Learning to Tag OOV Tokens by Integrating Contextual Representation and Background Knowledge

Keqing He, Yuanmeng Yan, Weiran Xu [*]
Pattern Recognition & Intelligent System Laboratory
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications, Beijing, China
{kqin,yanyuanmeng,xuweiran}@bupt.edu.cn

Abstract

Neural-based context-aware models for slot tagging have achieved state-of-the-art performance. However, the presence of OOV(out-of-vocab) words significantly degrades the performance of neural-based models, especially in a few-shot scenario. In this paper, we propose a novel knowledge-enhanced slot tagging model to integrate contextual representation of input text and the large-scale lexical background knowledge. Besides, we use multi-level graph attention to explicitly model lexical relations. The experiments show that our proposed knowledge integration mechanism achieves consistent improvements across settings with different sizes of training data on two public benchmark datasets.

1 Introduction

Slot tagging is a critical component of spoken language understanding(SLU) in dialogue systems. It aims at parsing semantic concepts from user utterances. For instance, given the utterance *"I'd also like to have lunch during my flight"* from the ATIS dataset, a slot tagging model might identify *lunch* as a *meal_description* type. Given sufficient training data, recent neural-based models (Mesnil et al., 2014; Liu and Lane, 2015, 2016; Goo et al., 2018; Haihong et al., 2019; He et al., 2020) have achieved remarkably good results.

However, these works often suffer from poor slot tagging accuracy when rare words or OOV(out-of-vocab) words exist. (Ray et al., 2018) has verified the presence of OOV words further degrades the performance of neural-based models, especially in a few-shot scenario where training data can not provide adequate contextual semantics. Previous context-aware models merely focus on how to capture deep contextual semantics to aid in recognizing slot entities, while neglecting ontology behind the words or large-scale background knowledge. Explicit lexical relations are vital to recognizing unseen words when there is not adequate training data, that is, few-shot scenarios. Fig 1 gives a motivating example of slot tagging to explain the phenomenon. This example suggests slot tagging requires not only understanding the complex linguistic context constraints but also reasoning explicit lexical relations via large-scale background knowledge graphs.

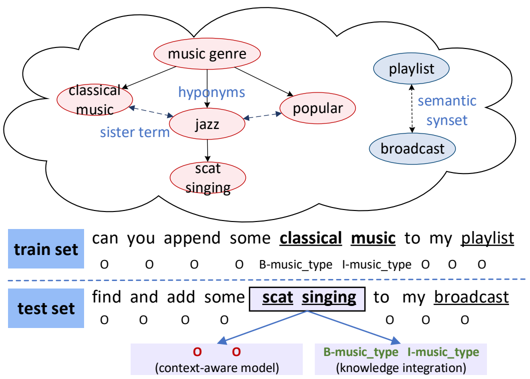

Figure 1: An example of slot tagging in the few-shot scenario where *scat singing* is unseen in the training set. The prior context-aware model fails to recognize its correct type because of low-coverage contextual information. After integrating background knowledge from WordNet, it succeeds to reason the correct type via lexical relations.

Previous state-of-the-art context-aware models (Goo et al., 2018; Haihong et al., 2019) only learn contextual information based on a multi-layer BiLSTM encoder and self-attention layer. (Dugas and Nichols, 2016; Williams, 2019; Shah et al., 2019) use handcrafted lexicons (also known as gazettes or dictionaries), which are typically collections of phrases semantically related, to improve slot tagging. One major limitation is that lexicons collected by domain experts are relatively small on the scale and fail to model complicated relations

[*] Weiran Xu is the corresponding author.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 619–624
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

between words, such as relation hierarchy.

In this paper, we propose a novel knowledge-enhanced method for slot tagging by integrating contextual representation of input text and the large-scale lexical background knowledge, enabling the model to reason explicit lexical relations. We aim to leverage both linguistic regularities covered by deep LMs and high-quality knowledge derived from curated KBs. Consequently, our model could infer rare and unseen words in the test dataset by incorporating contextual semantics learned from the training dataset and lexical relations from ontology. As depicted in Fig 2, given an input sequence, we first retrieve potentially relevant KB entities and encode them into distributed representations that describe global graph-structured information. Then we employ a BERT (Devlin et al., 2019) encoder layer to capture context-aware representations of the sequence and attend to the KB embeddings using multi-level graph attention. Finally, we integrate BERT embeddings and the desired KB embeddings to predict the slot type. Our main contributions are three-fold: (1) We investigate and demonstrate the feasibility of applying lexical ontology to facilitate recognizing OOV words in the few-shot scenario. To the best of our knowledge, this is the first to consider the large-scale background knowledge for enhancing context-aware slot tagging models. (2) We propose a knowledge integration mechanism and use multi-level graph attention to model explicit lexical relations. (3) Plenty of experiments on two benchmark datasets show that our proposed method achieves consistently better performance than various state-of-the-art context-aware methods.

2 Our Approach

In this work, we consider the slot tagging task in the few-shot scenario, especially for OOV tokens. Given a sequence with n tokens $X = \{x_i\}_{i=1}^{n}$, our goal is to predict a corresponding tagging sequence $Y = \{y_i\}_{i=1}^{n}$. This section first explains our BERT-based model and then introduces the proposed knowledge integration mechanism for inducing background commonsense. The overall model architecture is illustrated in Fig 2.

2.1 BERT-Based Model for Slot Tagging

The model architecture of BERT is a multi-layer bidirectional Transformer encoder. The input representation is a concatenation of WordPiece embeddings (Wu et al., 2016), positional embeddings, and the segment embeddings.

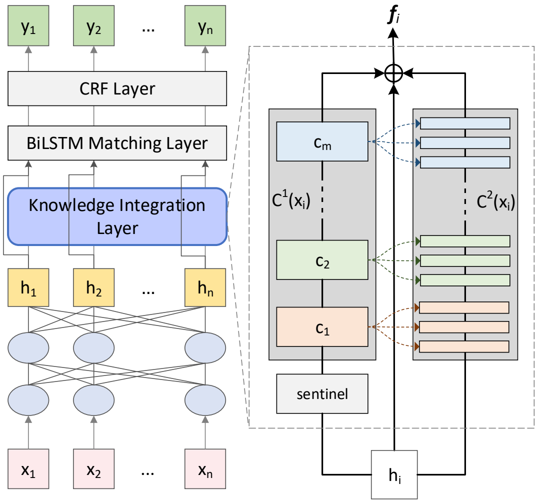

Figure 2: The overall architecture of the proposed slot tagging model.

Inspired by previous RNN-based works (Mesnil et al., 2014; Liu and Lane, 2016), we extend BERT to a slot tagging model. We first feed the input sequence $X = \{x_i\}_{i=1}^{n}$ to a pre-trained BERT encoding layer and then get final hidden states $H = (h_1, ..., h_n)$. To make this procedure compatible with the original BERT tokenization, we feed each input word into a WordPiece tokenizer and use the hidden state corresponding to the first sub-word as input to the softmax classifier.

$$y_i = \text{softmax}\left(\mathbf{W}h_i + b\right), i \in 1 \ldots n \quad (1)$$

where $h_i \in \mathcal{R}^{d_1}$ is the hidden state corresponding to the first sub-word of the i-th input word x_i and y_i is the slot label.

2.2 Knowledge Integration Mechanism

The knowledge integration mechanism aims at enhancing the deep contextual representation of input text via leveraging the large-scale lexical background knowledge, Wordnet (Miller, 1995), to recognize unseen tokens in the training set. Essentially, it applies multi-level graph attention to KB embeddings with the BERT representations from the previous layer to enhance the contextual BERT embeddings with human-curated background knowledge.

We first introduce the KB embedding and retrieval process. In this paper, we use the lexical KB, WordNet, stored as *(subject, relation, object)* triples, where each triple indicates a specific relation between word synsets, e.g., *(state, hypernym-*

of, california). Each synset expresses a distinct concept, organized by a human-curated tree hierarchy.

KB Embeddings We represent KB concepts as continuous vectors in this paper. The goal is that the KB tuples (s, r, o) can be measured in the dense vector space based on the embeddings. We adopt the BILINEAR model (Yang et al., 2014) which measures the relevance via a bilinear function: $f(\mathbf{s}, \mathbf{r}, \mathbf{o}) = \mathbf{s}^T \mathbf{M}_r \mathbf{o}$, where $\mathbf{s}, \mathbf{o} \in \mathcal{R}^{d_2}$ are the vector embeddings for s, o respectively and and $\mathbf{M}_r$ is a relation-specific embedding matrix. Then we train the embeddings using the max-margin ranking objective:

$$\sum_{q=(s,r,o)\in\mathcal{T}} \sum_{q'=(s,r,o')\in\mathcal{T}'} \max\left\{0, 1 - S_q + S_{q'}\right\} \tag{2}$$

where $\mathcal{T}$ denotes the set of triples in the KB and $\mathcal{T}'$ denotes the negative triples that are not observed in the KB. Finally we can acquire vector representations for concepts of the KB. Because we mainly focus on the slot tagging task, and the datasets are relatively small for joint learning KB embeddings. Furthermore, the KB contains many triplets not present in the ATIS and Snips dataset. Therefore we pre-train the KB vectors and keep them fixed while training the whole model to reduce the complexity.

KB Concepts Retrieval We need to retrieve all the concepts or synsets relevant to the input word x_i from the KB. Different from (Yang and Mitchell, 2017; Yang et al., 2019), for a word x_i, we first return its synsets as the first-level candidate set $C^1(x_i)$ of KB concepts. Then we construct the second-level candidate set $C^2(x_i)$ by retrieving all the direct hyponyms of each synset in $C^1(x_i)$, as shown in the right part of Fig 2.

Multi-Level Graph Attention After obtaining the two-level concept candidate sets, we apply the BERT embedding $\mathbf{h}_i$ of input token x_i to attending over the multi-level memory. The first-level attention, α, is calculated by a bilinear operation between $\mathbf{h}_i$ and each synset $\mathbf{c}_j$ in the first level set $C^1(x_i)$:

$$\alpha_{ij} \propto exp(\mathbf{c}_j^T \mathbf{W}_1 \mathbf{h}_i) \tag{3}$$

Then we add an additional sentinel vector $\bar{\mathbf{c}}$ (Yang and Mitchell, 2017) and accumulate all the embeddings as follows:

$$\mathbf{s}_i^1 = \sum_j \alpha_{ij} \mathbf{c}_j + \gamma_i \bar{\mathbf{c}} \tag{4}$$

	ATIS	Snips
Vocabulary Size	722	11,241
Percentage of OOV words	0.77%	5.95%
Number of Slots	120	72
Training Set Size	4,478	13,084
Development Set Size	500	700
Testing Set Size	893	700

Table 1: Statistics of ATIS and Snips datasets.

where γ_i is similar to α_{ij} and $\sum_j \alpha_{ij} + \gamma_i = 1$. Here $\mathbf{s}_i^1$ is regarded as a one-hop knowledge state vector for it only represents its directly linked synsets. Therefore, we perform the second-level graph attention to encode the hyponyms of its direct synsets to enrich the information of original synsets. Intuitively the second-level attention over the hyponyms can be viewed as a relational reasoning process. Because once a synset belongs to an entity type, its hyponyms always conform to the same type. Likewise, the second-level attention over $C^2(x_i)$ is calculated:

$$\beta_{ijk} \propto exp(\mathbf{c}_{jk}^T \mathbf{W}_2 \mathbf{h}_i) \tag{5}$$

where $\mathbf{c}_j$ is the j-th synset linked to token x_i and $\mathbf{c}_{jk}$ the k-th hyponym of $\mathbf{c}_j$. So we can obtain the multi-hop knowledge state vector $\mathbf{s}_i^2$:

$$\mathbf{s}_i^2 = \sum_j \sum_k \alpha_{ij} \beta_{ijk} \mathbf{c}_{jk} \tag{6}$$

Then we concat multi-level knowledge-aware vector $\mathbf{s}_i^1, \mathbf{s}_i^2$, and original BERT representation $\mathbf{h}_i$, and output $\mathbf{f}_i = [\mathbf{s}_i^1, \mathbf{s}_i^2, \mathbf{h}_i]$.

We also add a BiLSTM matching layer which takes as input the knowledge-enriched representations $\mathbf{f}_i$. Then we forward the hidden states to a CRF layer and predict the final results. The training objective is the sum of log-likelihood of all the words.

3 Experiments

3.1 Setup

Datasets To evaluate our approach, we conduct experiments on two public benchmark datasets, ATIS (Tür et al., 2010) and Snips (Coucke et al., 2018). ATIS contains 4,478 utterances in the training set and 893 utterances in the test set, while Snips contains 13,084 and 700 utterances, respectively. The percentage of OOV words between the training and test datasets is 0.77%(ATIS) and 5.95%(Snips).

Model	**ATIS**						**Snips**					
	1%	2%	5%	10%	50%	100%	1%	2%	5%	10%	50%	100%
Attention-Based	3.59	22.91	48.16	63.33	88.51	94.21	20.94	30.58	43.74	50.92	78.46	87.80
Slot-Gated Full	4.91	20.08	53.01	77.07	94.19	94.80	18.24	25.03	51.91	64.51	84.45	88.88
Slot-Gated Intent	3.45	18.81	55.64	79.59	94.53	95.20	22.88	30.71	57.94	69.43	83.80	88.30
SF-ID Network	6.18	18.89	63.96	83.35	94.34	95.80	19.25	31.50	55.87	69.65	86.01	92.23
RNN	5.86	21.27	62.53	80.59	94.42	95.17	19.92	25.91	56.30	65.88	88.65	89.30
RNN+KB	6.75	23.35	63.55	81.40	95.04	**95.63**	23.64	28.92	58.88	68.22	90.40	**90.81**
BERT	73.67	80.84	88.09	91.06	95.08	95.98	69.49	76.87	86.34	90.01	94.26	95.17
BERT+KB	74.71	81.70	88.81	91.55	95.39	**96.25**	71.50	78.65	87.84	91.24	95.43	**95.89**

Table 2: Slot tagging performance on ATIS and Snips datasets. % represents how much training data we randomly choose from the original training set. We report the F1 scores on the same test sets.

Samples in Snips are from different topics, such as getting weather and booking a restaurant, resulting in a larger vocabulary. By contrast, samples in ATIS are all about flight information with similar vocabularies across them. Therefore, Snips is much more complicated, mainly due to data diversity and the large vocabulary. The full statistics are shown in the Table 1.

To simulate the few-shot scenarios, we downsample the original training sets of ATIS and Snips to different extents while keeping valid and test sets fixed. We aim to evaluate the effectiveness of integrating external KB under the settings of varied sizes of training data available.

Evaluation We evaluate the performance of slot tagging using the F1 score metric. In the experiments, we use the English uncased BERT-base model, which has 12 layers, 768 hidden states, and 12 heads. The hidden size for the BiLSTM layer is set to 128. Adam (Kingma and Ba, 2014) is used for optimization with an initial learning rate of 1e-5. The dropout probability is 0.1, and the batch size is 64. We finetune all hyperparameters on the valid set.

3.2 Baselines

Attention-Based (Liu and Lane, 2016) uses an RNN layer and a self-attention layer to encode the input text. Slot-Gated (Goo et al., 2018), which has two variants, *Full Atten* and *Intent Atten*, applies the information of intent detection task to enhance slot tagging. SF-ID Network (Haihong et al., 2019) designs a multiple iteration mechanism to construct bi-directional interrelated connections between slot tagging and intent detection. Most of the previous methods consider improving the performance of slot tagging by joint learning with intent detection. However, the effectiveness of background knowledge for slot tagging is still unexplored. Consequently, our proposed approach intends to integrate the large-scale lexical background knowledge, WordNet, to enhance the deep contextual representation of input text. We hope to further improve the performance of slot tagging, especially in the few-shot scenario where there is no plenty of training data available. [1]

3.3 Overall Results

We display the experiment results in Table 2, where we choose two model architectures RNN and BERT as the encoding layer. Table 2 shows that our proposed knowledge integration mechanism significantly outperforms the baselines for both datasets, demonstrating that explicitly integrating the large-scale background knowledge and contextual representation can benefit slot tagging effectively. Moreover, the improvement of 0.72% over strong baseline BERT on Snips is considerably higher than 0.27% on ATIS. Considering the distinct complexity of the two datasets, the probable reason is that a simpler slot tagging task, such as ATIS, does not require much background knowledge to achieve good results. Because the vocabulary of ATIS is extremely smaller than that of Snips, therefore the context-aware models are capable of providing enough cues for recognizing rare or OOV words. Hence, our method makes a notable difference in a scenario where samples are linguistically diverse, and large vocab exists. The results also demonstrate that incorporating external knowledge will not bring in much noise since we use a knowledge sentinel for the better tradeoff between the impact of background knowledge and information from the context.

On the other hand, the main results of the

[1] We do not choose (Williams, 2019) as a baseline since it only performs experiments on private industrial datasets and does not open source. We can hardly figure out the details of manually collecting lexicons from the dataset.

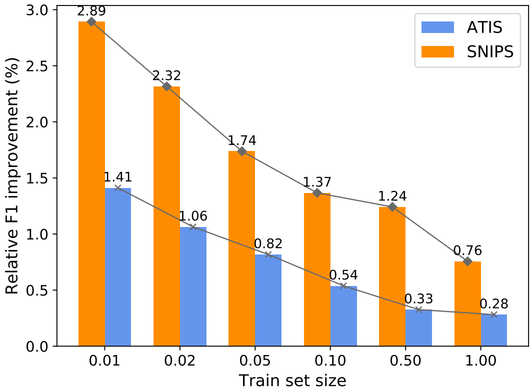

Figure 3: Relative F1 improvement over BERT baseline under the different sizes of training data.

RNN-based models are 95.17(+0.46) on ATIS and 89.30(+1.51) on Snips, where the scores in the brackets are the absolute improvements arisen by KB. Compared to the BERT-based models, 95.98(+0.27) on ATIS and 95.17(+0.72) on Snips, the RNN-based model achieves more significant improvements in BERT-based models. We believe BERT can effectively transfer prior linguistic context constraints, so that background knowledge benefits RNN-based models more. BERT does improve the model's ability to solve the OOV problem since it has learned linguistic knowledge from the large corpus. However, our method focuses more on the effect of using human-curated structured background knowledge and further enhances BERT in a distinct way.

4 Qualitative Analysis

4.1 Effect of Training Data Size

Fig 3 shows the relative improvement percentages on ATIS and Snips using different sizes of training data. Results substantiate knowledge integration better facilitates few-shot slot tagging. This is because traditional context-aware models can not learn enough contextual semantics well while only given several samples. Explicit lexical relations become essentially necessary when there is not adequate training data, especially for rare words or OOV words. Background KB enables the model to reason explicit lexical relations and helps recognize rare and unseen words. Meanwhile, incorporating background knowledge can also enhance the original representation of BERT, which can provide direct lexical relations.

Model	ATIS	Snips
Full Model	91.55	91.24
- w/o knowledge integration	91.20	90.22
- w/o the second-level graph attention	91.46	90.87
- w/o matching layer	91.42	91.05
- w/o CRF	91.38	90.96

Table 3: Ablation analysis under the 10% training data setting.

4.2 Ablation Study

To study the effect of each component of our method, we conduct ablation analysis under the 10% training data setting (Table 3). We can see that knowledge integration is crucial to the improvements. Besides, the first-level graph attention acquires better performance gain than the second-level attention. We assume that directly linked synsets are more significant than the hyponyms. The matching layer and CRF also play a role. The reason why the RNN matching layer matters is partly to build explicit interactions between knowledge vectors and context vectors.

5 Conclusion

We present a novel knowledge integration mechanism of incorporating background KB and deep contextual representations to facilitate the few-shot slot tagging task. Experiments confirm the effectiveness of modeling explicit lexical relations, which has not yet been explored by previous works. Moreover, we find that our method delivers more benefits to data scarcity scenarios. We hope to provide new guidance for the future slot tagging work.

Acknowledgments

The authors would like to thank the reviewers for their valuable comments. This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC ”Artifical Intelligence” Project No. MCM20190701.

References

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Fabrice Dugas and Eric Nichols. 2016. Deepnnner: Applying blstm-cnns and extended lexicons to named entity recognition in tweets. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 178–187.

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.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471.

Keqing He, Weiran Xu, and Yuanmeng Yan. 2020. Multi-level cross-lingual transfer learning with language shared and specific knowledge for spoken language understanding. *IEEE Access*, 8:29407–29416.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Bing Liu and Ian Lane. 2015. Recurrent neural network structured output prediction for spoken language understanding. In *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*.

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

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.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Avik Ray, Yilin Shen, and Hongxia Jin. 2018. Robust spoken language understanding via paraphrasing. *Interspeech 2018*.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

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

Kyle Williams. 2019. Neural lexicons for slot tagging in spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 83–89.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *ACL*.

Bishan Yang and Tom M. Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *ACL*.

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