CM-Net: A Novel Collaborative Memory Network for Spoken Language Understanding

Yijin Liu[1]*, Fandong Meng[2], Jinchao Zhang[2], Jie Zhou[2], Yufeng Chen[1] and Jinan Xu[1]†
[1]Beijing Jiaotong University, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
adaxry@gmail.com
{fandongmeng, dayerzhang, withtomzhou}@tencent.com
{chenyf, jaxu}@bjtu.edu.cn

Abstract

Spoken Language Understanding (SLU) mainly involves two tasks, intent detection and slot filling, which are generally modeled jointly in existing works. However, most existing models fail to fully utilize co-occurrence relations between slots and intents, which restricts their potential performance. To address this issue, in this paper we propose a novel **C**ollaborative **M**emory **N**etwork (CM-Net) based on the well-designed block, named CM-block. The CM-block firstly captures slot-specific and intent-specific features from memories in a collaborative manner, and then uses these enriched features to enhance local context representations, based on which the sequential information flow leads to more specific (slot and intent) global utterance representations. Through stacking multiple CM-blocks, our CM-Net is able to alternately perform information exchange among specific memories, local contexts and the global utterance, and thus incrementally enriches each other. We evaluate the CM-Net on two standard benchmarks (ATIS and SNIPS) and a self-collected corpus (CAIS). Experimental results show that the CM-Net achieves the state-of-the-art results on the ATIS and SNIPS in most of criteria, and significantly outperforms the baseline models on the CAIS. Additionally, we make the CAIS dataset publicly available for the research community [1].

1 Introduction

Spoken Language Understanding (SLU) is a core component in dialogue systems. It typically aims to identify the intent and semantic constituents for a given utterance, which are referred as intent detection and slot filling, respectively. Past years have witnessed rapid developments in diverse deep learning models (Haffner et al., 2003; Sarikaya et al., 2011) for SLU. To take full advantage of supervised signals of slots and intents, and share knowledge between them, most of existing works apply joint models that mainly based on CNNs (Xu and Sarikaya, 2013; Gupta et al., 2019), RNNs (Guo et al., 2014a; Liu and Lane, 2016), and asynchronous bi-model (Wang et al., 2018). Generally, these joint models encode words convolutionally or sequentially, and then aggregate hidden states into a utterance-level representation for the intent prediction, without interactions between representations of slots and intents.

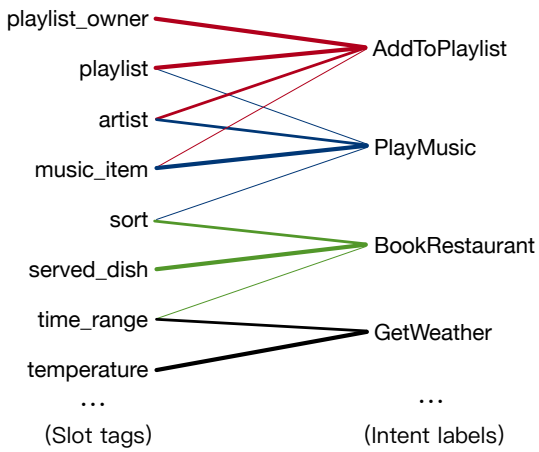

Figure 1: Statistical association of slot tags (on the left) and intent labels (on the right) in the SNIPS, where colors indicate different intents and thicknesses of lines indicate proportions.

Intuitively, slots and intents from similar fields tend to occur simultaneously, which can be observed from Figure 1 and Table 1. Therefore, it is beneficial to generate the representations of slots and intents with the guidance from each other. Some works explore enhancing the slot filling task

* This work was done when Yijin Liu was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China
† Jinan Xu is the corresponding author of the paper.
[1]Code is available at: https://github.com/Adaxry/CM-Net.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1051–1060, Hong Kong, China, November 3–7, 2019.

#	Utterance	Slot tag	Intent
1	play Roy Orbison tunes now	artist	PlayMusic
2	add this Roy Orbison song onto Women of Comedy	artist	AddToPlaylist
3	book a spot for seven at a bar with chicken french	served_dish	BookRestaurant
4	book french food for me and angeline at a restaurant	cuisine	BookRestaurant

Table 1: Examples in SNIPS with annotations of intent label for the utterance and slot tags for partial words.

unidirectionally with the guidance from intent representations via gating mechanisms (Goo et al., 2018; Li et al., 2018), while the predictions of intents lack the guidance from slots. Moreover, the capsule network with dynamic routing algorithms (Zhang et al., 2018a) is proposed to perform interactions in both directions. However, there are still two limitations in this model. The one is that the information flows from words to slots, slots to intents and intents to words in a pipeline manner, which is to some extent limited in capturing complicated correlations among words, slots and intents. The other is that the local context information which has been shown highly useful for the slot filling (Mesnil et al., 2014), is not explicitly modeled.

In this paper, we try to address these issues, and thus propose a novel **C**ollaborative **M**emory **N**etwork, named CM-Net. The main idea is to directly capture semantic relationships among words, slots and intents, which is conducted simultaneously at each word position in a collaborative manner. Specifically, we alternately perform information exchange among the task-specific features referred from memories, local context representations and global sequential information via the well-designed block, named CM-block, which consists of three computational components:

- **Deliberate Attention**: Obtaining slot-specific and intent-specific representations from memories in a collaborative manner.
- **Local Calculation**: Updating local context representations with the guidances of the referred slot and intent representations in the previous *Deliberate Attention*.
- **Global Recurrence**: Generating specific (slot and intent) global sequential representations based on local context representations from the previous *Local Calculation*.

Above components in each CM-block are conducted consecutively, which are responsible for encoding information from different perspectives. Finally, multiple CM-blocks are stacked together, and construct our CM-Net.

We firstly conduct experiments on two popular benchmarks, SNIPS (Coucke et al., 2018) and ATIS (Hemphill et al., 1990; Tur et al., 2010). Experimental results show that the CM-Net achieves the state-of-the-art results in 3 of 4 criteria (*e.g.,* intent detection accuracy on ATIS) on both benchmarks. Additionally, trials on our self-collected dataset, named CAIS, demonstrate the effectiveness and generalizability of the CM-Net.

Our main contributions are as follows:

- We propose a novel CM-Net for SLU, which explicitly captures semantic correlations among words, slots and intents in a collaborative manner, and incrementally enriches the specific features, local context representations and global sequential representations through stacked CM-blocks.
- Our CM-Net achieves the state-of-the-art results on two major SLU benchmarks (ATIS and SNIPS) in most of criteria.
- We contribute a new corpus CAIS with manual annotations of slot tags and intent labels to the research community.

2 Background

In principle, the slot filling is treated as a sequence labeling task, and the intent detection is a classification problem. Formally, given an utterance $X = \{x_1, x_2, \cdots, x_N\}$ with N words and its corresponding slot tags $Y^{slot} = \{y_1, y_2, \cdots, y_N\}$, the slot filling task aims to learn a parameterized mapping function $f_\theta : X \rightarrow Y$ from input words to slot tags. For the intent detection, it is designed to predict the intent label $\hat{y}^{int}$ for the entire utterance X from the predefined label set S^{int}.

Typically, the input utterance is firstly encoded into a sequence of distributed representations $\mathbf{X} =$

$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ by character-aware and pre-trained word embeddings. Afterwards, the following bidirectional RNNs are applied to encode the embeddings $\mathbf{X}$ into context-sensitive representations $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_N\}$. An external CRF (Lafferty et al., 2001) layer is widely utilized to calculate conditional probabilities of slot tags:

$$p(\mathbf{y}^{slot}|\mathbf{H}) = \frac{e^{F(\mathbf{H}, \mathbf{y}^{slot})}}{\sum_{\widetilde{\mathbf{y}}^{slot} \in \mathbf{Y}_x} e^{F(\mathbf{H}, \widetilde{\mathbf{y}}^{slot})}} \quad (1)$$

Here $\mathbf{Y}_x$ is the set of all possible sequences of tags, and $F(\cdot)$ is the score function calculated by:

$$F(\mathbf{h}, \mathbf{y}) = \sum_{i=1}^{N} \mathbf{A}_{y_i, y_{i+1}} + \sum_{i=1}^{N} \mathbf{P}_{i, y_i} \quad (2)$$

where $\mathbf{A}$ is the transition matrix that $\mathbf{A}_{i,j}$ indicates the score of a transition from i to j, and $\mathbf{P}$ is the score matrix output by RNNs. $P_{i,j}$ indicates the score of the j^{th} tag of the i^{th} word in a sentence (Lample et al., 2016).

When testing, the Viterbi algorithm (Forney, 1973) is used to search the sequence of slot tags with maximum score:

$$\hat{\mathbf{y}}^{slot} = \arg\max_{\widetilde{\mathbf{y}}^{slot} \in \mathbf{Y}_x} F(\mathbf{H}, \widetilde{\mathbf{y}}^{slot}) \quad (3)$$

As to the prediction of intent, the word-level hidden states $\mathbf{H}$ are firstly summarized into a utterance-level representation $\mathbf{v}^{int}$ via mean pooling (or max pooling or self-attention, *etc.*):

$$\mathbf{v}^{int} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{h}_t \quad (4)$$

The most probable intent label $\hat{y}^{int}$ is predicted by softmax normalization over the intent label set:

$$\begin{aligned} \hat{y}^{int} &= \arg\max_{\widetilde{y} \in S^{int}} P(\widetilde{y}|\mathbf{v}^{int}) \\ P(\widetilde{y} = j|\mathbf{v}^{int}) &= softmax(\mathbf{v}^{int})[j] \end{aligned} \quad (5)$$

Generally, both tasks are trained jointly to minimize the sum of cross entropy from each individual task. Formally, the loss function of the join model is computed as follows:

$$\begin{aligned} L &= (1-\lambda) \cdot L^{slot} + \lambda \cdot L^{int} \\ L^{int} &= -\sum_{i=1}^{|S^{int}|} \hat{y}_i^{int} log(y_i^{int}) \\ L^{slot} &= -\sum_{j=1}^{N} \sum_{i=1}^{|S^{slot}|} \hat{y}_{i,j}^{slot} log(y_{i,j}^{slot}) \end{aligned} \quad (6)$$

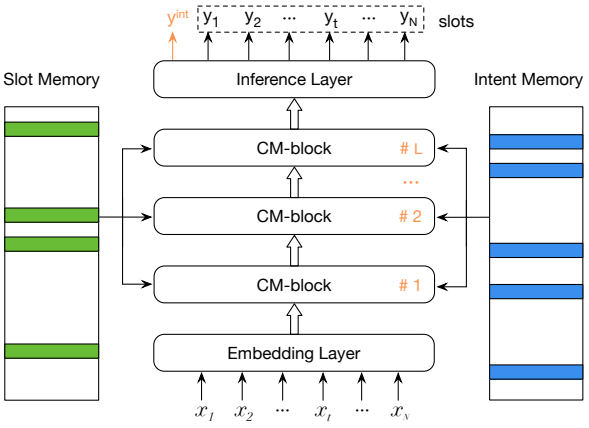

Figure 2: Overview of our proposed CM-Net. The input utterance is firstly encoded with the Embedding Layer (bottom), and then is transformed by multiple CM-blocks with the assistance of both slot and intent memories (on both sides). Finally we make predictions of slots and the intent in the Inference Layer (top).

where y_i^{int} and $y_{i,j}^{slot}$ are golden labels, and λ is hyperparameter, and $|S^{int}|$ is the size of intent label set, and similarly for $|S^{slot}|$.

3 CM-Net

3.1 Overview

In this section, we start with a brief overview of our CM-Net and then proceed to introduce each module. As shown in Figure 2, the input utterance is firstly encoded with the Embedding Layer, and then is transformed by multiple CM-blocks with the assistance of slot and intent memories, and finally make predictions in the Inference Layer.

3.2 Embedding Layers

Pre-trained Word Embedding The pre-trained word embeddings has been indicated as a de-facto standard of neural network architectures for various NLP tasks. We adapt the cased, 300d Glove[2] (Pennington et al., 2014) to initialize word embeddings, and keep them frozen.

Character-aware Word Embedding It has been demonstrated that character level information (*e.g.* capitalization and prefix) (Collobert et al., 2011) is crucial for sequence labeling. We use one layer of CNN followed by max pooling to generate character-aware word embeddings.

3.3 CM-block

The CM-block is the core module of our CM-Net, which is designed with three computational com-

[2]https://nlp.stanford.edu/projects/glove/

ponents: *Deliberate Attention*, *Local Calculation* and *Global Recurrence* respectively.

Deliberate Attention

To fully model semantic relations between slots and intents, we build the slot memory $\mathbf{M^{slot}}$ and intent memory $\mathbf{M^{int}}$, and further devise a collaborative retrieval approach. For the slot memory, it keeps $|S^{slot}|$ slot cells which are randomly initialized and updated as model parameters. Similarly for the intent memory. At each word position, we take the hidden state $\mathbf{h}_t$ as query, and obtain slot feature $\mathbf{h}_t^{slot}$ and intent feature $\mathbf{h}_t^{int}$ from both memories by the deliberate attention mechanism, which will be illustrated in the following.

Specifically for the slot feature $\mathbf{h}_t^{slot}$, we firstly get a rough intent representation $\widetilde{\mathbf{h}}_t^{int}$ by the word-aware attention with hidden state $\mathbf{h}_t$ over the intent memory $\mathbf{M^{int}}$, and then obtain the final slot feature $\mathbf{h}_t^{slot}$ by the intent-aware attention over the slot memory $\mathbf{M^{slot}}$ with the intent-enhanced representation $[\mathbf{h}_t; \widetilde{\mathbf{h}}_t^{int}]$. Formally, the above-mentioned procedures are computed as follows:

$$\begin{aligned} \widetilde{\mathbf{h}}_t^{int} &= ATT(\mathbf{h_t}, \mathbf{M^{int}}) \\ \mathbf{h}_t^{slot} &= ATT([\mathbf{h_t}; \widetilde{\mathbf{h}}_t^{int}], \mathbf{M^{slot}}) \end{aligned} \tag{7}$$

where $ATT(\cdot)$ is the query function calculated by the weighted sum of all cells $\mathbf{m}_i^x$ in memory $\mathbf{M}^x$ $(x \in \{slot, int\})$:

$$\begin{aligned} ATT(\mathbf{h_t}, \mathbf{M}^x) &= \sum_i \alpha_i \mathbf{m_i^x} \\ \alpha_i &= \frac{\exp(\mathbf{u}^\top s_i)}{\sum_j \exp(\mathbf{u}^\top s_j)} \\ s_i &= \mathbf{h}_t^\top \mathbf{W} \mathbf{m_i^x} \end{aligned} \tag{8}$$

Here $\mathbf{u}$ and $\mathbf{W}$ are model parameters. We name the above calculations of two-round attentions (Equation 7) as "deliberate attention".

The intent representation $\mathbf{h}_t^{int}$ is computed by the deliberate attention as well:

$$\begin{aligned} \widetilde{\mathbf{h}}_t^{slot} &= ATT(\mathbf{h_t}, \mathbf{M^{slot}}) \\ \mathbf{h}_t^{int} &= ATT([\mathbf{h_t}; \widetilde{\mathbf{h}}_t^{slot}], \mathbf{M^{int}}) \end{aligned} \tag{9}$$

These two deliberate attentions are conducted simultaneously at each word position in such collaborative manner, which guarantees adequate knowledge diffusions between slots and intents. The retrieved slot features $\mathbf{H}_t^{slot}$ and intent features $\mathbf{H}_t^{int}$ are utilized to provide guidances for the next local calculation layer.

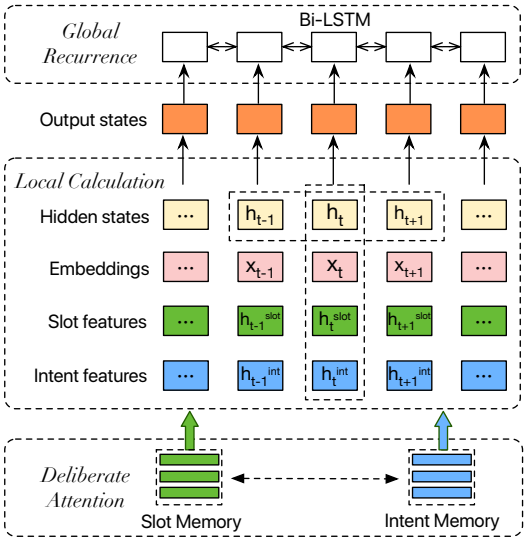

Figure 3: The internal structure of our CM-Block, which is composed of *deliberate attention*, *local calculation* and *global recurrent* respectively.

Local Calculation

Local context information is highly useful for sequence modeling (Kurata et al., 2016; Wang et al., 2016b). Zhang et al. (2018b) propose the S-LSTM to encode both local and sentence-level information simultaneously, and it has been shown more powerful for text representation when compared with the conventional BiLSTMs. We extend the S-LSTM with slot-specific features $\mathbf{H}_t^{slot}$ and intent-specific features $\mathbf{H}_t^{slot}$ retrieved from memories.

Specifically, at each input position t, we take the local window context ξ_t, word embedding $\mathbf{x}_t$, slot feature $\mathbf{h}_t^{slot}$ and intent feature $\mathbf{h}_t^{int}$ as inputs to conduct combinatorial calculation simultaneously. Formally, in the l^{th} layer, the hidden state $\mathbf{h_t}$ is updated as follows:

$$\begin{aligned} \xi_t^{l-1} &= [\mathbf{h}_{t-1}^{l-1}, \mathbf{h}_t^{l-1}, \mathbf{h}_{t+1}^{l-1}] \\ \hat{\mathbf{i}}_t^l &= \sigma(\mathbf{W}_1^i \xi_t^{l-1} + \mathbf{W}_2^i \mathbf{x}_t + \mathbf{W}_3^i \mathbf{h}_t^{slot} + \mathbf{W}_4^i \mathbf{h}_t^{int}) \\ \hat{\mathbf{o}}_t^l &= \sigma(\mathbf{W}_1^o \xi_t^{l-1} + \mathbf{W}_2^o \mathbf{x}_t + \mathbf{W}_3^o \mathbf{h}_t^{slot} + \mathbf{W}_4^o \mathbf{h}_t^{int}) \\ \hat{\mathbf{f}}_t^l &= \sigma(\mathbf{W}_1^f \xi_t^{l-1} + \mathbf{W}_2^f \mathbf{x}_t + \mathbf{W}_3^f \mathbf{h}_t^{slot} + \mathbf{W}_4^f \mathbf{h}_t^{int}) \\ \hat{\mathbf{l}}_t^l &= \sigma(\mathbf{W}_1^l \xi_t^{l-1} + \mathbf{W}_2^l \mathbf{x}_t + \mathbf{W}_3^l \mathbf{h}_t^{slot} + \mathbf{W}_4^l \mathbf{h}_t^{int}) \\ \hat{\mathbf{r}}_t^l &= \sigma(\mathbf{W}_1^r \xi_t^{l-1} + \mathbf{W}_2^r \mathbf{x}_t + \mathbf{W}_3^r \mathbf{h}_t^{slot} + \mathbf{W}_4^r \mathbf{h}_t^{int}) \\ \mathbf{u}_t^l &= \tanh(\mathbf{W}_1^u \xi_t^{l-1} + \mathbf{W}_2^u \mathbf{x}_t + \mathbf{W}_3^u \mathbf{h}_t^{slot} \\ &\qquad + \mathbf{W}_4^u \mathbf{h}_t^{int}) \\ \mathbf{i}_t^l, \mathbf{f}_t^l, \mathbf{l}_t^l, \mathbf{r}_t^l &= softmax(\hat{\mathbf{i}}_t^l, \hat{\mathbf{f}}_t^l, \hat{\mathbf{l}}_t^l, \hat{\mathbf{r}}_t^l) \\ \mathbf{c}_t^l &= \mathbf{f}_t^l \odot \mathbf{c}_t^{l-1} + \mathbf{l}_t^l \odot \mathbf{c}_{t-1}^{l-1} + \mathbf{r}_t^l \odot \mathbf{c}_{t+1}^{l-1} \\ &\qquad + \mathbf{i}_t^l \odot \mathbf{u}_t^{l-1} \\ \mathbf{h}_t^l &= \mathbf{o}_t^l \odot \tanh \mathbf{c}_t^l \end{aligned} \tag{10}$$

where ξ_t^l is the concatenation of hidden states in a local window, and $\mathbf{i}_t^l$, $\mathbf{f}_t^l$, $\mathbf{o}_t^l$, $\mathbf{l}_t^l$ and $\mathbf{r}_t^l$ are gates to control information flows, and $\mathbf{W}_n^x$ $(x \in \{i, o, f, l, r, u\}, n \in \{1, 2, 3, 4\})$ are model parameters. More details about the state transition can be referred in (Zhang et al., 2018b). In the first CM-block, the hidden state $\mathbf{h}_t$ is initialized with the corresponding word embedding. In other CM-blocks, the $\mathbf{h}_t$ is inherited from the output of the adjacent lower CM-block.

At each word position of above procedures, the hidden state is updated with abundant information from different perspectives, namely word embeddings, local contexts, slots and intents representations. The local calculation layer in each CM-block has been shown highly useful for both tasks, and especially for the slot filling task, which will be validated in our experiments in Section 5.2.

Global Recurrence

Bi-directional RNNs, especially the BiLSTMs (Hochreiter and Schmidhuber, 1997) are regarded to encode both past and future information of a sentence, which have become a dominant method in various sequence modeling tasks (Hammerton, 2003; Sundermeyer et al., 2012). The inherent nature of BiLSTMs is able to supplement global sequential information, which is insufficiently modeled in the previous local calculation layer. Thus we apply an additional BiLSTMs layer upon the local calculation layer in each CM-block. By taking the slot- and intent-specific local context representations as inputs, we can obtain more specific global sequential representations. Formally, it takes the hidden state $\mathbf{h}_t^{l-1}$ inherited from the local calculation layer as input, and conduct recurrent steps as follows:

$$\begin{aligned} \mathbf{h}_t^l &= [\overrightarrow{\mathbf{h}_t^l}; \overleftarrow{\mathbf{h}_t^l}] \\ \overrightarrow{\mathbf{h}_t^l} &= \overrightarrow{\mathbf{LSTM}}(\mathbf{h}_t^{l-1}, \overrightarrow{\mathbf{h}}_{t-1}^l; \overrightarrow{\theta}) \\ \overleftarrow{\mathbf{h}_t^l} &= \overleftarrow{\mathbf{LSTM}}(\mathbf{h}_t^{l-1}, \overleftarrow{\mathbf{h}}_{t+1}^l; \overleftarrow{\theta}) \end{aligned} \tag{11}$$

The output "states" of the BiLSTMs are taken as "states" input of the local calculation in next CM-block. The global sequential information encoded by the BiLSTMs is shown necessary and effective for both tasks in our experiments in Section 5.2.

3.4 Inference Layer

After multiple rounds of interactions among local context representations, global sequential information, slot and intent features, we conduct predictions upon the final CM-block. For the predictions of slots, we take the hidden states $\mathbf{H}$ along with the retrieved slot $\mathbf{H}^{slot}$ representations (both are from the final CM-block) as input features, and then conduct predictions of slots similarly with the Equation (3) in Section 2:

$$\hat{\mathbf{y}}^{slot} = \underset{\widetilde{\mathbf{y}}^{slot} \in \mathbf{Y}_x}{\arg\max} F([\mathbf{H}; \mathbf{H}^{slot}], \widetilde{\mathbf{y}}^{slot}) \tag{12}$$

For the prediction of intent label, we firstly aggregate the hidden state $\mathbf{h}_t$ and the retrieved intent representation $\mathbf{h}_t^{int}$ at each word position (from the final CM-block as well) via mean pooling:

$$\mathbf{v}^{int} = \frac{1}{N} \sum_t^N [\mathbf{h_t}; \mathbf{h}_t^{int}] \tag{13}$$

and then take the summarized vector $\mathbf{v}^{int}$ as input feature to conduct prediction of intent consistently with the Equation (5) in Section 2.

Dataset	SNIPS	ATIS	CAIS
Vocab Size	11241	722	2146
Average Length	9.15	11.28	8.65
# Intents	7	18	11
# Slots	72	128	75
# Train Set	13084	4478	7995
# Validation Set	700	500	994
# Test Set	700	893	1012

Table 2: Dataset statistics.

4 Experiments

4.1 Datasets and Metrics

We evaluate our proposed CM-Net on three real-word datasets, and statistics are listed in Table 2.

ATIS The Airline Travel Information Systems (ATIS) corpus (Hemphill et al., 1990) is the most widely used benchmark for the SLU research. Please note that, there are extra named entity features in the ATIS, which almost determine slot tags. These hand-crafted features are not generally available in open domains (Zhang and Wang, 2016; Guo et al., 2014b), therefore we train our model purely on the training set without additional hand-crafted features.

SNIPS SNIPS Natural Language Understanding benchmark [3] (Coucke et al., 2018) is collected in a crowsourced fashion by Snips. The intents of this

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

Models	SNIPS		ATIS	
	Slot (F_1)	Intent (Acc)	Slot (F_1)	Intent (Acc)
Joint GRU (Zhang and Wang, 2016)	–	–	95.49	98.10
Self-Attention, Intent Gate(Li et al., 2018)	–	–	96.52	98.77
Bi-model (Wang et al., 2018)	–	–	**96.89**	98.99
Attention Bi-RNN (Liu and Lane, 2016) *	87.80	96.70	95.98	98.21
Joint Seq2Seq (Hakkani-Tür et al., 2016) *	87.30	96.90	94.20	92.60
Slot-Gated (Intent Atten.) (Goo et al., 2018)	88.30	96.80	95.20	94.10
Slot-Gated (Full Atten.) (Goo et al., 2018)	88.80	97.00	94.80	93.60
CAPSULE-NLU(Zhang et al., 2018a)	91.80	97.70	95.20	95.00
Dilated CNN, Label-Recurrent (Gupta et al., 2019)	93.11	98.29	95.54	98.10
Sentence-State LSTM (Zhang et al., 2018b) †	95.80	98.30	95.65	98.21
BiLSTMs + EMLoL (Siddhant et al., 2018)	93.29	98.83	95.62	97.42
BiLSTMs + EMLo (Siddhant et al., 2018)	93.90	**99.29**	95.42	97.30
Joint BERT (Chen et al., 2019)	97.00	98.60	96.10	97.50
CM-Net (Ours)	**97.15**	**99.29**	96.20	**99.10**

Table 3: Results on test sets of the SNIPS and ATIS, where our CM-Net achieves state-of-the-art performances in most cases. "*" indicates that results are retrieved from Slot-Gated (Goo et al., 2018), and "†" indicates our implementation.

dataset are more balanced when compared with the ATIS. We split another 700 utterances for validation set following previous works (Goo et al., 2018; Zhang et al., 2018a).

CAIS We collect utterances from the **C**hinese **A**rtificial **I**ntelligence **S**peakers (CAIS), and annotate them with slot tags and intent labels. The training, validation and test sets are split by the distribution of intents, where detailed statistics are provided in the supplementary material. Since the utterances are collected from speaker systems in the real world, intent labels are partial to the *PlayMusic* option. We adopt the BIOES tagging scheme for slots instead of the BIO2 used in the ATIS, since previous studies have highlighted meaningful improvements with this scheme (Ratinov and Roth, 2009) in the sequence labeling field.

Metrics Slot filling is typically treated as a sequence labeling problem, and thus we take the conlleval [4] as the token-level F_1 metric. The intent detection is evaluated with the classification accuracy. Specially, several utterances in the ATIS are tagged with more than one labels. Following previous works (Tur et al., 2010; Zhang and Wang, 2016), we count an utterrance as a correct classification if any ground truth label is predicted.

[4]https://www.clips.uantwerpen.be/conll2000/chunking/conlleval.txt

4.2 Implementation Details

All trainable parameters in our model are initialized by the method described in Glorot and Bengio (2010). We apply dropout (Srivastava et al., 2014) to the embedding layer and hidden states with a rate of 0.5. All models are optimized by the Adam optimizer (Kingma and Ba, 2014) with gradient clipping of 3 (Pascanu et al., 2013). The initial learning rate α is set to 0.001, and decrease with the growth of training steps. We monitor the training process on the validation set and report the final result on the test set. One layer CNN with a filter of size 3 and max pooling are utilized to generate 100d word embeddings. The cased 300d Glove is adapted to initialize word embeddings, and kept fixed when training. In auxiliary experiments, the output hidden states of BERT are taken as additional word embeddings and kept fixed as well. We share parameters of both memories with the parameter matrices in the corresponding softmax layers, which can be taken as introducing supervised signals into the memories to some extent. We conduct hyper-parameters tuning for layer size (finally set to 3) and loss weight λ (finally set to 0.5), and empirically set other parameters to the values listed in the supplementary material.

4.3 Main Results

Main results of our CM-Net on the SNIPS and ATIS are shown in Table 3. Our CM-Net achieves the state-of-the-art results on both datasets in terms of slot filling F_1 score and intent detection

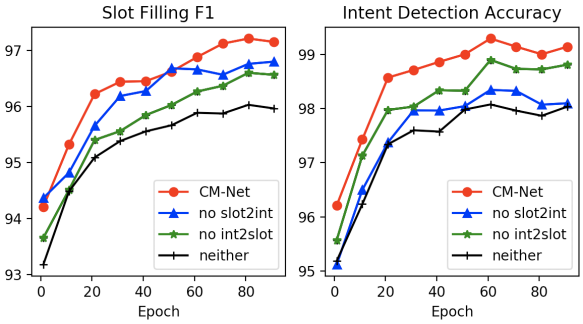

Figure 4: Investigations of the collaborative retrieval approach on slot filling (on the left) and intent detection (on the right), where "no slot2int" indicates removing slow-aware attention for the intent representation, and similarly for "no int2slot" and "neither".

#	Models	SNIPS	
		Slot (F_1)	Intent (Acc)
0	CM-Net	**97.15**	**99.29**
1	– slot memory	96.64	99.14
2	– intent memory	96.95	98.84
3	– local calculation	96.73	99.00
4	– global recurrence	96.80	98.57

Table 4: Ablation experiments on the SNIPS to investigate the impacts of various components, where "- slot memory" indicates removing the slot memory and its interactions with other components correspondingly. Similarly for the other options.

accuracy, except for the F_1 score on the ATIS. We conjecture that the named entity feature in the ATIS has a great impact on the slot filling result as illustrated in Section 4.1. Since the SNIPS is collected from multiple domains with more balanced labels when compared with the ATIS, the slot filling F_1 score on the SNIPS is able to demonstrate the superiority of our CM-Net.

It is noteworthy that the CM-Net achieves comparable results when compared with models that exploit additional language models (Siddhant et al., 2018; Chen et al., 2019). We conduct auxiliary experiments by leveraging the well-known BERT (Devlin et al., 2018) as an external resource for a relatively fair comparison with those models, and report details in Section 5.3.

5 Analysis

Since the SNIPS corpus is collected from multiple domains and its label distributions are more balanced when compared with the ATIS, we choose the SNIPS to elucidate properties of our CM-Net and conduct several additional experiments.

5.1 Whether Memories Promote Each Other?

In the CM-Net, the deliberate attention mechanism is proposed in a collaborative manner to perform information exchange between slots and intents. We conduct experiments to verify whether such kind of knowledge diffusion in both memories can promote each other. More specifically, we remove one unidirectional diffusion (*e.g.* from slot to intent) or both in each experimental setup. The results are illustrated in Figure 4.

We can observe obvious drops on both tasks when both directional knowledge diffusions are removed (CM-Net *vs.* neither). For the slot filling task (left part in Figure 4), the F_1 scores decrease slightly when the knowledge from slot to intent is blocked (CM-Net *vs.* "no slot2int"), and a more evident drop occurs when the knowledge from intent to slot is blocked (CM-Net *vs.* "no int2slot"). Similar observations can be found for the intent detection task (right part in Figure 4).

In conclusion, the bidirectional knowledge diffusion between slots and intents are necessary and effective to promote each other.

5.2 Ablation Experiments

We conduct ablation experiments to investigate the impacts of various components in our CM-Net. In particular, we remove one component among slot memory, intent memory, local calculation and global recurrence. Results of different combinations are presented in Table 4.

Once the slot memory and its corresponding interactions with other components are removed, scores on both tasks decrease to some extent, and a more obvious decline occurs for the slot filling (row 1 *vs.* row 0), which is consistent with the conclusion of Section 5.1. Similar observations can be found for the intent memory (row 2). The local calculation layer is designed to capture better local context representations, which has an evident impact on the slot filling and slighter effect on the intent detection (row 3 *vs.* row 0). Opposite observations occur in term of global recurrence, which is supposed to model global sequential information and thus has larger effect on the intent detection (row 4 *vs.* row 0).

5.3 Effects of Pre-trained Language Models

Recently, there has been a growing body of works exploring neural language models that trained on

Models	SNIPS	
	Slot (F_1)	Intent (Acc)
BiLSTMs + EMLoL	93.29	98.83
BiLSTMs + EMLo	93.90	99.29
Joint BERT	97.00	98.60
CM-Net + BERT	**97.31**	**99.32**

Table 5: Results on the SNIPS benchmark with the assistance of pre-trained language model, where we establish new state-of-the-art results on the SNIPS.

Models	CAIS	
	Slot (F_1)	Intent (Acc)
BiLSTMs + CRF	85.32	93.25
S-LSTM + CRF †	85.74	94.36
CM-Net	**86.16**	**94.56**

Table 6: Results on our CAIS dataset, where "†" indicates our implementation of the S-LSTM.

massive corpora to learn contextual representations (*e.g.* BERT (2018) and EMLo (2018)). Inspired by the effectiveness of language model embeddings, we conduct experiments by leveraging the BERT as an additional feature. The results emerged in Table 5 show that we establish new state-of-the-art results on both tasks of the SNIPS.

5.4 Evaluation on the CAIS

We conduct experiments on our self-collected CAIS to evaluate the generalizability in different language. We apply two baseline models for comparison, one is the popular *BiLSTMs + CRF* architecture (Huang et al., 2015) for sequence labeling task, and the other one is the more powerful sententce-state LSTM (Zhang et al., 2018b). The results listed in Table 6 demonstrate the generalizability and effectiveness of our CM-Net when handling various domains and different languages.

6 Related Work

Memory Network Memory network is a general machine learning framework introduced by Weston et al. (2014), which have been shown effective in question answering (Weston et al., 2014; Sukhbaatar et al., 2015), machine translation (Wang et al., 2016a; Feng et al., 2017), aspect level sentiment classification (Tang et al., 2016), *etc.* For spoken language understanding, Chen et al. (2016) introduce memory mechanisms to encode historical utterances. In this paper, we propose two memories to explicitly capture the semantic correlations between slots and the intent in a given utterance, and devise a novel collaborative retrieval approach.

Interactions between slots and intents Considering the semantic proximity between slots and intents, some works propose to enhance the slot filling task unidirectionally with the guidance of intent representations via gating mechanisms (Goo et al., 2018; Li et al., 2018). Intuitively, the slot representations are also instructive to the intent detection task and thus bidirectional interactions between slots and intents are benefical for each other. Zhang et al. (2018a) propose a hierarchical capsule network to perform interactions from words to slots, slots to intents and intents to words in a pipeline manner, which is relatively limited in capturing the complicated correlations among them. In our CM-Net, information exchanges are performed simultaneously with knowledge diffusions in both directions. The experiments demonstrate the superiority of our CM-Net in capturing the semantic correlations between slots and intents.

Sentence-State LSTM Zhang et al. 2018b propose a novel graph RNN named S-LSTM, which models sentence between words simultaneously. Inspired by the new perspective of state transition in the S-LSTM, we further extend it with task-specific (*i.e.,* slots and intents) representations via our collaborative memories. In addition, the global information in S-LSTM is modeled by aggregating the local features with gating mechanisms, which may lose sight of sequential information of the whole sentence. Therefore, We apply external BiLSTMs to supply global sequential features, which is shown highly necessary for both tasks in our experiments.

7 Conclusion

We propose a novel Collaborative Memory Network (CM-Net) for jointly modeling slot filling and intent detection. The CM-Net is able to explicitly capture the semantic correlations among words, slots and intents in a collaborative manner, and incrementally enrich the information flows with local context and global sequential information. Experiments on two standard benchmarks and our CAIS corpus demonstrate the effectiveness and generalizability of our proposed CM-Net. In addition, we contribute the new corpus (CAIS) to the research community.

Acknowledgments

Liu, Chen and Xu are supported by the National Natural Science Foundation of China (Contract 61370130, 61976015, 61473294 and 61876198), and the Beijing Municipal Natural Science Foundation (Contract 4172047), and the International Science and Technology Cooperation Program of the Ministry of Science and Technology (K11F100010). We sincerely thank the anonymous reviewers for their thorough reviewing and valuable suggestions.

References

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling.

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.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Alice Coucke, Alaa Saade, Adrien Ball, Thodore Bluche, Alexandre Caulier, David Leroy, Clment Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Mal Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

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*.

Yang Feng, Shiyue Zhang, Andi Zhang, Dong Wang, and Andrew Abel. 2017. Memory-augmented neural machine translation.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

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

Daniel Guo, Gokhan Tur, Wen-tau Yih, and Geoffrey Zweig. 2014a. Joint semantic utterance classification and slot filling with recursive neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 554–559. IEEE.

Daniel Guo, Gokhan Tur, Wen-tau Yih, and Geoffrey Zweig. 2014b. Joint semantic utterance classification and slot filling with recursive neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 554–559. IEEE.

Arshit Gupta, John Hewitt, and Katrin Kirchhoff. 2019. Simple, fast, accurate intent classification and slot labeling.

Patrick Haffner, Gokhan Tur, and Jerry H Wright. 2003. Optimizing svms for complex call classification. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

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.

James Hammerton. 2003. Named entity recognition with long short-term memory. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 172–175. Association for Computational Linguistics.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling. *arXiv preprint arXiv:1601.01530*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833, Brussels, Belgium. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*, pages 2227–2237. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning*, pages 147–155. Association for Computational Linguistics.

Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. 2011. Deep belief nets for natural language call-routing. In *2011 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 5680–5683. IEEE.

Aditya Siddhant, Anuj Goyal, and Angeliki Metallinou. 2018. Unsupervised transfer learning for spoken language understanding in intelligent agents. *AAAI*.

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

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

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*.

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

Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2016a. Memory-enhanced decoder for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 278–286, Austin, Texas. Association for Computational Linguistics.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016b. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428–2437.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 309–314, New Orleans, Louisiana. Association for Computational Linguistics.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83. IEEE.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018a. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, pages 2993–2999.

Yue Zhang, Qi Liu, and Linfeng Song. 2018b. Sentence-state lstm for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327. Association for Computational Linguistics.