Joint Slot Filling and Intent Detection via Capsule Neural Networks

Chenwei Zhang[†], Yaliang Li[§], Nan Du[‡], Wei Fan[‡], Philip S. Yu[†¶]
[†]University of Illinois at Chicago, Chicago, IL 60607 USA
[§]Alibaba Group, Bellevue, WA 98004 USA
[‡]Tencent Medical AI Lab, Palo Alto, CA 94301 USA
[¶]Institute for Data Science, Tsinghua University, Beijing, China
{czhang99,psyu}@uic.edu, yaliang.li@alibaba-inc.com,
nandu2048@gmail.com, davidwfan@tencent.com

Abstract

Being able to recognize words as slots and detect the intent of an utterance has been a keen issue in natural language understanding. The existing works either treat slot filling and intent detection separately in a pipeline manner, or adopt joint models which sequentially label slots while summarizing the utterance-level intent without explicitly preserving the hierarchical relationship among words, slots, and intents. To exploit the semantic hierarchy for effective modeling, we propose a capsule-based neural network model which accomplishes slot filling and intent detection via a dynamic routing-by-agreement schema. A re-routing schema is proposed to further synergize the slot filling performance using the inferred intent representation. Experiments on two real-world datasets show the effectiveness of our model when compared with other alternative model architectures, as well as existing natural language understanding services.

1 Introduction

With the ever-increasing accuracy in speech recognition and complexity in user-generated utterances, it becomes a critical issue for mobile phones or smart speaker devices to understand the natural language in order to give informative responses. Slot filling and intent detection play important roles in Natural Language Understanding (NLU) systems. For example, given an utterance from the user, the slot filling annotates the utterance on a word-level, indicating the slot type mentioned by a certain word such as the slot `artist` mentioned by the word `Sungmin`, while the intent detection works on the utterance-level to give categorical intent label(s) to the whole utterance. Figure 1 illustrates this idea.

To deal with diversely expressed utterances without additional feature engineering, deep neural network based user intent detection models (Hu et al., 2009; Xu and Sarikaya, 2013; Zhang et al., 2016; Liu and Lane, 2016; Zhang et al., 2017; Chen et al., 2016; Xia et al., 2018) are proposed to classify user intents given their utterances in the natural language.

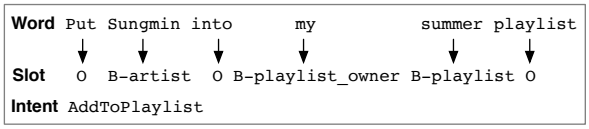

Figure 1: An example of an utterance with BOI format annotation for slot filling, which indicates the slot of artist, play list owner, and play list name from an utterance with an intent AddToPlaylist.

Currently, the slot filling is usually treated as a sequential labeling task. A neural network such as a recurrent neural network (RNN) or a convolution neural network (CNN) is used to learn context-aware word representations, along with sequence tagging methods such as conditional random field (CRF) (Lafferty et al., 2001) that infer the slot type for each word in the utterance.

Word-level slot filling and utterance-level intent detection can be conducted simultaneously to achieve a synergistic effect. The recognized slots, which possess word-level signals, may give clues to the utterance-level intent of an utterance. For example, with a word `Sungmin` being recognized as a slot `artist`, the utterance is more likely to have an intent of `AddToPlayList` than other intents such as `GetWeather` or `BookRestaurant`.

Some existing works learn to fill slots while detecting the intent of the utterance (Xu and Sarikaya, 2013; Hakkani-Tür et al., 2016; Liu and Lane, 2016; Goo et al., 2018): a convolution layer or a recurrent layer is adopted to sequentially label word with their slot types: the last hidden state of the recurrent neural network, or an attention-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5259–5267
Florence, Italy, July 28 - August 2, 2019.

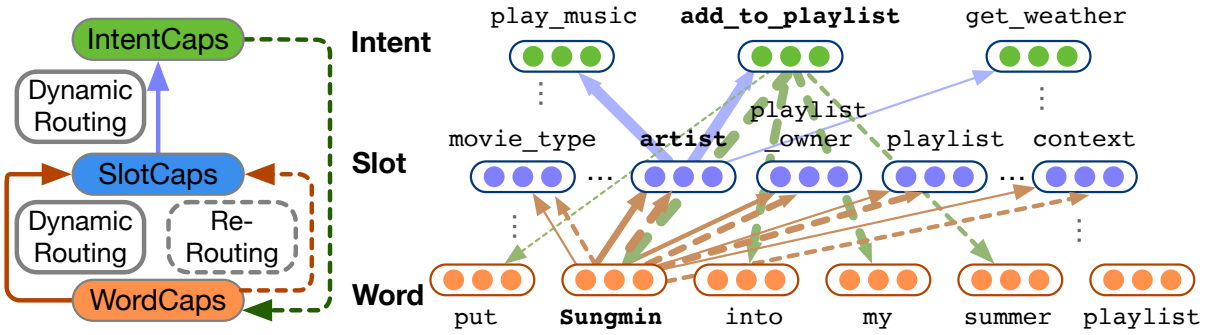

Figure 2: Illustration of the proposed CAPSULE-NLU model for joint slot filling and intent detection. The model does slot filling by learning to assign each word in the WordCaps to the most appropriate slot in SlotCaps via dynamic routing. The weights learned via dynamic routing indicate how strong each word in WordCaps belongs to a certain slot type in SlotCaps. The dynamic routing also learns slot representations using WordCaps and the learned weight. The learned slot representations in SlotCaps are further aggregated to predict the utterance-level intent of the utterance. Once the intent label of the utterance is determined, a novel re-routing process is proposed to help improve word-level slot filling by the inferred utterance-level intent label. The solid lines indicate the dynamic-routing process and dash lines indicate the re-routing process.

weighted sum of all convolution outputs are used to train an utterance-level classification module for intent detection. Such approaches achieve decent performances but do not explicitly consider the hierarchical relationship between words, slots, and intents: intents are sequentially summarized from the word sequence. As the sequence becomes longer, it is risky to simply rely on the gate function of RNN to compress all context information in a single vector (Cheng et al., 2016).

In this work, we make the very first attempt to bridge the gap between word-level slot modeling and the utterance-level intent modeling via a hierarchical capsule neural network structure (Hinton et al., 2011; Sabour et al., 2017). A capsule houses a vector representation of a group of neurons. The capsule model learns a hierarchy of feature detectors via a routing-by-agreement mechanism: capsules for detecting low-level features send their outputs to high-level capsules only when there is a strong agreement of their predictions to high-level capsules.

The aforementioned properties of capsule models are appealing for natural language understanding from a hierarchical perspective: words such as `Sungmin` are routed to concept-level slots such as `artist`, by learning how each word matches the slot representation. Concept-level slot features such as `artist`, `playlist owner`, and `playlist` collectively contribute to an utterance-level intent `AddToPlaylist`. The dynamic routing-by-agreement assigns a larger weight from a lower-level capsule to a higher-level when the low-level feature is more predictive to one high-level feature, than other high-level features. Figure 2 illustrates this idea.

The inferred utterance-level intent is also helpful in refining the slot filling result. For example, once an `AddToPlaylist` intent representation is learned in IntentCaps, the slot filling may capitalize on the inferred intent representation and recognize slots that are otherwise neglected previously. To achieve this, we propose a re-routing schema for capsule neural networks, which allows high-level features to be actively engaged in the dynamic routing between WordCaps and SlotCaps, which improves the slot filling performance.

To summarize, the contributions of this work are as follows:

- Encapsulating the hierarchical relationship among word, slot, and intent in an utterance by a hierarchical capsule neural network structure.
- Proposing a dynamic routing schema with re-routing that achieves synergistic effects for joint slot filling and intent detection.
- Showing the effectiveness of our model on two real-world datasets, and comparing with existing models as well as commercial NLU services.

2 Approach

We propose to model the hierarchical relationship among each word, the slot it belongs to, and

the intent label of the whole utterance by a hierarchical capsule neural network structure called CAPSULE-NLU. The proposed architecture consists of three types of capsules: 1) WordCaps that learn context-aware word representations, 2) SlotCaps that categorize words by their slot types via dynamic routing, and construct a representation for each type of slot by aggregating words that belong to the slot, 3) IntentCaps determine the intent label of the utterance based on the slot representation as well as the utterance contexts. Once the intent label has been determined by IntentCaps, the inferred utterance-level intent helps re-recognizing slots from the utterance by a re-routing schema.

2.1 WordCaps

Given an input utterance $x = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_T)$ of T words, where each word is initially represented by a vector of dimension D_W. Here we simply trained word represenations from scratch. Various neural network structures can be used to learn context-aware word representations. For example, a recurrent neural network such as a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) can be applied to learn representations of each word in the utterance:

$$\begin{aligned} \vec{\mathbf{h}}_t &= \text{LSTM}_{fw}(\mathbf{w}_t, \vec{\mathbf{h}}_{t-1}), \\ \overleftarrow{\mathbf{h}}_t &= \text{LSTM}_{bw}(\mathbf{w}_t, \overleftarrow{\mathbf{h}}_{t+1}). \end{aligned} \tag{1}$$

For each word $\mathbf{w}_t$, we concatenate each forward hidden state $\vec{\mathbf{h}}_t$ obtained from the forward LSTM_{fw} with a backward hidden state $\overleftarrow{\mathbf{h}}_t$ from LSTM_{bw} to obtain a hidden state $\mathbf{h}_t$. The whole hidden state matrix can be defined as $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T) \in \mathbb{R}^{T \times 2D_H}$, where D_H is the number of hidden units in each LSTM. In this work, the parameters of WordCaps are trained with the whole model, while sophisticated pre-trained models such as ELMo (Peters et al., 2018) or BERT (Devlin et al., 2018) may also be integrated.

2.2 SlotCaps

Traditionally, the learned hidden state $\mathbf{h}_t$ for each word $\mathbf{w}_t$ is used as the logit to predict its slot tag. When $\mathbf{H}$ for all words in the utterance is learned, sequential tagging methods like the linear-chain CRF models the tag dependencies by assigning a transition score for each transition pattern between adjacent tags to ensure the best tag sequence of the utterance from all possible tag sequences.

Instead of doing slot filling via sequential labeling which does not directly consider the dependencies among words, the SlotCaps learn to recognize slots via dynamic routing. The routing-by-agreement explicitly models the hierarchical relationship between capsules. For example, the routing-by-agreement mechanism send a low-level feature, e.g. a word representation in WordCaps, to high-level capsules, e.g. SlotCaps, only when the word representation has a strong agreement with a slot representation.

The agreement value on a word may vary when being recognized as different slots. For example, the word `three` may be recognized as a `party_size_number` slot or a `time` slot. The SlotCaps first convert the word representation obtained in WordCaps with respect to each slot type. We denote $\mathbf{p}_{k|t}$ as the resulting prediction vector of the t-th word when being recognized as the k-th slot:

$$\mathbf{p}_{k|t} = \sigma(\mathbf{W}_k \mathbf{h}_t^T + \mathbf{b}_k), \tag{2}$$

where $k \in \{1, 2, ..., K\}$ denotes the slot type and $t \in \{1, 2, ..., T\}$. σ is the activation function such as $tanh$. $\mathbf{W}_k \in \mathbb{R}^{D_P \times 2D_H}$ and $\mathbf{b}_k \in \mathbb{R}^{D_P \times 1}$ are the weight and bias matrix for the k-th capsule in SlotCaps, and D_P is the dimension of the prediction vector.

Slot Filling by Dynamic Routing-by-agreement We propose to determine the slot type for each word by dynamically route prediction vectors of each word from WordCaps to SlotCaps. The dynamic routing-by-agreement learns an agreement value c_{kt} that determines how likely the t-th word agrees to be routed to the k-th slot capsule. c_{kt} is calculated by the dynamic routing-by-agreement algorithm (Sabour et al., 2017), which is briefly recalled in Algorithm 1.

Algorithm 1 Dynamic routing-by-agreement

```
procedure DYNAMIC ROUTING(p_{k|t}, iter)
    for each WordCaps t and SlotCaps k: b_{kt} ← 0.
    for iter iterations do
        for all WordCaps t: c_t ← softmax(b_t)
        for all SlotCaps k: s_k ← Σ_r c_{kt} p_{k|t}
        for all SlotCaps k: v_k = squash(s_k)
        for all WordCaps t and SlotCaps k: b_{kt} ←
    b_{kt} + p_{k|t} · v_k
    end for
    Return v_k
end procedure
```

The above algorithm determines the agreement

value c_{kt} between WordCaps and SlotCaps while learning the slot representations $\mathbf{v}_k$ in an unsupervised, iterative fashion. $\mathbf{c}_t$ is a vector that consists of all c_{kt} where $k \in K$. b_{kt} is the logit (initialized as zero) representing the log prior probability that the t-th word in WordCaps agrees to be routed to the k-th slot capsule in SlotCaps (Line 2). During each iteration (Line 3), each slot representation $\mathbf{v}_k$ is calculated by aggregating all the prediction vectors for that slot type $\{\mathbf{p}_{k|t}|t \in T\}$, weighted by the agreement values c_{kt} obtained from b_{kt} (Line 5-6):

$$\mathbf{s}_k = \sum_t^T c_{kt}\mathbf{p}_{k|t}, \tag{3}$$

$$\mathbf{v}_k = \text{squash}(\mathbf{s}_k) = \frac{\|\mathbf{s}_k\|^2}{1+\|\mathbf{s}_k\|^2}\frac{\mathbf{s}_k}{\|\mathbf{s}_k\|}, \tag{4}$$

where a squashing function $\text{squash}(\cdot)$ is applied on the weighted sum $\mathbf{s}_k$ to get $\mathbf{v}_k$ for each slot type. Once we updated the slot representation $\mathbf{v}_k$ in the current iteration, the logit b_{kt} becomes larger when the dot product $\mathbf{p}_{k|t} \cdot \mathbf{v}_k$ is large. That is, when a prediction vector $\mathbf{p}_{k|t}$ is more similar to a slot representation $\mathbf{v}_k$, the dot product is larger, indicating that it is more likely to route this word to the k-th slot type (Line 7). An updated, larger b_{kt} will lead to a larger agreement value c_{kt} between the t-th word and the k-th slot in the next iteration. On the other hand, it assigns low c_{kt} when there is inconsistency between $p_{k|t}$ and $\mathbf{v}_k$. The agreement values learned via the unsupervised, iterative algorithm ensures the outputs of the WordCaps get sent to appropriate subsequent SlotCaps after $iter_{\text{slot}}$ iterations.

Cross Entropy Loss for Slot Filling

For the t-th word in an utterance, its slot type is determined as follows:

$$\hat{y}_t = \arg\max_{k \in K}(c_{kt}). \tag{5}$$

The slot filling loss is defined over the utterance as the following cross-entropy function:

$$\mathcal{L}_{slot} = -\sum_t \sum_k y_t^k \log(\hat{y}_t^k), \tag{6}$$

where y_t^k indicates the ground truth slot type for the t-th word. $y_t^k = 1$ when the t-th word belongs to the k-th slot type.

2.3 IntentCaps

The IntentCaps take the output $\mathbf{v}_k$ for each slot $k \in \{1, 2, ..., K\}$ in SlotCaps as the input, and determine the utterance-level intent of the whole utterance. The IntentCaps also convert each slot representation in SlotCaps with respect to the intent type:

$$\mathbf{q}_{l|k} = \sigma(\mathbf{W}_l\mathbf{v}_k^T + b_l), \tag{7}$$

where $l \in \{1, 2, ..., L\}$ and L is the number of intents. $\mathbf{W}_l \in \mathbb{R}^{D_L \times D_P}$ and $\mathbf{b}_l \in \mathbb{R}^{D_L \times 1}$ are the weight and bias matrix for the l-th capsule in IntentCaps.

IntentCaps adopt the same dynamic routing-by-agreement algorithm, where:

$$\mathbf{u}_l = \text{DYNAMIC ROUTING}(\mathbf{q}_{l|k}, iter_{\text{intent}}). \tag{8}$$

Max-margin Loss for Intent Detection

Based on the capsule theory, the orientation of the activation vector $\mathbf{u}_l$ represents intent properties while its length indicates the activation probability. The loss function considers a max-margin loss on each labeled utterance:

$$\begin{aligned}\mathcal{L}_{intent} = \sum_{l=1}^{L}\{&[\![z = z_l]\!] \cdot \max(0, m^+ - \|\mathbf{u}_l\|)^2 \\ &+ \lambda\,[\![z \neq z_l]\!] \cdot \max(0, \|\mathbf{u}_l\| - m^-)^2\},\end{aligned} \tag{9}$$

where $\|\mathbf{u}_l\|$ is the norm of $\mathbf{u}_l$ and $[\![]\!]$ is an indicator function, z is the ground truth intent label for the utterance x. λ is the weighting coefficient, and m^+ and m^- are margins.

The intent of the utterance can be easily determined by choosing the activation vector with the largest norm $\hat{z} = \arg\max_{l \in \{1,2,...,L\}} \|\mathbf{u}_l\|$.

2.4 Re-Routing

The IntentCaps not only determine the intent of the utterance by the length of the activation vector, but also learn discriminative intent representations of the utterance by the orientations of the activation vectors. Previously, the dynamic routing-by-agreement shows how low-level features such as slots help construct high-level ideas such as intents. While the high-level features also work as a guide that helps learn low-level features. For example, the `AddToPlaylist` intent activation vector in IntentCaps also helps strength the existing slots such as `artist_name` during slot filling on the words `Sungmin` in SlotCaps.

Thus we propose a re-routing schema for SlotCaps where the dynamic routing-by-agreement is realized by the following equation that replaces the Line 7 in Algorithm 1:

$$\mathrm{b}_{kt} \leftarrow \mathrm{b}_{kt} + \mathbf{p}_{k|t} \cdot \mathbf{v}_k + \alpha \cdot \mathbf{p}_{k|t}^T \mathbf{W}_{RR} \hat{\mathbf{u}}_{\hat{z}}^T, \quad (10)$$

where $\hat{\mathbf{u}}_{\hat{z}}$ is the intent activation vector with the largest norm. $\mathbf{W}_{RR} \in \mathbb{R}^{D_P \times D_L}$ is a bi-linear weight matrix, and α as the coefficient. The routing information for each word is updated toward the direction where the prediction vector not only coincides with representative slots, but also towards the most-likely intent of the utterance. As a result, the re-routing makes SlotCaps obtain updated routing information as well as updated slot representations.

3 Experiment Setup

To demonstrate the effectiveness of our proposed models, we compare the proposed model CAPSULE-NLU with existing alternatives, as well as commercial natural language understanding services.

Datasets For each task, we evaluate our proposed models by applying it on two real-word datasets: SNIPS Natural Language Understanding benchmark[1] (SNIPS-NLU) and the Airline Travel Information Systems (ATIS) dataset (Tur et al., 2010). The statistical information on two datasets are shown in Table 1.

Dataset	SNIPS-NLU	ATIS
Vocab Size	11,241	722
Average Sentence Length	9.05	11.28
#Intents	7	21
#Slots	72	120
#Training Samples	13,084	4,478
#Validation Samples	700	500
#Test Samples	700	893

Table 1: Dataset statistics.

SNIPS-NLU contains natural language corpus collected in a crowdsourced fashion to benchmark the performance of voice assistants. ATIS is a widely used dataset in spoken language understanding, where audio recordings of people making flight reservations are collected.

Baselines We compare the proposed capsule-based model CAPSULE-NLU with other alternatives: 1) CNN TriCRF (Xu and Sarikaya, 2013) introduces a Convolution Neural Network (CNN) based sequential labeling model for slot filling. The hidden states for each word are summed up to predict the utterance intent. We adopt the performance with lexical features. 2) Joint Seq. (Hakkani-Tür et al., 2016) adopts a Recurrent Neural Network (RNN) for slot filling and the last hidden state of the RNN is used to predict the utterance intent. 3) Attention BiRNN (Liu and Lane, 2016) further introduces a RNN based encoder-decoder model for joint slot filling and intent detection. An attention weighted sum of all encoded hidden states is used to predict the utterance intent. 4) Slot-gated Full Atten. (Goo et al., 2018) utilizes a slot-gated mechanism as a special gate function in Long Short-term Memory Network (LSTM) to improve slot filling by the learned intent context vector. The intent context vector is used for intent detection. 5) DR-AGG (Gong et al., 2018) aggregates word-level information for text classification via dynamic routing. The high-level capsules after routing are concatenated, followed by a multi-layer perceptron layer that predicts the utterance label. We used this capsule-based text classification model for intent detection only. 6) IntentCapsNet (Xia et al., 2018) adopts a multi-head self-attention to extract intermediate semantic features from the utterances, and uses dynamic routing to aggregate semantic features into intent representations for intent detection. We use this capsule-based model for intent detection only.

We also compare our proposed model CAPSULE-NLU with existing commercial natural language understanding services, including api.ai (Now called DialogFlow)[2], Waston Assistant[3], Luis[4], wit.ai[5], snips.ai[6], recast.ai[7], and Amazon Lex[8].

Implementation Details The hyperparameters used for experiments are shown in Table 2.

Dataset	D_W	D_H	D_P	D_L	$iter_{\text{slot}}$	$iter_{\text{intent}}$
SNIPS-NLU	1024	512	512	128	2	2
ATIS	1024	512	512	256	3	3

Table 2: Hyperparameter settings.

[1] https://github.com/snipsco/nlu-benchmark/

[2] https://dialogflow.com/

[3] https://www.ibm.com/cloud/watson-assistant/

[4] https://www.luis.ai/

[5] https://wit.ai/

[6] https://snips.ai/

[7] https://recast.ai/

[8] https://aws.amazon.com/lex/

Model	SNIPS-NLU			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
CNN TriCRF (Xu and Sarikaya, 2013)	-	-	-	0.944	-	-
Joint Seq. (Hakkani-Tür et al., 2016)	0.873	0.969	0.732	0.942	0.926	0.807
Attention BiRNN (Liu and Lane, 2016)	0.878	0.967	0.741	0.942	0.911	0.789
Slot-Gated Full Atten. (Goo et al., 2018)	0.888	0.970	0.755	0.948	0.936	0.822
DR-AGG (Gong et al., 2018)	-	0.966	-	-	0.914	-
IntentCapsNet (Xia et al., 2018)	-	0.974	-	-	0.948	-
CAPSULE-NLU	**0.918**	0.973	**0.809**	**0.952**	**0.950**	**0.834**
CAPSULE-NLU w/o Intent Detection	0.902	-	-	0.948	-	-
CAPSULE-NLU w/o Joint Training	0.902	**0.977**	0.804	0.948	0.847	0.743

Table 3: Slot filling and intention detection results using CAPSULE-NLU on two datasets.

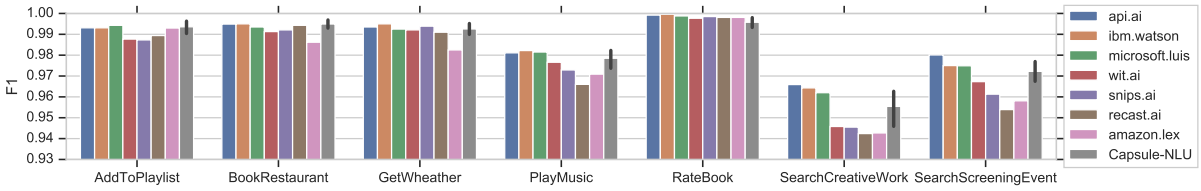

Figure 3: Stratified 5-fold cross validation for benchmarking with existing NLU services on SNIPS-NLU dataset. Black bars indicate the standard deviation.

We use the validation data to choose hyperparameters. For both datasets, we randomly initialize word embeddings using Xavier initializer and let them train with the model. In the loss function, the down-weighting coefficient λ is 0.5, margins m^+ and m^- are set to 0.8 and 0.2 for all the existing intents. α is set as 0.1. RMSProp optimizer (Tieleman and Hinton, 2012) is used to minimize the loss. To alleviate over-fitting, we add the dropout to the LSTM layer with a dropout rate of 0.2.

4 Results

Quantitative Evaluation The intent detection results on two datasets are reported in Table 3, where the proposed capsule-based model performs consistently better than current learning schemes for joint slot filling and intent detection, as well as capsule-based neural network models that only focuses on intent detection. These results demonstrate the novelty of the proposed capsule-based model CAPSULE-NLU in jointly modeling the hierarchical relationships among words, slots and intents via the dynamic routing between capsules.

Also, we benchmark the intent detection performance of the proposed model with existing natural language understanding services[9] in Figure 3. Since the original data split is not available, we report the results with stratified 5-fold cross validation. From Figure 3 we can see that the proposed model CAPSULE-NLU is highly competitive with off-the-shelf systems that are available to use. Note that, our model archieves the performance without using pre-trained word representations: the word embeddings are simply trained from scratch.

Ablation Study To investigate the effectiveness of CAPSULE-NLU in joint slot filling and intent detection, we also report ablation test results in Table 3. "w/o Intent Detection" is the model without intent detection: only a dynamic routing is performed between WordCaps and SlotCaps for the slot filling task, where we minimize $\mathcal{L}_{slot}$ during training; "w/o Joint Training" adopts a two-stage training where the model is first trained for slot filling by minimizing $\mathcal{L}_{slot}$, and then use the fixed slot representations to train for the intent detection task which minimizes $\mathcal{L}_{intent}$. From the lower part of Table 3 we can see that by using a capsule-based hierarchical modeling between words and slots, the model CAPSULE-NLU w/o Intent Detection is already able to outperform current alternatives on slot filling that adopt a sequential labeling schema. The joint training of slot filling and intent detection is able to give each subtask further improvements when the model parameters are updated jointly.

[9] https://www.slideshare.net/KonstantinSavenkov/nlu-intent-detection-benchmark-by-intento-august-2017

Visualizing Agreement Values between Capsule Layers Thanks to the dynamic routing-by-agreement schema, the dynamically learned agreement values between different capsule layers naturally reflect how low-level features are collectively aggregated into high-level ones for each input utterance. In this section, we harness the intepretability of the proposed capsule-based model via hierarchical modeling and provide case studies and visualizations.

Between WordCaps and SlotCaps First we study the agreement value c_{kt} between the t-th word in the WordCaps and the k-th slot capsule in SlotCaps. As shown in Figure 4, we observe that the dynamic routing-by-agreement is able to converge to an agreement quickly after the first iteration (shown in blue bars). It is able to assign a confident probability assignment close to 0 or 1. After the second iteration (shown in orange bars), the model is more certain about the routing decisions: probabilities are more leaning towards 0 or 1 as the model is confident about routing a word in WordCaps to its most appropriate slot in SlotCaps.

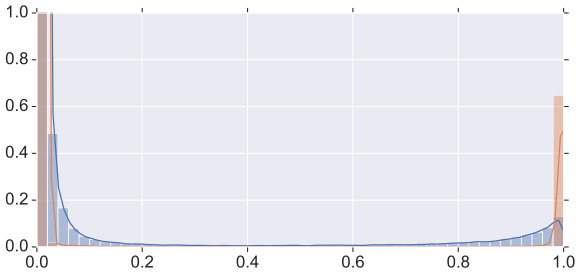

Figure 4: The distribution of all agreement values between WordCaps and SlotCaps on the test split of SNIPS-NLU dataset. Blue: the distribution of values after the first iteration. Yellow: the distribution after the second iteration.

However, we do find that when unseen slot values like new object names emerge in utterances like `show me the movie operetta for the theatre organ` with an intent of `SearchCreativeWork`, the iterative dynamic routing process would be even more appealing. Figure 5 shows the agreement values learned by dynamic routing-by-agreement. Since the dynamic routing-by-agreement is an iterative process controlled by the variable $iter_{slot}$, we show the agreement values after the first iteration in the left part of Figure 5, and the values after the second iteration in the right part.

From the left part of Figure 5, we can see that after the first iteration, the model considers the word `operetta` itself alone is likely to be an object name, probably because the following word `for` is usually a context word being annotated as `O`. Thus it tends to route word `for` to both the slot `O` and the slot `I-object_name`. However, from the right part of Figure 5 we can see that after the second iteration, the dynamic routing found an agreement and is more certain to have `operetta for the theatre organ` as a whole for the slot `B-object_name` and `I-object_name`.

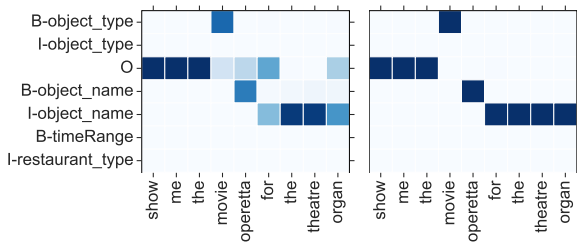

Figure 5: The learned agreement values between WordCaps (x-axis) and SlotCaps (y-axis). A sample from the test split of SNIPS-NLU dataset is shown (Left: after the fist routing iteration. Right: after the second iteration). Due to space limitations, only part of slots (7/72) are shown on the y-axis.

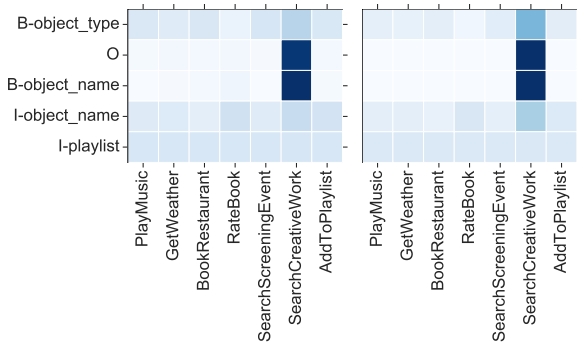

Figure 6: The learned agreement values between SlotCaps (y-axis) and IntentCaps (x-axis). Left: after the first iteration. Right: after the second iteration. The same sample utterance used in Figure 5 is used here.

Between SlotCaps and IntentCaps Similarly, we visualize the agreement values between each slot capsule in SlotCaps and each intent capsule in IntentCaps. The left part of Figure 6 shows that after the first iteration, since the model is not able to correctly recognize `operetta for the theatre organ` as a whole, only the context slot `O` (correspond to the word `show me the`) and `B-object_name` (correspond to the word `operetta`) contribute significantly to the final intent capsule. From the right part of Figure 6, we found that with the word `operetta for the theatre organ` being recognized in the lower capsule, the slots `I-object_name` and `B-object_type` contribute more to the correct intent capsule `SearchCreativeWork`, when comparing with other routing alternatives to other intent capsules.

5 Related Works

Intent Detection With recent developments in deep neural networks, user intent detection models (Hu et al., 2009; Xu and Sarikaya, 2013; Zhang et al., 2016; Liu and Lane, 2016; Zhang et al., 2017; Chen et al., 2016; Xia et al., 2018) are proposed to classify user intents given their diversely expressed utterances in the natural language. As a text classification task, the decent performance on utterance-level intent detection usually relies on hidden representations that are learned in the intermediate layers via multiple non-linear transformations.

Recently, various capsule based text classification models are proposed that aggregate word-level features for utterance-level classification via dynamic routing-by-agreement (Gong et al., 2018; Zhao et al., 2018; Xia et al., 2018). Among them, Xia et al. (2018) adopts self-attention to extract intermediate semantic features and uses a capsule-based neural network for intent detection. However, existing works do not study word-level supervisions for the slot filling task. In this work, we explicitly model the hierarchical relationship between words and slots on the word-level, as well as intents on the utterance-level via dynamic routing-by-agreement.

Slot Filling Slot filling annotates the utterance with finer granularity: it associates certain parts of the utterance, usually named entities, with predefined slot tags. Currently, the slot filling is usually treated as a sequential labeling task. A recurrent neural network such as Gated Recurrent Unit (GRU) or Long Short-term Memory Network (LSTM) is used to learn context-aware word representations, and Conditional Random Fields (CRF) are used to annotate each word based on its slot type. Recently, Shen et al. (2017); Tan et al. (2017) introduce the self-attention mechanism for CRF-free sequential labeling.

Joint Modeling via Sequence Labeling To overcome the error propagation in the word-level slot filling task and the utterance-level intent detection task in a pipeline, joint models are proposed to solve two tasks simultaneously in a unified framework. Xu and Sarikaya (2013) propose a Convolution Neural Network (CNN) based sequential labeling model for slot filling. The hidden states corresponding to each word are summed up in a classification module to predict the utterance intent. A Conditional Random Field module ensures the best slot tag sequence of the utterance from all possible tag sequences. Hakkani-Tür et al. (2016) adopt a Recurrent Neural Network (RNN) for slot filling and the last hidden state of the RNN is used to predict the utterance intent. Liu and Lane (2016) further introduce an RNN based encoder-decoder model for joint slot filling and intent detection. An attention weighted sum of all encoded hidden states is used to predict the utterance intent. Some specific mechanisms are designed for RNNs to explicitly encode the slot from the utterance. For example, Goo et al. (2018) utilize a slot-gated mechanism as a special gate function in Long Short-term Memory Network (LSTM) to improve slot filling by the learned intent context vector. However, as the sequence becomes longer, it is risky to simply rely on the gate function to sequentially summarize and compress all slots and context information in a single vector (Cheng et al., 2016).

In this paper, we harness the capsule neural network to learn a hierarchy of feature detectors and explicitly model the hierarchical relationships among word-level slots and utterance-level intent. Also, instead of doing sequence labeling for slot filling, we use a dynamic routing-by-agreement schema between capsule layers to route each word in the utterance to its most appropriate slot type. And we further route slot representations, which are learned dynamically from words, to the most appropriate intent capsule for intent detection.

6 Conclusions

In this paper, a capsule-based model, namely CAPSULE-NLU, is introduced to harness the hierarchical relationships among words, slots, and intents in the utterance for joint slot filling and intent detection. Unlike treating slot filling as a sequential prediction problem, the proposed model assigns each word to its most appropriate slots in SlotCaps by a dynamic routing-by-agreement schema. The learned word-level slot representations are futher aggregated to get the utterance-level intent representations via dynamic routing-by-agreement. A re-routing schema is proposed to further synergize the slot filling performance using the inferred intent representation. Experiments on two real-world datasets show the effectiveness of the proposed models when compared with other alternatives as well as existing NLU services.

7 Acknowledgments

We thank the reviewers for their valuable comments. This work is supported in part by NSF through grants IIS-1526499, IIS-1763325, and CNS-1626432.

References

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.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuanjing Huang. 2018. Information aggregation via dynamic routing for sequence encoding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2742–2752.

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)*, volume 2, pages 753–757.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *WWW*, pages 471–480. ACM.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, ICML 2001, pages 282–289.

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

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NIPS*, pages 3859–3869.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding. *arXiv preprint arXiv:1709.04696*.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2017. Deep semantic role labeling with self-attention. *arXiv preprint arXiv:1712.01586*.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *Spoken Language Technology Workshop (SLT), 2010 IEEE*, pages 19–24. IEEE.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099.

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

Chenwei Zhang, Nan Du, Wei Fan, Yaliang Li, Chun-Ta Lu, and Philip S Yu. 2017. Bringing semantic structures to user intent detection in online medical queries. In *IEEE Big Data*, pages 1019–1026.

Chenwei Zhang, Wei Fan, Nan Du, and Philip S Yu. 2016. Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In *WWW*, pages 1373–1384.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.