section.

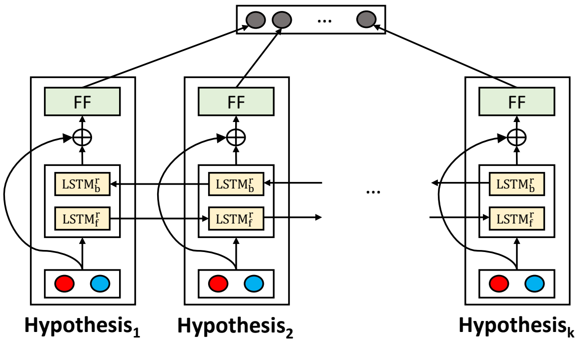

Figure 3: The architecture of our neural Hypotheses Reranker model that takes in k hypotheses with rich contextual information for more refined ranking.

5.1 Hypothesis Representation

A hypothesis is formed for each domain with the following three categories of contextual information: NLU interpretation, user preferences, and domain index.

NLU interpretation Each domain has three corresponding NLU models for binary domain classification, multi-class intent classification, and sequence tagging for slots. From the domain-intent-slot semantic analysis, we use the confidence score from the shortlister, the intent classification confidence, Viterbi path score of the slot sequence from a slot tagger, and the average confidence score of the tagged slots[3].

To pre-train domain embeddings, we use a word-level BiLSTM with each utterance as a sequence of word embedding vector $\in \mathbb{R}^{100}$ in the input layer. The BiLSTM outputs, each a vector $\in \mathbb{R}^{25}$, are concatenated and projected to an output vector for all domains in the output layer. The learned projection weight matrix is extracted as domain embeddings. The output vector dimension used was $\in \mathbb{R}^{1500}$ for the large-scale setting and $\in \mathbb{R}^{20}$ for the traditional small-scale setting in our experiments (Section 6.1). For intent and slot embeddings, we take the same process with the only difference in the output vector with the dimension $\in \mathbb{R}^{6991}$ for all unique intents across all domains and with the dimension $\in \mathbb{R}^{2845}$ for all unique slots.

Once pre-trained, the domain or intent embeddings are used simply as a lookup table per domain or per intent. For slot embeddings, there can be more than one slot per utterance, and in case of multiple tagged slots, we sum up each slot embedding vector to combine the information. In summary, these are the three domain-intent-slots embeddings we used: $e^d \in \mathbb{R}^{50}$ for a domain vector, $e^i \in \mathbb{R}^{50}$ for an intent vector, and $e^s \in \mathbb{R}^{50}$ for a vector of slots.

User Preferences User-specific signals are designed to capture each user's behavioral history or preferences. In particular, we encode whether a user has specific domains enabled in his/her IPDA setting and whether he/she triggered certain domains within 7, 14 or 30 days in the past.

Domain Index From this category, we encode domain popularity and quality as rated by the user population. For example, if the utterance "*I need a ride to work*" can be equally handled by `TAXI_A` domain or `TAXI_B` domain but the user has never used any, the signals in this category could give a boost to `TAXI_A` domain due to its higher popularity.

5.2 HypRank Model

The proposed model is trained to rerank the domain hypotheses formed from Shortlister results. Let $(p_1, \dots, p_k)$ be the sequence of k input hypothesis vectors that are sorted in decreasing order of Shortlister scores.

Category	$\lvert\mathcal{D}\rvert$	Example
`Device`	177	smart home, smart car
`Food`	99	recipe, nutrition
`Ent.`	465	movie, music, game
`Info.`	399	travel, lifestyle
`News`	159	local, sports, finance
`Shopping`	39	retail, food, media
`Util.`	162	productivity, weather
Total	1,500	

Table 1: The categories of the 1,500 domains for our large-scale IPDA. $\lvert\mathcal{D}\rvert$ denotes the number of domains.

We utilize a BiLSTM layer for transforming the input sequence to the BiLSTM output sequence $(h_1, \dots, h_k)$ as follows:

$$\begin{aligned} f_i^r &= \phi_f^r\left(p_i, f_{i-1}^r\right) \\ b_i^r &= \phi_b^r\left(p_i, b_{i+1}^r\right) \\ h_i &= f_i^r \oplus b_i^r \qquad \forall i \in \{1, \dots, k\}, \end{aligned}$$

where ϕ_f^r and ϕ_b^r are the forward LSTM and the backward LSTM, respectively.

Since the BiLSTM utilizes both the previous and the next sub-sequences as the context, each of the BiLSTM outputs is computed considering cross-hypothesis information.

For the i-th hypothesis, we either sum or concatenate the input vector and the BiLSTM output to utilize both of them as an intermediate representation as $g_i = d_i \oplus h_i$. Then, we use a feed-forward neural network with a single hidden layer to transform g to a k-dimensional vector p as follows:

$$p_i = W_2 \cdot \sigma\left(W_1 \cdot g_i + b_1\right) + b_2 \quad \forall i \in \{1, \dots, k\},$$

where σ indicates scaled exponential linear unit (SeLU) for normalized activation outputs (Klambauer et al., 2017); the outputs of all the hypotheses are generated by using the same parameter set $\{W_1, b_1, W_2, b_2\}$ for consistency regardless of the hypothesis order.

Finally, we obtain a k-dimensional output vector o by taking a softmax function:

$$o = softmax\left(p\right).$$

$argmax_i\{o_1, .., o_k\}$ is the index of the predicted hypothesis after the reranking. Cross entropy is used for training as follows:

$$\mathcal{L}_r = -\sum_{i=1}^{k} l_i \log o_i, \tag{3}$$

where l is a k-dimensional ground-truth one-hot vector.

6 Experiments

This section gives detail of our experimental setup, followed by results and discussion.

[3] We use off-the-shelf intent classifiers and slot taggers achieving 98% and 96% accuracies on average, respectively.

	SL train	SL dev	HR train	HR dev	test
Traditional	3M	415K	715K	20K	420K
Large-Scale	5M	530K	830K	20K	530K

Table 2: The number of train, development and test utterances. SL denotes Shortlister and HR denotes HypRank.

6.1 Experimental Setup

We evaluated our shortlisting-reranking approach in two different settings of traditional small-scale IPDA and large-scale IPDA for comparison:

Traditional IPDA For this setting, we simulated the traditional small-scale IPDA with only 20 domains that are commonly present in any IPDAs. Since these domains are built-in, which are carefully designed to be non-overlapping and of high quality, the signals from user preferences and domain index become irrelevant compared to the large-scale setting. The dataset comprises of more than 4M labeled utterances in text evenly distributed across 20+ domains.

Large-Scale IPDA This setting is a large-scale IPDA with 1,500 domains as shown in Table 1 that could be overlapping with a varying level of quality. For instance, there could be multiple domains to get a recipe, and a high quality domain could have more recipes with more capabilities such as making recommendations compared to a low quality one. The dataset comprises of more than 6M utterances having strict invocation patterns. For instance, we extract *"get me a ride"* as a preprocessed sample belonging to *TAXI* skill for the original utterance, *"Ask {TAXI} to {get me a ride}."*

Shortlister For Shortlister, we show the results of using 2 different softmax functions of $softmax_a$ (smx_a) and $softmax_b$ (smx_b) as described in Section 4. The results are shown in k-best classification accuracies, where the 5-best accuracy means the percentage of test samples that have the ground-truth domain included in the top 5 domains returned by Shortlister.

Hypotheses Reranker We also evaluate different variations of the reranking model for comparison.

- SL: Shortlister 1-best result, which is our baseline without using a reranking model.
- LR: LR point-wise: A binary logistic regression model with the hypothesis vector as features (see Section 5.1). We run it for each hypothesis made from Shortlister's k-best list and select the highest-scoring one, hence the domain in that hypothesis.
- N^{PO}: Neural point-wise: A feed-forward (FF) layer between the hypothesis vector and the nonlinear output layer. We run it for each hypothesis made from Shortlister's k-best list and select the highest-scoring hypothesis.
- N^{PA}: Neural pair-wise: A FF layer between the concatenation of two hypothesis vectors and the nonlinear output layer. We run it k - 1 times for a pair of hypotheses in a series of tournament-like competitions in the order of the k-best list. For instance, the 1st and 2nd hypothesis compete first and the winner of the two competes with the 3rd hypothesis next and so on until the kth hypothesis.
- N^{CH}: Neural quasi list-wise with manual cross-hypothesis features added to N^{PO}, following past approaches (Robichaud et al., 2014; Crook et al., 2015; Khan et al., 2015) such as the ratio of Shortlister scores to the maximum score, relative number of slots across all hypotheses, etc.
- $LSTM^{O}$: Using only the BiLSTM output vectors as the input to the FF-layer.
- $LSTM^{S}$: Summing up the hypothesis vector and the BiLSTM output vectors as the input to the FF-layer, similar to residual networks (He et al., 2016).
- $LSTM^{C}$: Concatenating the hypothesis vector and the BiLSTM output vectors as the input to the FF-layer.
- $LSTM^{CH}$: Same as $LSTM^{C}$ except that manual cross-hypothesis features used for N^{CH} were also added to see if combining manual and automatic cross-hypothesis features help.
- $UPPER$: Upper bound of HypRank accuracy set by the performance of Shortlister.

	Traditional IPDA		Large-Scale IPDA	
	smx_a	smx_b	smx_a	smx_b
1-best	95.58	95.56	81.38	81.49
3-best	98.45	98.43	92.53	92.81
5-best	**98.81**	**98.77**	**95.77**	**95.93**

Table 3: The k-best classification accuracies (%) of Shortlister using different softmax functions in the traditional and large-scale IPDA settings.

6.2 Methodology

Table 2 shows the distribution of the training, development and test sets for each setting of traditional and large-scale IPDAs. Note that we ensure

Model	Traditional IPDA		Large-Scale IPDA	
	smx_a	smx_b	smx_a	smx_b
SL	95.58	95.56	81.38	81.49
LR	95.50	95.59	86.74	87.50
N^{PO}	95.26	95.46	88.65	90.38
N^{PA}	96.08	96.37	88.29	90.82
N^{CH}	96.20	96.65	88.43	91.13
$LSTM^{O}$	94.36	94.45	81.37	82.98
$LSTM^{S}$	97.44	97.54	92.43	93.79
$LSTM^{C}$	**97.47**	**97.55**	**92.49**	**93.83**
$LSTM^{CH}$	97.22	97.34	92.18	93.46
$UPPER$	98.81	98.77	95.77	95.93

Table 4: The final classification accuracies (%) for different Shortlisting-HypRank models under the traditional and large-scale IPDA settings. The input hypothesis size is 5.

no overlap between the Shortlister and HypRank training sets so that HypRank is not overly tuned on Shortlister results. For the NLU models, the intent and slot models are trained on roughly 70% of the available training data.

In our experiments, all the models were implemented using Dynet (Neubig et al., 2017) and were trained with Adam (Kingma and Ba, 2015). We used the initial learning rate of 4 $\times 10^{-4}$ and left all the other hyper-parameters as suggested in Kingma and Ba (2015). We also used variational dropout (Gal and Ghahramani, 2016) for regularization.

6.3 Results and Discussion

Table 3 summarizes the k-best classification accuracy results for our Shortlister. With only 20 domains in the traditional IPDA setting, the accuracy is over 95% even when we take 1-best or top domain returned from Shortlister. The accuracy approaches 99% when we consider Shortlister correct if the ground-truth domain is present in the top 5 domains. The results suggest that the character and word-level information by itself, coupled with BiLSTMs, can already show significant discriminative power for our task.

With a scale of 1,500 domains, the results indicate that just using the top domain returned from Shortlister is not enough to have comparable performance shown in the traditional IPDA setting. However, the performance catches up to close to 96% as we include more domains in the k-best list, and although not shown here, it starts to level off at 5-best list. The k-best results from Shortlister set an upper bound for HypRank performance. We note that it could be possible to include more contextual information at the shortlisting stage to bring Shortlister's performance up with some trade-offs in terms of real-time systems, which we leave for future work. In addition, using smx_b shows a tendency of slightly better performance compared to using smx_a, which takes a softmax over all domains and tends to emphasize only the top domain while suppressing all others even when there are many overlapping and very similar domains.

The classification performance after the reranking stage with HypRank using Shortlister's 5-best results is summarized in Table 4. SL shows the results of taking the top domain from Shortlister without any reranking step, and UPPER shows the performance upper bound of HypRank set by the shortlisting stage. In general, the pair-wise approach is shown to be better than the point-wise approaches, with the best performance coming from the list-wise ones. Looking at the lowest accuracy from $LSTM^{O}$, it suggests that the raw hypothesis vectors themselves are important features that should be combined with the cross-hypothesis contextual features from the LSTM outputs for best results. Adding manual cross-hypothesis features to the automatic ones from the LSTM outputs do not improve the performance.

The performance trend is very similar for smx_a and smx_b, but there is a gap between them in the large-scale setting. An explanation for this is similar to that for Shortlister results that smx_a emphasizes only the top domain while suppressing all the rest, which might not be suitable in a large-scale setting with many overlapping domains. For both traditional and large-scale settings, the best accuracy is shown with the list-wise model of $LSTM^{C}$.

7 Conclusion

We have described an efficient and scalable shortlisting-reranking neural models for large-scale domain classification. The models first efficiently prune all domains to only a small number of k candidates using minimal information and subsequently rerank them using additional contextual information that could be more expensive in terms of computing resources. We have shown the effectiveness of our approach with 1,500 domains in a real IPDA system and evaluated using different variations of the shortlisting model and our novel reranking models, in terms of point-wise, pair-wise, and list-wise ranking approaches.

Acknowledgments

We thank Sunghyun Park and Sungjin Lee for helpful discussion and feedback.

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 349–359.

Roberto Basili, Emanuele Bastianelli, Giuseppe Castellucci, Daniele Nardi, and Vittorio Perera. 2013. *Kernel-based discriminative re-ranking for spoken command understanding in HRI*, Springer International Publishing, Cham, pages 169–180.

A. Bhargava, Asli elikyilmaz, Dilek Z. Hakkani-Tur, and Ruhi Sarikaya. 2013. Easy contextual intent prediction and slot detection. *IEEE International Conference on Acoustics, Speech and Signal Processing* pages 8337–8341.

Christopher J C Burges, Krysta M Svore, Paul N. Bennett, Andrzej Pastusiak, and Qiang Wu. 2011. Learning to Rank Using an Ensemble of Lambda-Gradient Models. *Journal of Machine Learning Research (JMLR)* 14:25–35.

Asli Celikyilmaz, Ruhi Sarikaya, Dilek Hakkani-Tür, Xiaohu Liu, Nikhil Ramesh, and Gökhan Tür. 2016. A new pre-training method for training deep learning models with application to spoken language understanding. In *Interspeech*. pages 3255–3259.

John Chen, Srinivas Bangalore, Michael Collins, and Owen Rambow. 2002. Reranking an n-gram supertagger. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+ 6)*. pages 259–268.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016a. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. pages 6045–6049.

Yun-Nung Chen, Dilek Hakkani-Tür, Gokhan Tur, Jianfeng Gao, and Li Deng. 2016b. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *Interspeech*.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics* 31(1):25–70.

Paul A Crook, Jean-Philippe Martin, and Ruhi Sarikaya. 2015. Multi-language hypotheses ranking and domain tracking for open domain. In *Interspeech*.

Ali El-Kahky, Xiaohu Liu, Ruhi Sarikaya, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2014. Extending domain coverage of language understanding systems via intent transfer between domains using knowledge graphs and search query click logs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pages 4067–4071.

Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine 29(5):1189–1232.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*. pages 1019–1027.

Kaiming He, Xiang Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 770–778.

Chiori Hori, Takaaki Hori, Shinji Watanabe, and John R Hershey. 2016. Context-sensitive and role-dependent spoken language understanding using bidirectional and attention lstms. *Interspeech* pages 3236–3240.

Aaron Jaech, Larry Heck, and Mari Ostendorf. 2016. Domain adaptation of recurrent neural networks for natural language understanding. In *Interspeech*.

Omar Zia Khan, Jean-Philippe Robichaud, Paul A. Crook, and Ruhi Sarikaya. 2015. Hypotheses ranking and state tracking for a multi-domain dialog system using multiple ASR alternates. In *Interspeech*.

Young-Bum Kim, Minwoo Jeong, Karl Stratos, and Ruhi Sarikaya. 2015a. Weakly supervised slot tagging with partially labeled sequences from web search click logs. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 84–92.

Young-Bum Kim, Sungjin Lee, and Ruhi Sarikaya. 2017a. Speaker-sensitive dual memory networks for multi-turn slot tagging. In *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, pages 547–553.

Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017b. Onenet: Joint domain, intent, slot prediction for spoken language understanding. In *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, pages 547–553.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017c. Adversarial adaptation of synthetic or stale data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1297–1307.

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

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2015b. Pre-training of hidden-unit crfs. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. volume 2, pages 192–198.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Frustratingly easy neural domain adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 387–396.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2017e. A framework for pre-training hidden-unit conditional random fields and its extension to long short term memory networks. *Computer Speech & Language* 46:311–326.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015c. New transfer learning techniques for disparate label sets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. volume 1, pages 473–482.

Diederik Kingma and Jimmy Ba. 2015. ADAM: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems 30 (NIPS)*. pages 972–981.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1520–1530.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech*. pages 685–689.

Bing Liu and Ian Lane. 2017. Multi-domain adversarial learning for slot filling in spoken language understanding. In *NIPS Workshop on Conversational AI*.

F. Morbini, K. Audhkhasi, R. Artstein, M. Van Segbroeck, K. Sagae, P. Georgiou, D. R. Traum, and S. Narayanan. 2012. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *IEEE Spoken Language Technology Workshop (SLT)*. pages 49–54.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980* .

Truc-Vien T Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2010. Kernel-based reranking for named-entity extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, pages 901–909.

Brian Roark, Yang Liu, Mary Harper, Robin Stewart, Matthew Lease, Matthew Snover, Izhak Shafran, Bonnie Dorr, John Hale, Anna Krasnyanskaya, et al. 2006. Reranking for sentence boundary detection in conversational speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jean-Philippe Robichaud, Paul A. Crook, Puyang Xu, Omar Zia Khan, and Ruhi Sarikaya. 2014. Hypotheses ranking for robust domain classification and tracking in dialogue systems. In *Interspeech*. pages 145–149.

Ruhi Sarikaya. 2017. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine* 34(1):67–81.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

Gokhan Tur and Renato de Mori. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. New York, NY: John Wiley and Sons.

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

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *International Conference on Learning Representation (ICLR)* .

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *International Joint Conference on Artificial Intelligence (IJCAI)*. pages 2993–2999.