SYNTAX OR SEMANTICS? KNOWLEDGE-GUIDED JOINT SEMANTIC FRAME PARSING

Yun-Nung Chen[⋆] *Dilek Hakanni-Tür*[†] *Gokhan Tur*[†] *Asli Celikyilmaz*[‡] *Jianfeng Guo*[‡] *Li Deng*[‡]

[⋆]National Taiwan University, Taipei, Taiwan
[†]Google Research, Mountain View, CA
[‡]Microsoft Research, Redmond, WA

{[⋆]y.v.chen, [†]dilek, [†]gokhan.tur, [‡]asli}@ieee.org, {[‡]jfgao, [‡]deng}@microsoft.com

ABSTRACT

Spoken language understanding (SLU) is a core component of a spoken dialogue system, which involves intent prediction and slot filling and also called semantic frame parsing. Recently recurrent neural networks (RNN) obtained strong results on SLU due to their superior ability of preserving sequential information over time. Traditionally, the SLU component parses semantic frames for utterances considering their flat structures, as the underlying RNN structure is a linear chain. However, natural language exhibits linguistic properties that provide rich, structured information for better understanding. This paper proposes to apply knowledge-guided structural attention networks (K-SAN), which additionally incorporate non-flat network topologies guided by prior knowledge, to a language understanding task. The model can effectively figure out the salient substructures that are essential to parse the given utterance into its semantic frame with an attention mechanism, where two types of knowledge, syntax and semantics, are utilized. The experiments on the benchmark Air Travel Information System (ATIS) data and the conversational assistant Cortana data show that 1) the proposed K-SAN models with syntax or semantics outperform the state-of-the-art neural network based results, and 2) the improvement for joint semantic frame parsing is more significant, because the structured information provides rich cues for sentence-level understanding, where intent prediction and slot filling can be mutually improved.

***Index Terms*—** Spoken language understanding, joint semantic frame parsing, knowledge-guided structural attention networks, deep learning, spoken dialogue system

1. INTRODUCTION

In the past decade, goal-oriented spoken dialogue systems (SDS), such as the virtual personal assistants Microsoft's Cortana and Apple's Siri, are being incorporated in various devices and allow users to speak to systems in order to finish tasks more efficiently. A key component of these conversational systems is the spoken language understanding (SLU) module-it refers to the targeted understanding of human speech directed at machines [1]. The goal of SLU is to convert the recognized user speech into a task-specific semantic representation of the user's intention, at each turn, that aligns with the back-end knowledge and action sources for task completion. The dialogue manager then interprets the semantics of the user's request and associated back-end results, and decides the most appropriate system action.

A typical pipeline of SLU includes: domain classification, intent determination, and slot filling [1], where the SLU module first decides the domain of user's request given the input utterance, and based on the domain, predicts the intent and fills associated slots corresponding to a domain-specific semantic template. For example, Figure 1 shows a user utterance, "*tell vivian to be quiet*" and its semantic frame, send_text(contact_name="vivian", message="be quiet") that can be executed in the communication domain. In this example, it is easy to see the relationship between the receiver and the message, although these do not appear next to each other in the sentence. Traditionally, domain detection and intent prediction are framed as utterance classification problems, where several classifiers such as support vector machines and maximum entropy are employed [2, 3, 4]. Then slot filling is framed as a word sequence tagging task, where the IOB (in-out-begin) format is applied for representing slot tags as illustrated in Figure 1, and hidden Markov models (HMM) or conditional random fields (CRF) have been employed for slot tagging [5, 6, 7].

W	tell	vivian	to	be	quiet
	↓	↓	↓	↓	↓
S	O	B-contact_name	O	B-message	I-message
D	communication				
I	send_text				

Fig. 1. An example utterance annotated with its semantic slots in the IOB format (S), domain (D), and intent (I).

With the advances on deep learning, neural network based approaches have been applied to domain and intent classification [8, 9, 10]. Recently, Ravuri and Stolcke proposed an RNN architecture for intent determination [11]. For slot filling, deep learning has been viewed as a feature generator and the neural architecture can be merged with CRFs [12, 13]. Yao et al. and Mesnil et al. later employed RNNs for sequence labeling in order to perform slot filling [14, 15]. However, such pipelining of tasks results in transfer of errors from one task to the following tasks. Hakkani-Tür proposed an RNN architecture that incorporates both intent prediction and slot filling so that the information can be mutually enhanced [16]. Nevertheless, the above studies benefit from large training data without leveraging any existing knowledge. When tagging sequences RNNs consider them as flat structures, with their underlying linear chain structures, potentially ignoring the structured information typical of natural language sequences.

Hierarchical structures and semantic relationships contain linguistic characteristics of input word sequences forming sentences, and such information may help interpret their meaning. Furthermore, prior knowledge would help in the tagging of sequences, especially when dealing with previously unseen sequences. Prior work exploited external web-scaled knowledge graphs such as Freebase and Wikipedia for improving SLU [17, 18, 4, 19], Liu et al. and Chen et al. proposed approaches that leverage linguistic knowledge encoded in parse trees for language understanding, where the extracted syntactic structural features and semantic dependency features enhance inference model learning, and the model achieves better understanding performance in various domains [20, 21]. Even with the emerging paradigm of integrating deep learning and linguistic knowledge for different NLP tasks [22], most of the previous work utilized such linguistic knowledge and knowledge bases as additional features as input to neural networks, and then learned the models for tagging sequences. These feature enrichment based approaches have some possible limitations: 1) poor generalization and 2) error propagation. Poor generalization comes from the mismatch between knowledge bases and the input data, and then the incorrectly extracted features due to errors in previous processing propagate errors to the neural models.

This paper focuses on leveraging the benefits from both 1) joint semantic frame parsing and 2) prior linguistic knowledge. This paper proposes to apply knowledge-guided structural attention networks, K-SAN [23], to automatically learn the attention guided by external or prior knowledge and generate sentence-based representations specifically for joint semantic frame parsing, where two types of prior knowledge are investigated. The main difference between K-SAN and previous approaches is that knowledge plays the role of a teacher to guide networks where and how much to focus attention considering the whole linguistic structure simultaneously. Our main contributions are three-fold:

- Joint end-to-end learning
  To our knowledge, this is the first neural network approach that utilizes general knowledge as guidance in an end-to-end fashion for a joint task, where the model automatically learns important substructures with an attention mechanism and considers intent prediction and slot filling simultaneously.
- Investigation of different knowledge
  Different types of knowledge are employed in the model, and the analysis can guide the future research directions.
- Efficiency and parallelizability
  Because the knowledge-guided substructures from the input utterance are modeled separately, modeling time may not increase linearly with respect to the number of words in the input sentence.

2. RELATED WORK

Knowledge-Based Representation There is an emerging trend of learning representations at different levels, such as word embeddings [24], character embeddings [25], and sentence embeddings [26, 27]. In addition to fully unsupervised embedding learning, knowledge bases have been widely utilized to learn entity embeddings with specific functions or relations [28, 29]. This paper focuses on leveraging substructure embeddings for joint semantic frame parsing.

Recently linguistic structures are taken into account in the deep learning framework. Ma et al. and Tai et al. both proposed dependency-based approaches to combine deep learning and linguistic structures, where the model used tree-based n-grams instead of surface ones to capture knowledge-guided relations for sentence modeling and classification [30, 31]. Roth and Lapata utilized lexicalized dependency paths to learn embedding representations for semantic role labeling [32]. However, the performance of these approaches highly depends on the quality of "whole" sentence parsing, and there is no control of degree of attentions on different substructures. Learning robust representations incorporating whole structures still remains unsolved.

Neural Attention and Memory Model The earliest work with a memory component applied to language processing is memory networks [33, 34], which encode facts into vectors and store them in the memory for question answering (QA). Following their success, Xiong proposed dynamic memory networks (DMN) to additionally capture position and temporality of transitive reasoning steps for different QA tasks [35]. The idea is to encode important knowledge and store it into memory for future usage with attention mechanisms. Attention mechanisms allow neural network models to selectively pay attention to specific parts. There are also various tasks showing the effectiveness of attention mechanisms [36].

However, most previous work focused on the classification or prediction tasks (predicting a single word given a question), and there are few studies for SLU tasks (slot tagging). Based on the fact that the linguistic or knowledge-based substructures can be treated as prior knowledge to benefit language understanding, this work borrows the idea from memory models to improve SLU. Unlike the prior SLU work that utilized representations learned from knowledge bases to enrich features of the current sentence, this paper directly learns a sentence representation incorporating memorized substructures with an automatically decided attention mechanism in an end-to-end manner.

3. KNOWLEDGE-GUIDED STRUCTURAL ATTENTION NETWORKS (K-SAN)

For the SLU task, given an utterance with a sequence of words/tokens $\vec{s} = w_1, ..., w_T$, our model is to predict corresponding semantic frame including an intent and associated slots. The whole semantic frame is formulated into a semantic tag sequence $\vec{y} = y_1, ..., y_T, y_{T+1}$, where y_{T+1} denotes the intent label and $y_1, ..., y_T$ represent the slot tags associated with $\vec{s}$ [16]. The proposed model is to predict the semantic frame $\vec{y}$ given a word sequence $\vec{s}$ with incorporation of knowledge-guided structures, which is composed of a *knowledge encoding module* and a *RNN tagging module*. The framework of knowledge-guided structural attention networks (K-SAN) is illustrated in Figure 2 [23].

The knowledge encoding module first leverages external knowledge to generate a linguistic structure for the utterance, where a discrete set of knowledge-guided substructures $\{x_i\}$ is encoded into a set of vector representations (§ 3.1). The model learns the representation for the whole sentence by paying different attention on the substructures. Then the input vector and the structural vector are encoded together into a knowledge-guided representation, and then is utilized to improve the semantic tagger (§ 5). The whole architecture is trained in an end-to-end setting, where the final objective is to maximize the tagging performance.

3.1. Knowledge Encoding Module

The model embeds all knowledge-guided substructures into a continuous space and stores embeddings of all x's in the knowledge memory. The representation of the input utterance is then compared with encoded knowledge representations to integrate the carried structure guided by knowledge via an attention mechanism. Then the knowledge-guided representation of the sentence is taken together

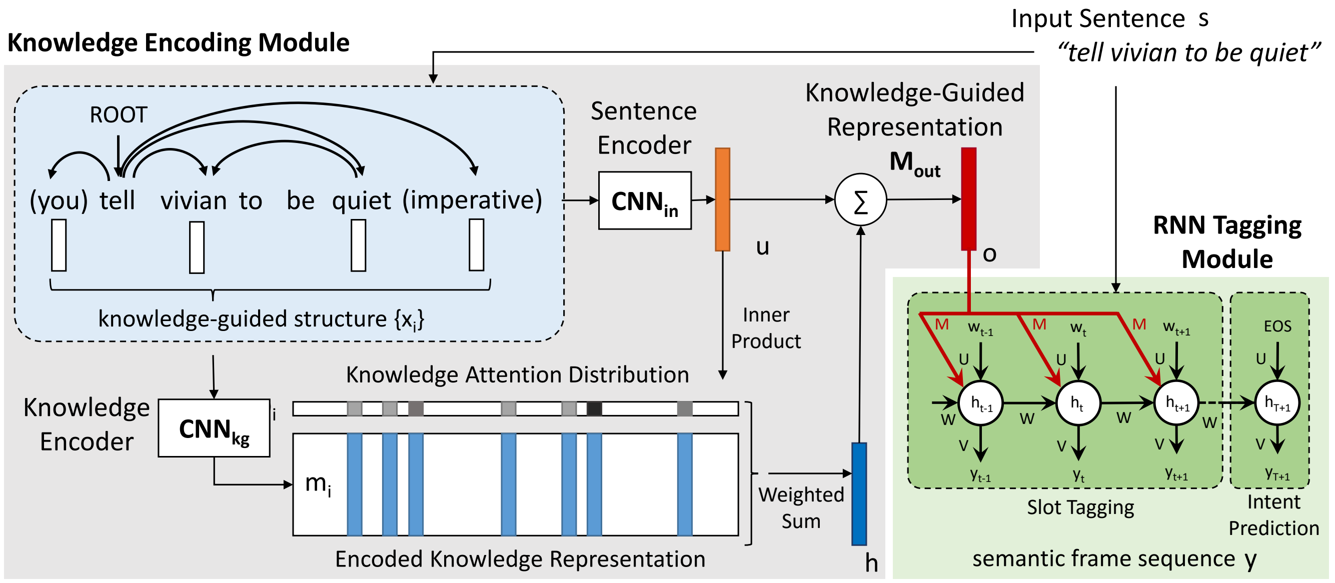

Knowledge Encoding Module
Input Sentence s
"tell vivian to be quiet"
ROOT
(you) tell vivian to be quiet (imperative)
knowledge-guided structure {x_i}
Sentence Encoder
CNN_in
u
Knowledge-Guided Representation
M_out
Σ
o
Inner Product
Knowledge Attention Distribution
Knowledge Encoder
CNN_kg
m_i
Weighted Sum
h
Encoded Knowledge Representation
RNN Tagging Module
M
w_t-1
w_t
w_t+1
EOS
U
W
V
h_t-1
h_t
h_t+1
h_T+1
y_t-1
y_t
y_t+1
y_T+1
Slot Tagging
Intent Prediction
semantic frame sequence y

Fig. 2. The illustration of knowledge-guided structural attention networks (K-SAN) for joint semantic frame parsing.

with the word sequence for estimating the semantic tags. Four main procedures are described below.

Encoded Knowledge Representation To store the knowledge-guided structure, we convert each substructure within the utterance, x_i, into a structure vector m_i with dimension d by embedding the substructure in a continuous space through the knowledge encoding module CNN_{kg}. The construction procedure of knowledge-guided substructures is detailed in § 4. The input utterance s is also embedded to a vector u with the same dimension through the encoder CNN_{in}.

$$m_i = \text{CNN}_{\text{kg}}(x_i), \tag{1}$$

$$u = \text{CNN}_{\text{in}}(s). \tag{2}$$

We apply convolutional neural networks (CNN) with a window size 3 and a max-pooling operation for knowledge encoding models, CNN_{kg} and CNN_{in}, in order to model multiple words from a substructure x_i or an input sentence s into a vector representation. For example, we assume that one of substructures associated with the utterance "*tell vivian to be quiet*" is "*tell quiet vivian*". Figure 3 illustrates the procedure for encoding knowledge-guided substructures into a vector embedding. In the experiments, the weights of CNN_{kg} and CNN_{in} are tied together during optimization.

Knowledge Attention Distribution In the embedding space, we compute the match between the current utterance vector u and its substructure vector m_i by taking their inner product followed by a softmax.

$$p_i = \text{softmax}(u^T m_i), \tag{3}$$

where $\text{softmax}(z_i) = e^{z_i} / \sum_j e^{z_j}$ and p_i can be viewed as attention distribution for modeling important substructures from external knowledge in order to understand the current utterance.

Sentence Representation In order to encode the knowledge-guided structure, a vector h is a sum over the encoded knowledge embeddings weighted by the attention distribution.

$$h = \sum_i p_i m_i, \tag{4}$$

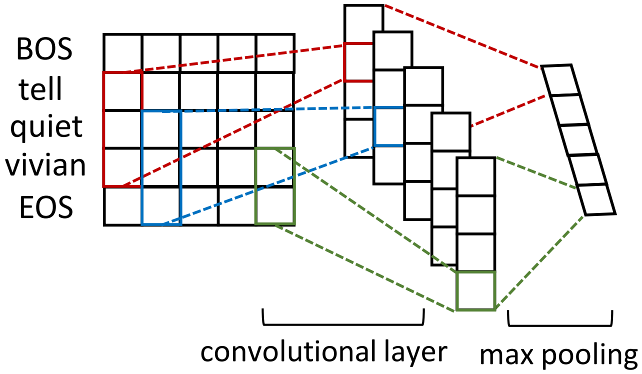


Fig. 3. The illustration of CNN knowledge encoder on a knowledge-guided substructure "*tell quiet vivian*".

which indicates that the sentence pays different attention to different substructures guided from external knowledge. Because the function from input to output is smooth, we can easily compute gradients and back propagate through it. Then the sum of the substructure vector h and the current input embedding u are then passed through a neural network model M_{out} to generate an output knowledge-guided representation o.

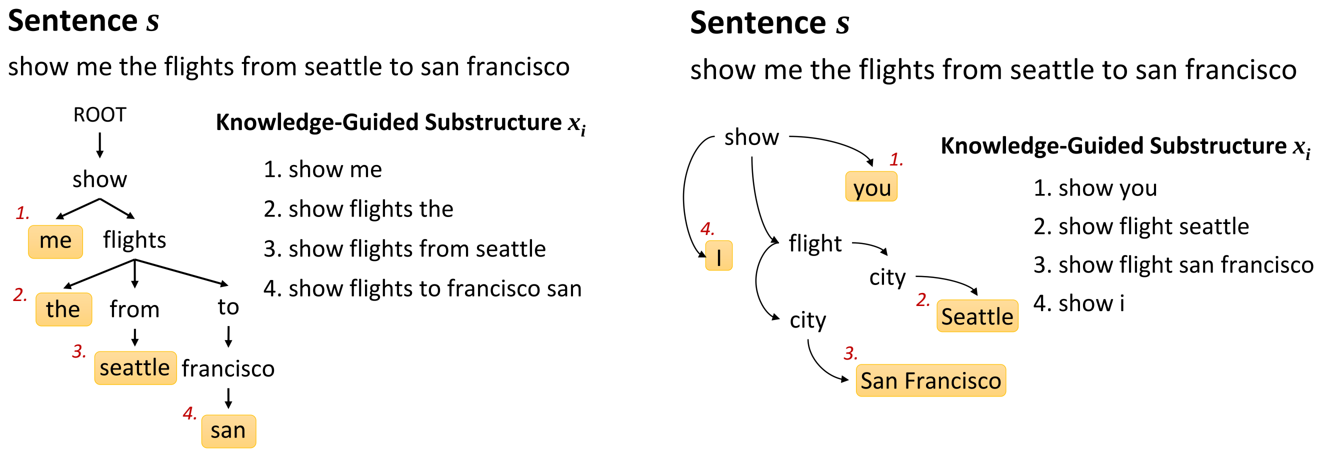
$$o = M_{\text{out}}(h + u), \tag{5}$$

where we employ a fully-connected dense network for M_{out}.

3.2. RNN Tagging Module

To estimate the tag sequence $\vec{y}$ corresponding to an input word sequence $\vec{s}$, we use an RNN module for training a slot tagger, where the knowledge-guided representation o is fed into the input of the model in order to incorporate the structure information. The detail is presented in § 5.

$$\vec{y} = \text{RNN}(o, \vec{s}) \tag{6}$$


Sentence s
show me the flights from seattle to san francisco
ROOT
show
1.
me
flights
2.
the
from
to
3.
seattle
francisco
4.
san
Knowledge-Guided Substructure x_i
1. show me
2. show flights the
3. show flights from seattle
4. show flights to francisco san
Sentence s
show me the flights from seattle to san francisco
show
1.
you
4.
I
flight
city
2.
Seattle
city
3.
San Francisco
Knowledge-Guided Substructure x_i
1. show you
2. show flight seattle
3. show flight san francisco
4. show i

(a) Syntax: the dependency tree parsed by SyntexNet

(b) Semantics: the AMR graph

Fig. 4. The constructing procedure of knowledge-guided substructures, x_i, on an example sentence s.

4. KNOWLEDGE-GUIDED STRUCTURE CONSTRUCTION

The prior knowledge obtained from external resources, such as dependency relations, knowledge bases, etc., provides richer information to help decide the semantic tags given an input utterance. The top-left component of Figure 2 illustrates the procedure for modeling knowledge-guided substructures. Two types of main linguistic properties, syntax and semantics, are applied in the K-SAN model.

4.1. Syntax: Dependency Parsing Tree

The dependency relation is the basic syntactic information for knowledge encoding. Several prior studies demonstrated the effectiveness of such knowledge for better understanding. The input utterance is parsed by a dependency parser, and the substructures are built according to the paths from the root to all leaves [37]. For example, the dependency parsing of the utterance "*show me the flights from seattle to san francisco*" is shown in Figure 4(a), where the associated substructures are obtained from the parsing tree for knowledge encoding. Here we only utilize the dependency relations without their labels in the experiments. Note that the number of substructures may be less than the number of words in the utterance, because non-leaf nodes do not have corresponding substructure in order to reduce the duplicated information in the model.

4.2. Semantics: Abstract Meaning Representation (AMR)

Abstract Meaning Representation (AMR) is a semantic formalism in which the meaning of a sentence is encoded as a rooted, directed, acyclic graph [38], where nodes represent concepts, and labeled directed edges represent the relations between two concepts. The formalism is based on propositional logic and neo-Davidsonian event representations [39, 40]. The semantic concepts in AMR were leveraged to benefit multiple NLP tasks [41]. Figure 4(b) shows an AMR graph associated with the same example utterance.

5. RECURRENT NEURAL NETWORK TAGGER

5.1. Chain-Based RNN Tagger

Given $\vec{s} = w_1, ..., w_T$, the model is to predict $\vec{y} = y_1, ..., y_T$ where the tag y_i is aligned with the word w_i. We use the Elman RNN architecture, consisting of an input layer, a hidden layer, and an output layer [42]. The input, hidden and output layers consist of a set of neurons representing the input, hidden, and output at each time step t, w_t, h_t, and y_t, respectively.

$$h_t = \phi(Ww_t + Uh_{t-1}), \quad (7)$$

$$\hat{y}_t = \text{softmax}(Vh_t), \quad (8)$$

where ϕ is a smooth bounded function such as tanh, and $\hat{y}_t$ is the probability distribution over semantic tags given the current hidden state h_t. The sequence probability can be formulated as

$$p(\vec{y} \mid \vec{s}) = p(\vec{y} \mid w_1, ..., w_T) = \prod_i p(y_i \mid w_1, ..., w_i). \quad (9)$$

The model can be trained using backpropagation to maximize the conditional likelihood of the training set labels.

To overcome the frequent vanishing gradients issue when modeling long-term dependencies, gated RNN was designed to use a more sophisticated activation function than a usual activation function, consisting of affine transformation followed by a simple element-wise nonlinearity by using gating units [43], such as long short-term memory (LSTM) and gated recurrent unit (GRU) [44, 45]. RNNs employing either of these recurrent units have been shown to perform well in tasks that require capturing long-term dependencies [15, 46, 47, 48]. In this paper, we mainly use RNN with GRU cells to allow each recurrent unit to adaptively capture dependencies of different time scales [45, 43], because RNN-GRU can yield comparable performance as RNN-LSTM with fewer parameters and less data for generalization [43]

A GRU has two gates, a *reset gate* r, and an *update gate* z [45, 43]. The reset gate determines the combination between the new input and the previous memory, and the update gate decides how

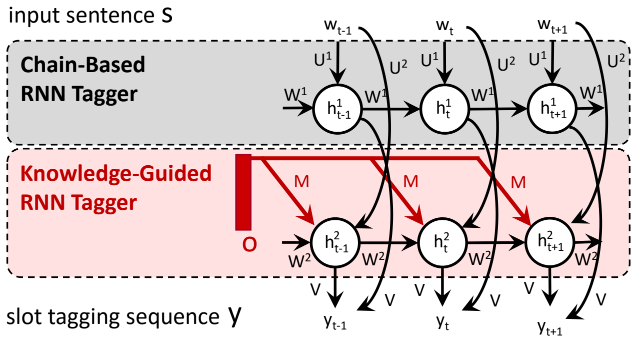


Fig. 5. The joint tagging model that incorporates a chain-based RNN tagger (upper block) and a knowledge-guided RNN tagger (lower block).

much the unit updates its activation, or content.

$$
\begin{aligned}
r &= \sigma(W^r w_t + U^r h_{t-1}), && (10)\\
z &= \sigma(W^z w_t + U^z h_{t-1}), && (11)
\end{aligned}
$$

where σ is a logistic sigmoid function, W^r and U^r are the matrices corresponding to the *reset gate*, and W^z and U^z are the matrices corresponding to the *update gate*.

Then the final activation of the GRU at time t, h_t, is a linear interpolation between the previous activation h_{t-1} and the candidate activation $\tilde{h_t}$:

$$
\begin{aligned}
h_t &= (1 - z) \odot \tilde{h_t} + z \odot h_{t-1}, && (12)\\
\tilde{h_t} &= \phi(W^h w_t + U^h (h_{t-1} \odot r))), && (13)
\end{aligned}
$$

where $\odot$ is an element-wise multiplication, and W^h and U^h are the weight matrices. When the reset gate is off, it effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state. Then $\hat{y}_t$ can be computed by (8).

5.2. Knowledge-Guided RNN Tagger

In order to model the encoded knowledge from previous turns, for each time step t, the knowledge-guided sentence representation o in (5) is fed into the RNN model together with the word w_t. For the plain RNN, the hidden layer can be formulated as

$$h_t = \phi(Mo + Ww_t + Uh_{t-1}) \qquad (14)$$

to replace (7) as illustrated in the right block of Figure 2. RNN-GRU can incorporate the encoded knowledge in the similar way, where Mo can be added into gating mechanisms for modeling contextual knowledge similarly.

5.3. Joint RNN Tagger

Because the chain-based tagger and the knowledge-guided tagger carry different information, the joint RNN tagger is proposed to balance the information between two model architectures. Figure 5 presents the architecture of the joint RNN tagger.

$$
\begin{aligned}
h_t^1 &= \phi(W^1 w_t + U^1 h_{t-1}), && (15)\\
h_t^2 &= \phi(Mo + W^2 w_t + U^2 h_{t-1}), && (16)\\
\hat{y}_t &= \text{softmax}(V(\alpha h_t^1 + (1-\alpha) h_t^2)), && (17)
\end{aligned}
$$

Table 1. The statistics of datasets

Dataset	Train			Dev	Test	#Intent	#Slot
	Small	Medium	Large				
ATIS	129	515	4,478	500	893	17	79
Cortana	230	1,148	10,479	1,000	2,300	25	20

where α is the weight for balancing chain-based and knowledge-guided information. By jointly considering chain-based information (h_t^1) and knowledge-guided information (h_t^2), the joint RNN tagger is expected to achieve better generalization, and the performance may be less sensitive to poor structures from external knowledge. In the experiments, α is set to 0.5 for balancing two sides. Other model parameters are trained by maximizing the sequence probability $p(\vec{y} \mid \vec{s})$ in (9).

6. EXPERIMENTS

6.1. Experimental Setup

In the experiments, two datasets are used.

- ATIS: is a benchmark dataset that is extensively used by the SLU community [15]. There are 4978 training utterances selected from Class A (context independent) in the ATIS-2 and ATIS-3, while there are 893 utterances selected from the ATIS-3 Nov93 and Dec94.
- Cortana Communication: is conversational data addressed to a virtual intelligent assistant, Cortana, and related to the communication domain.

In the experiments, we conduct the experiments with 3 different sizes of data (Small, Medium, and Large) in order to show the robustness to data scarcity. The number of training, development and test utterances of these datasets are listed in Table 1. The evaluation metrics for SLU includes F-measure on the predicted slots and accuracy on the whole semantic frames[1].

For experiments with K-SAN, we parse all data with the pre-trained parsers from SyntaxNet[2] to generate syntactic knowledge and JAMR[3] for semantic knowledge. We represent words as its embeddings trained on the in-domain data, the loss function is cross-entropy, and the optimizer we use is adam with the default setting [49], where the learning rate $\lambda = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e^{-08}$. The maximum iteration for training our K-SAN models is set as 300. The dimensionality of input word embeddings is 100, and the hidden layer sizes are in $\{50, 100, 150\}$. The dropout rates are set as $\{0.25, 0.50\}$. All reported results are from the joint RNN tagger, and the hyperparameters are tuned in the dev set for all experiments.

6.2. Baseline

To validate the effectiveness of the proposed model, we compare the performance with the following baselines.

- RNN Tagger (Indep): predicts a slot tag for each word in the given utterance.

[1] The slot filling results are performed via an evaluation script conlleval.

[2] https://github.com/tensorflow/models/tree/master/syntaxnet

[3] https://github.com/jflanigan/jamr

Table 2. The F1 scores of predicted slots on the different size of ATIS and Cortana training examples. Dataset Small: 1/40 set; Dataset Medium: 1/10 set; Dataset Large: original set. The performance for slot filling is F-measure and for the whole frame is accuracy (%).

Data	Model		Small (1/40)			Medium (1/10)			Large (full)		
	Approach	Knowledge	Slot Filling		Frame	Slot Filling		Frame	Slot Filling		Frame
			Indep	Joint		Indep	Joint		Indep	Joint	
ATIS	RNN Tagger	✗	73.83	72.97	33.45	85.55	86.42	58.45	93.11	93.42	79.73
	RNN Encoder-Tagger	✗	72.79	71.90	35.16	88.26	87.51	61.93	94.75	93.11	82.53
	K-SAN	Syntax	74.35	**74.56**	**37.63**	**88.40**	88.24	63.49	95.00	**95.38**	**84.32**
	K-SAN	Semantics	74.27	73.41	37.07	88.27	88.08	**63.61**	94.89	95.08	83.76
Cortana	RNN Tagger	✗	45.47	50.25	48.87	69.03	69.84	68.22	80.35	79.83	79.52
	RNN Encoder-Tagger	✗	45.49	47.71	52.70	69.42	73.07	71.43	85.71	**85.97**	**83.87**
	K-SAN	Syntax	45.04	**55.10**	**57.17**	69.46	**75.30**	73.48	85.04	84.54	83.48
	K-SAN	Semantics	45.10	54.96	54.13	69.08	74.27	**73.78**	85.33	85.18	83.43

- RNN Tagger (Joint): predicts the whole semantic frame for the given utterance.
- RNN Encoder-Tagger (Indep): encodes the input word sequence into a vector and then tags each word by additionally incorporating the sentence-level vector.
- RNN Encoder-Tagger (Joint): predicts the semantic frame by additionally incorporating the sentence-level vector.

6.3. Slot Filling Results

Table 2 shows the performance of slot filling using independent and joint approaches on different size of training data, where there are three datasets (Small, Medium, and Large). For baselines, RNN Encoder-Tagger performs better than RNN Tagger on the Medium and Large sets (for both ATIS and Cortana data), but slightly worse on the Small set. The probable reason is that the Small set does not contain enough training data to learn more parameters in the RNN Encoder-Tagger.

In terms of the proposed K-SAN approaches, both syntax and semantics are useful for achieving better tagging performance in all cases. Among them, syntax helps improve slot tagging results more than semantics for most of cases, probably because syntactic relations are more general and the number of relations is larger.

6.4. Semantic Frame Parsing Results

Table 2 also shows the frame-level accuracy of joint approaches, which better correlates to system performance. Between two baselines, RNN Encoder-Tagger approaches outperform RNN tagger methods, indicating the correlation between slots and intents and showing the benefit of joint semantic frame parsing.

For K-SAN approaches, all semantic frame results but one from Cortana Large data are better than the baselines, where the improvement is significantly larger on the smaller sets (35.2% to 37.6% of frame accuracy on the ATIS Small set and 52.7% to 57.2% on the Cortana Small set). The reason of no improvement on Cortana Larger data may be that the baseline model already achieves good enough performance based on 10K utterances, and the additional knowledge does not provide further improvement.

6.5. Discussion

6.5.1. Effectiveness of K-SAN

In the experiments, the proposed models outperform the baselines in most of cases, where the improvement for the small dataset is more significant. This suggests that the proposed models carry better generalization and are less sensitive to unseen data. The proposed model presents the state-of-the-art performance on the Small and Medium datasets, showing the effectiveness of leveraging knowledge-guided structures for learning embeddings that can be used for specific tasks and the robustness to data scarcity and mismatch.

6.5.2. Comparing between Syntax and Semantics

Considering the fact that syntactic relations are general and semantic relations are specific, the number of syntactic relations is larger than the number of semantic relations. For smaller datasets, more relations may offer richer information so that the model can learn the attention for substructures better. On the other hand, although the number of semantic relations is less, the specific and accurate semantic relations provide stronger guidance for the model. Less but stronger relations allow the model to achieve similar performance with less training time.

6.5.3. Comparing between Independent and Joint Parsing

Comparing between independent and joint methods, for Cortana Small and Medium sets, joint approaches significantly outperform independent ones for both baselines and proposed models. However, the difference between independent and joint methods is not significant for ATIS data. It implies that the intent information is more informative for Cortana data than for ATIS data, because the ratio between intents and slots of Cortana is 25:20 ($= 1.25$), which is much larger than one of ATIS (17:79 ≈ 0.22). Therefore, joint semantic frame parsing benefits the data with diverse intents more than the data with less intents, and the Cortana results show the great potential of the joint method.

7. CONCLUSION

This paper investigates the effectiveness of different types of linguistic properties in knowledge-guided structural attention networks (K-SAN), which additionally incorporate non-flat network topologies guided by prior knowledge, for joint semantic frame parsing. The experiments demonstrate the effectiveness of syntax and semantics for guiding the attention and the mutual improvement between intent prediction and slot filling the structural knowledge brings, showing the feasibility of applying to neural network training with limited training data for spoken language understanding.

8. REFERENCES

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

[2] Patrick Haffner, Gokhan Tur, and Jerry H Wright, "Optimizing svms for complex call classification," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP)*. IEEE, 2003, vol. 1, pp. I–632.

[3] Cipriun Chelba, Monika Mahajan, and Alex Acero, "Speech utterance classification," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP)*. IEEE, 2003, vol. 1, pp. I–280.

[4] Yun-Nung Chen, Dilek Hakkani-Tur, and Gokan Tur, "Deriving local relational surface forms from dependency-based entity embeddings for unsupervised spoken language understanding," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 242–247.

[5] Roberto Pieraccini, Evelyne Tzoukermann, Zakhar Gorelov, Jean-Luc Gauvain, Esther Levin, Chin-Hui Lee, and Jay G Wilpon, "A speech understanding system based on statistical representation of semantics," in *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 1992, vol. 1, pp. 193–196.

[6] Ye-Yi Wang, Li Deng, and Alex Acero, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, 2005.

[7] Christian Raymond and Giuseppe Riccardi, "Generative and discriminative algorithms for spoken language understanding.," in *INTERSPEECH*, 2007, pp. 1605–1608.

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

[9] Gokhan Tur, Li Deng, Dilek Hakkani-Tür, and Xiaodong He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5045–5048.

[10] Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 778–784, 2014.

[11] Suman Ravuri and Andreas Stolcke, "Recurrent neural network and lstm models for lexical utterance classification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[12] Puyang Xu and Ruhi Sarikaya, "Convolutional neural network based triangular CRF for joint intent detection and slot filling," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 78–83.

[13] Vedran Vukotic, Christian Raymond, and Guillaume Gravier, "Is it time to switch to word embedding and recurrent neural networks for spoken language understanding?," in *InterSpeech*, 2015.

[14] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu, "Recurrent neural networks for language understanding.," in *INTERSPEECH*, 2013, pp. 2524–2528.

[15] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al., "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2015.

[16] Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang, "Multi-domain joint semantic frame parsing using bi-directional rnn-lstm," in *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*, 2016.

[17] Larry P Heck, Dilek Hakkani-Tür, and Gokhan Tur, "Leveraging knowledge graphs for web-scale unsupervised semantic parsing.," in *INTERSPEECH*, 2013, pp. 1594–1598.

[18] Yi Ma, Paul A Crook, Ruhi Sarikaya, and Eric Fosler-Lussier, "Knowledge graph inference for spoken dialog systems," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5346–5350.

[19] Yun-Nung Chen and Alexander I Rudnicky, "Dynamically supporting unexplored domains in conversational interactions by enriching semantics with neural word embeddings," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 590–595.

[20] Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and James Glass, "Query understanding enhanced by hierarchical parsing structures," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 72–77.

[21] Yun-Nung Chen, William Yang Wang, Anatole Gershman, and Alexander I Rudnicky, "Matrix factorization with knowledge graph propagation for unsupervised spoken language understanding," *Proceedings of ACL-IJCNLP*, 2015.

[22] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014.

[23] Yun-Nung Chen, Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Jianfeng Gao, and Li Deng, "Knowledge as a teacher: Knowledge-guided structural attention networks," *arXiv preprint arXiv:1609.03286*, 2016.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[25] Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso, "Finding function in form: Compositional character models for open vocabulary word representation," *arXiv preprint arXiv:1508.02096*, 2015.

[26] Quoc V Le and Tomas Mikolov, "Distributed representations of sentences and documents," *arXiv preprint arXiv:1405.4053*, 2014.

[27] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 2333–2338.

[28] Asli Celikyilmaz and Dilek Hakkani-Tur, “Convolutional neural network based semantic tagging with entity embeddings,” in *NIPS Workshop on Machine Learning for SLU and Interaction*, 2015.

[29] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng, “Embedding entities and relations for learning and inference in knowledge bases,” *arXiv preprint arXiv:1412.6575*, 2014.

[30] Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou, “Dependency-based convolutional neural networks for sentence embedding,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 174–179.

[31] Kai Sheng Tai, Richard Socher, and Christopher D Manning, “Improved semantic representations from tree-structured long short-term memory networks,” *arXiv preprint arXiv:1503.00075*, 2015.

[32] Michael Roth and Mirella Lapata, “Neural semantic role labeling with dependency path embeddings,” *arXiv preprint arXiv:1605.07515*, 2016.

[33] Jason Weston, Sumit Chopra, and Antoine Bordesa, “Memory networks,” in *International Conference on Learning Representations (ICLR)*, 2015.

[34] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al., “End-to-end memory networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2431–2439.

[35] Caiming Xiong, Stephen Merity, and Richard Socher, “Dynamic memory networks for visual and textual question answering,” *arXiv preprint arXiv:1603.01417*, 2016.

[36] Yun-Nung Chen, Dilek Hakkani-Tür, Gokhan Tur, Jianfeng Gao, and Li Deng, “End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding,” in *Proceedings of Interspeech*, 2016.

[37] Danqi Chen and Christopher D Manning, “A fast and accurate dependency parser using neural networks.,” in *EMNLP*, 2014, pp. 740–750.

[38] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider, “Abstract meaning representation for sembanking,” in *Proceedings of the Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[39] Terence Parsons, “Events in the semantics of english: A study in subatomic semantics,” 1990.

[40] Donald Davidson, “The logical form of action sentences,” 1967.

[41] Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith, “Toward abstractive summarization using semantic representations,” in *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 1077–1086.

[42] Jeffrey L Elman, “Finding structure in time,” *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[43] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[44] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[46] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi, “Spoken language understanding using long short-term memory neural networks,” in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 189–194.

[47] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[48] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[49] Diederik Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.