规则知识与神经网络的融合
Integration of Rules into Neural Networks

刘群 LIU Qun

华为诺亚方舟实验室 Huawei Noah's Ark Lab

全球人工智能技术大会自然语言处理论坛 GAITC 2022 NLP Forum

2022-11-27

NOAH'S ARK LAB

HUAWEI

Content

Content

Background

- Neural network based deep learning methods obtain great succsss in almost all NLP tasks.
  基于神经网络的深度学习方法在几乎所有NLP任务中都取得了重大成功。
- But in many real industrial applications, rule-based methods are still unavoidable because:
  但在很多真实的工业级应用中，基于规则的方法仍然是无可取代的，因为：
  - NNs are not interpretable, thus not reliable.
    神经网络无法解释，因而不可信赖。
  - NNs are not controlable. Users are not able to change the behaviour of NNs according to their expertise.
    神经网络无法控制。用户无法根据自己的专家经验去改变神经网络的行为。
- Rule-based systems have their own shortcomings:
  基于规则的系统有自身的缺点：
  - Low coverage. 覆盖率低。
  - Low efficiency. 执行效率低。

Our work

- Our methods: 我们的方法：
  - Integrate rules into neural networks. 将规则融入神经网络。
  - Convert rule-based systems to neural networks. 将规则系统编程神经网络系统。
  - Execute rules via neural networks. 通过神经网络来执行规则。
- More specifically, we proposed methods to integrate the following three types of rules into neural networks:
  具体来说，我们提出不同方法，将以下三种形式的规则融入到神经网络中：
  - large-scale dictionaries. 大规模词典。
  - regular expressions. 正则表达式。
  - context-free rules. 上下文无关规则。

Content

Background

- Lexicons are very important

Please open the IronMan

IronMan may be an app, a device, or a video which your smart assistant haven't seen

play Just a Little While Longer now on IronMan

Your smart assistant can't distinguish whether Just a Little While Longer is a song name or a movie name

Background

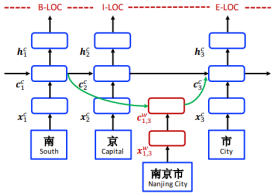
Previous method

Feature based

Figure 2: Example of feature vector construction. The character with the red shadow is the character x_i. The character segments with rounded rectangle are the words in the dictionary D.

Zhang, Qi, Xiaoyu Liu, and Jinlan Fu. "Neural networks incorporating dictionaries for chinese word segmentation." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

Lattice based



(c) Lattice model.

Zhang, Yue, and Jie Yang. "Chinese ner using lattice lstm." arXiv preprint arXiv:1805.02023 (2018).

Graph based

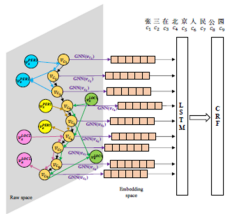


Figure 2: System architecture

Ding, Ruixue, et al. "A neural multi-digraph model for Chinese NER with gazetteers." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

Weakness

- Some depends on word embedding
- Ignore the category of words
- No explicit disambiguation

DyLex: Incorporating Dynamic Lexicons into BERT for Sequence Labeling

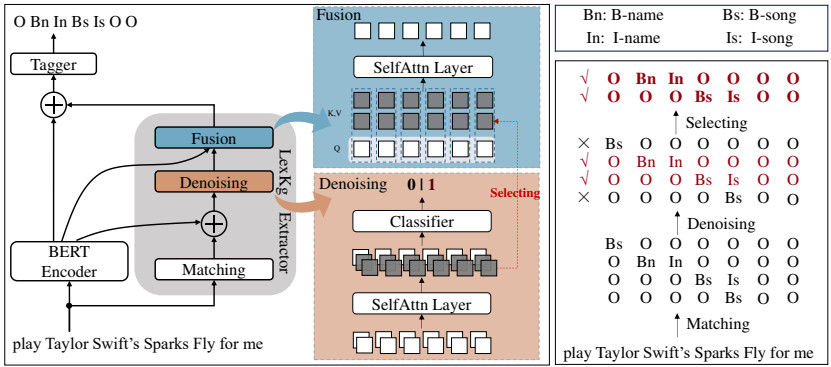


Figure 2: (a) The overall architecture of the proposed DyLex framework, it consists of two parts, namely BERT-based sequence tagger and LexKg Extractor. The Extractor has three submodules: the Matching, the Denoising and the Fusing. (b) A concrete example of lexicon matching and denoising.

DyLex: Incorporating Dynamic Lexicons into BERT for Sequence Labeling

- We propose a general framework for effectively introducing external lexical knowledge into sequence labeling tasks.
- Our framework supports dynamic updates of lexicons to facilitate industrial deployment.
  - We use word-agnostic tag embeddings instead of pre-trained word embeddings.
- We devise a novel knowledge denoising module to make full use of large-scale lexicons.

Result

The CWS task

Primary Baselines

- BERT-based Sequence Tagger
- Glyce (for CWS)
- FLAT (for Chinese NER)
- HSCRF+softdict (for English NER)

Lexicon

- the lexical entry contains item and category
- lexicon used in experiment consistent with the previous related work

Task	Item	Category	Tag
CWS	words	-	B: Begining of a word I: Continuation of a word
NER	words	Song name	B-song: Begining of a song name I-song: Continuation of a song name

南京/长江大桥/建成/于/1968年

The Nanjing Yangtze River Bridge was completed in 1968.

南京
南京市
长江
长江大桥
……

the location dictionary

Model	LEX	PKU	CITYU
Yang et al. (2017a)	✗	96.30	96.94
Ma et al. (2018)	✗	96.10	97.23
Huang et al. (2020a)	✗	96.60	97.60
BERT (Devlin et al., 2019)	✗	96.50	97.60
Glyce (Meng et al., 2019)	✓	96.70	97.90
DyLex	✓	**97.14**	**98.60**

Result

The NER task

[Jim]$_{\text{Person}}$ bought 300 shares of [Acme Corp.]$_{\text{Organization}}$ in [2006]$_{\text{Time}}$.

- Experiment both on Chinese and English NER
- Chinese
  - Weibo, MSRA, Resume, OntoNotes
  - used lexicon is the same is *Li et al.(2020)*
- English
  - Conll2003, OntoNotes5.0
  - used lexicon is the same as *Liu et al.(2019a)*

Methods	LEX	Weibo	MSRA	Resume	Ontonotes	AVG
BiLSTM-CRF (Huang et al., 2015)	✗	56.75	91.87	94.41	71.81	78.71
TENER (Yan et al., 2019)	✗	58.39	93.01	95.25	72.82	79.86
BERT (Devlin et al., 2019)	✗	68.20	94.95	95.53	80.14	84.70
LSTM+ExSoftWord (Ma et al., 2020)	✓	56.02	92.38	95.43	72.40	79.05
Lattice-LSTM (Zhang and Yang, 2018)	✓	58.79	93.18	94.46	73.88	80.07
LR-CNN (Gui et al., 2019a)	✓	59.92	93.71	95.11	74.45	80.79
FLAT+BERT+CRF (Li et al., 2020)	✓	68.55	96.09	95.86	**81.82**	85.58
DyLex	✓	**71.12**	**96.49**	**95.99**	81.48	**86.27**

Table 2: F1 scores of different methods on Chinese NER dataset. AVG stands for the average of each row.

Methods	LEX	Conll2003	OntoNotes5.0	AVG
BiLSTM-CRF (Huang et al., 2015)	✗	91.03	86.28	88.65
TENER (Yan et al., 2019)	✗	91.33	88.43	89.88
LSTM-CNNs (Chiu and Nichols, 2016)	✗	91.62	86.28	88.95
BERT (Devlin et al., 2019)	✗	92.40	89.13	90.76
CSE (Akbik et al., 2018)	✗	92.72	89.71	91.40
SENNA (Collobert et al., 2011)	✓	89.56	-	-
JERL (Luo et al., 2015)	✓	91.20	-	-
ID-CNN (Strubell et al., 2017)	✓	90.54	86.84	88.69
GRN (Chen et al., 2019a)	✓	91.44	87.67	89.55
HSCRF (Liu et al., 2019a)	✓	92.75	89.94	91.34
LUKE (Yamada et al., 2020)	✓	94.30	-	-
DyLex	✓	**94.30**	**90.19**	**92.25**

Table 3: F1 scores of different methods on English NER dataset. The setting is the same with Table2.Note that LUKE incorporate the entity information during the pre-training phase.

Result

The NLU task

xiaoyi result

MODELS	TEST		SINGLE		MULTI		MEDIA		DISAMB
	intent	slot	intent	slot	intent	slot	intent	slot	
BERT	96.67	95.12	13.83	54.66	77.13	81.22	95.46	92.88	-
DyLex	**97.43**	**96.65**	**77.81**	**92.10**	**90.89**	**93.03**	**95.96**	**95.09**	**97.74**

Table 5: Performance on the industrial dataset (F1). The TEST set is divided into three parts, SINGLE, MULTI, and MEDIA. The slot in SINGLE can only correspond to one tag in lexicon, and the one in MULTI can correspond to multiple tag. The sentence in MEDIA has obvious indicator words, such as words like "play music".

Models	LEX	Snips			ATIS			AVG
		Intent	Slot	$match_{sen}$	Intent	Slot	$match_{sen}$	
Atten-joint (Liu and Lane, 2016)	✗	96.7	87.8	74.1	91.1	94.2	78.9	87.13
Slot-Gated (Goo et al., 2018)	✗	97.0	88.8	75.5	94.1	95.2	82.6	88.86
SF-ID (E et al., 2019)	✗	97.4	92.2	80.5	97.7	95.8	86.7	91.71
Joint BERT (Chen et al., 2019b)	✗	98.6	97.0	92.8	97.5	**96.1**	88.2	95.03
HSCRF* (Liu et al., 2019a)	✓	98.7	97.6	93.1	97.7	96.0	88.4	95.25
DyLex	✓	**99.8**	**99.1**	**98.1**	**98.2**	95.7	**88.5**	**96.52**

Table 6: NLU performance on Snips and ATIS datasets. The metrics are intent classification accuracy, slot filling F1, and sentence-level semantic frame accuracy (%). The results marked with * are reported from our recurrence.

Two scenarios

- Industrial dataset
  To evaluate the method with large scale lexicon, the size of lexicon is 16M
- Public dataset
  used common nlu datasets, Snips and ATIS

Content

Content

Background

- Regular Expressions (REs) are broadly used in various NLP scenarios 正则表达式在各种NLP场合广泛应用
  - Named Entity Recognition (NER) 命名实体识别
  - Data Cleaning 数据清洗
  - Sensitive content filtering 敏感内容过滤
  - Dialog Intent Identification and Slot Filling 对话意图识别和槽位提取
- Strength and Weakness 优缺点
  - No need of large scale training data 无需大量人工标注的训练数据
  - Easy to read, write and edit for human experts 易于理解，方便人工撰写和修改
  - Hard to balance between precision and recall 很难在准确率和召回率之间把握平衡
    - low coverage while not using wildcards 不使用通配符容易导致低覆盖率
    - too general while using wildcards too much 使用很多通配符会导致过于宽泛
  - low efficiency when too many REs 正则表达式太多会导致执行效率大大降低

Contribution of this work

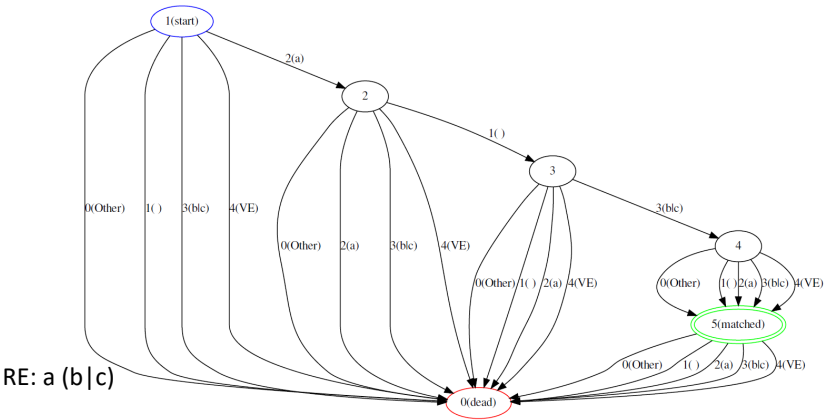
- A method to convert a set of REs to a Neural Network (namely DFA-NN)
  提出了一种方法可以将一组正则表达式转化为一个神经网络（称为DFA-NN）
- The DFA-NN is much more efficicent than a RE Recognizer
  DFA-NN的执行效率远高于正则表达式识别器
- The DFA-NN provides a sort of soft matching ability
  DFA-NN提供了某种软匹配的能力
- The DFA-NN can be trained with both REs and training examples
  DFA-NN可以同时使用正则表达式规则和训练样例进行训练
- The DFA-NN performs better than the simple classifier NN, especially when the training instances are few
  DFA-NN在训练数据很少的时候表现远好于简单的NN分类器。

Content

DFA-NN: Integrating REs with NNs via DFA

- Convert an RE or a set of REs into a Minimal DFA (MDFA).
  将一个或者一组正则表达式转换为一个最小确定性有限状态自动机（MDFA）。
- For each instance:
  - We send the input sequence to the DFA, and get a DFA state for each token in the sequence.
    将输入序列送到DFA中，对于序列中的每一个token，我们得到一个DFA状态。
  - We can also assign a sequence classification (True/False) depending on whether the final DFA state of the sequence is an end state. It can also be a multi-class label if there are multiple REs.
    根据最后的DFA状态是否为一个终止状态，我们也可以给整个序列一个标记（是/否）。如果有多个正则表达式，也可以使用多分类标签。

DFA-NN: Integrating REs with NNs via DFA



a b c ⇒ [“a”, ” _“, “b”, ” _“, “c”] ⇒ [1,2,3,4,5,0]

text — symbol sequence — DFA state sequence

DFA-NN: Integrating REs with NNs via DFA

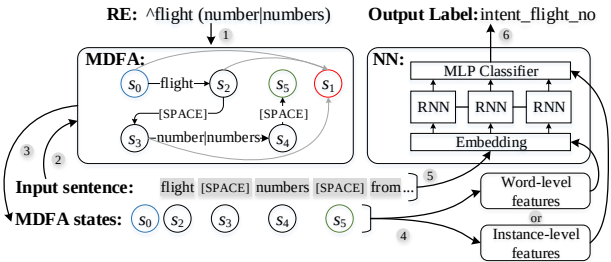
- Then we train an BiLSTM-based classifier based on the training data with token-level and instance-level labels.
  然后我们使用这些带token级和instance级标记的训练数据来训练一个基于BiLSTM的分类器。
  - The sequences are fed to the BiLSTM and we obtain a hidden state vector for each token.
    把每个序列送到BiLSTM中，我们可以给序列中的每个token赋予一个隐状态向量。
  - Then we train a small multi-layer perceptron (MLP) to predict the DFA state for each token using its hidden state vector.
    然后我们训练一个小的多层感知机网络（MLP）来通过每个token对应的隐状态向量来其对应的DFA状态。
    (...To be continued 续下页)

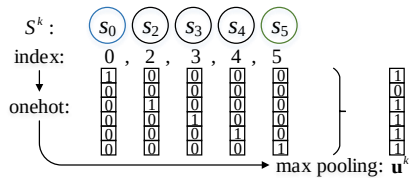
DFA-NN: Integrating REs with NNs via DFA

- Then we train an BiLSTM-based classifier based on the training data with token-level and instance-level labels.
  然后我们使用这些带token级和instance级标记的训练数据来训练一个基于BiLSTM的分类器。

  (...Continued 接上页)
  - We pool the obtained DFA states of all the tokens in the sequence to obtained an instance-level DFA state vector.
    我们将预测得到的所有token的DFA状态通过池化操作得到一个instance级的DFA状态向量。
  - Finally we train another MLP to predict the instance-level label according to its instance-level state vector and all the token-level hidden state vectors.
    最后我们训练另一个多层感知机网络，根据instance级的DFA状态向量和token级的所有隐状态向量，来预测整个sequence的类别标记。

DFA-NN: Integrating REs with NNs via DFA





Content

Experiments and Results

We employ the widely used ATIS intent classification dataset[2]. This dataset contains 4,978 training samples and 893 test samples with 18 intent labels. There are 54 manually written REs for intent classification obtained from Luo et al. (2018)[3]. We build few-shot training sets by randomly selecting q samples for each class from the full training set. The models are trained on each training set respectively and evaluated on the official test set.

Model	q=5	q=10	q=20	q=5 w/300	q=10 w/300	q=20 w/300
MLP-I*	63.72	73.46	83.20	94.23	94.90	95.71
MLP-O*	58.68	77.83	89.25	91.82	97.09	97.15
ATTN (luo et al.)	75.36	85.44	88.80	92.05	96.98	97.76
FOL (hu et al.)	56.22	68.42	84.10	91.94	96.75	97.42
NNSC (ours)	65.28	73.90	86.89	94.06	96.97	98.20
INSTANCE (ours)	**81.97**	84.99	**91.04**	94.51	**97.64**	97.87
WORD (ours)	80.17	**86.67**	90.25	**94.62**	97.20	**98.32**

Table 1: Comparisons with the other methods that combine REs with NN without MDFA and the NN-based Model (denoted as NNSC). * indicates the baseline hybrid models described in (Luo et al., 2018). The results are reported in accuracy.

Experiments and Results

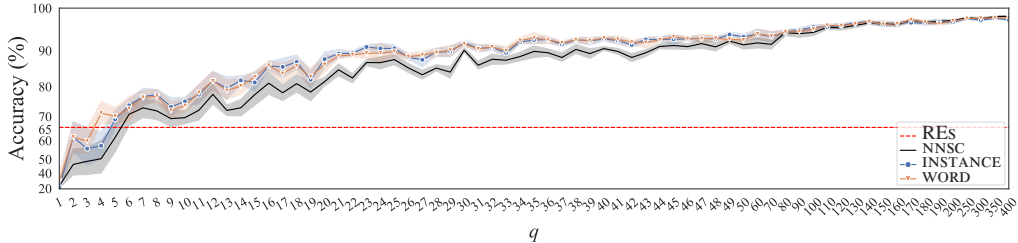


Figure 3: Comparisons of the REs, NN-based model (denoted as NNSC) and the proposed hybrid models (denoted as INSTANCE and WORD).

Content

Content

Rule-based engine for Dialog Systems

- Although deep learning has been great successful in NLP, rule-based dialog engines are still broadly used in industries，because:

  虽然深度学习在NLP中取得了巨大成功，但工业界还在普遍使用基于规则的对话引擎，因为：

  - It is hard to encode complex business logic within a NN.
    使用神经网络来实现复杂的业务逻辑非常困难。
  - It is hard to change the business logic given an existings NN.
    修改已有神经网络中的业务逻辑非常困难。

- A rule-based dialog engine has further advantages because its hierarchical representation:

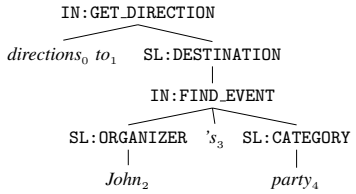
  由于采用层次表示，基于规则的对话引擎还有更多优点：

  - Support multi-intent utterances.
    支持多意图表达。
  - Support negative expressions.
    支持否定表达。

Rule-based engine for Dialog Systems

- However, compared to NNs, rule-based dialog engines have some disadvantages:
  然而，相对于神经网络，基于规则的对话引擎有以下劣势：
  - Fragility. Possibly fail when the users change their expressions slightly.
    脆弱性。用户略微改变其表达方式可能就无法识别。
  - Low efficiency, especially when the rule-base is very large.
    效率低，尤其当规则数量非常的时候。

Span-based Hierarchical Semantic Parsing for TOD



$$
\begin{aligned}
T = \{&(0,5) : (\texttt{IN:GET\_DIRECTION}),\\
&(2,5) : (\texttt{SL:DESTINATION}, \texttt{IN:FIND\_EVENT}),\\
&(2,3) : (\texttt{SL:ORGANIZER}),\\
&(4,5) : (\texttt{SL:CATEGORY}),\\
&(0,1) : \varnothing, \quad (0,2) : \varnothing, \quad (0,3) : \varnothing, \quad \dots \}
\end{aligned}
$$

Figure 1: An example TOP tree and its mapping representation T. (`IN:` = intent; `SL:` = slot)

Method	Acc	F1	Time
seq2seq	78.24	90.78	8m
RNNG	80.63	92.61	1h 16m
Stern-chart	80.66	93.03	25m
Stern-greedy	80.79	92.83	22m
ours (no f_e)	80.80	93.35	**5m**
ours (+ f_e)	**81.80**	**93.63**	9m

Table 1: The test exact match accuracy, labeled bracket F1, and training time per epoch of different methods.

Pasupat et al., Span-based Hierarchical Semantic Parsing for Task-Oriented Dialog, EMNLP2019

Span-based Hierarchical Semantic Parsing for TOD

- This method outperformed existing methods, while have a high training and inference speed as fast as seq2seq model.
  这种方法比已有方法获得了更好的性能，同时具有跟seq2seq模型快的训练和推理速度。
- However, we see there are still drawbasks as:
  不过，这种方法还有一些缺点：
  - There are still space to improve the accuracy.
    识别准确率还有待继续提高。
  - The performance will drop when the lexicon is updated.
    词典更新会导致性能下降。

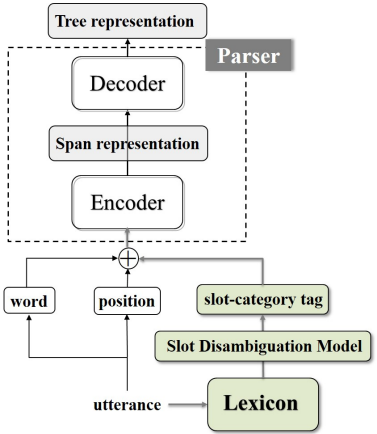
Pasupat et al., Span-based Hierarchical Semantic Parsing for Task-Oriented Dialog, EMNLP2019

Content

Our Method: Lexicon-injected Semantic Parsing for TOD

- Our method improves the model proposed by (Pasupat et al., 2019) in the following ways:
  我们对(Pasupat et al., 2019)的模型进行了以下改进：
  - We improve the performance of the span-based parser by incorporating a splitting feature into the span representation.
    我们在Span表示中引入了断点特征，从而提高了Span-based分析器的性能。
  - We propose a novel lexicon-injected method and a slot disambiguation method to further improve the semantic parsing performance, to ensure the parser can keep a high performance when updating the lexicon.
    我们提出了一种新颖的词典注入方法和槽位排歧方法，可以进一步改进语义分析的性能，并确保在词典更新后语义分析器的性能不会下降。

Our Method: Lexicon-injected Semantic Parsing for TOD



Model structure

We define the representation of span (i, j) as:

$$r_{i,j} = f_{j,j+1} - f_{i-1,i}$$

We define a new *span-splitting representation* $\hat{r}_{i,j}$, that adds the boundary representation at the splitting point to the parent:

$$\hat{r}_{i,j} = r_{i,j} + f_{k^*-1,k^*} \tag{6}$$

where k^* is the best splitting point dynamically computed as follows, dividing parent span into two separate spans $(i, k-1)$ and (k, j):

$$k^* = \arg\max_k [s^*(i, k-1) + s^*(k, j)]$$

Incorporate the splitting feature into the span representation

Our Method: Lexicon-injected Semantic Parsing for TOD

- For each word appears in a slot with the category tag t_i, we assign t_i to the word in the lexicon. Thus each word may have multiple tags. We train an embedding q_i for each tag t_i which is initialized randomly.
- For those words which appear outside the slots, we define a special tag t_0.
- We concatenate the category embeddings to the previous defined word representation:

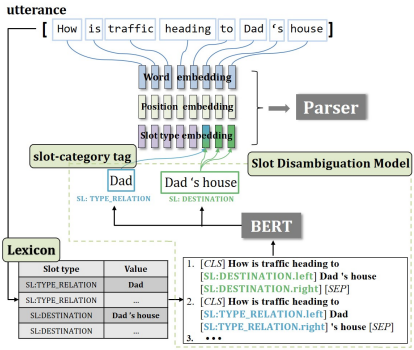
$$x_i = [w_i; p_i; q_i]$$

- We define a new generalized word representation by replacing the word embedding with its tag embedding:

$$x_i = \begin{cases} [w_i; p_i; q_i], & t_i = t_o \\ [q_i; p_i; q_i], & \text{otherwise.} \end{cases}$$

- Thus we can make the model more informative of the slot-categories rather the specific word, and ensure the model performance will remain when we update the lexicons with unseen entities in the training data.

Lexicon-injected Model



Slot Disambiguation

Content

Experiments and Results

Method	Acc	F1
Non-lexicon-injected parser:		
Pasupat	80.80	93.35
Pasupat-edge	81.80	93.63
Decoupled RoBERTa	84.52	-
Decoupled BART	87.10	-
Seq2SeqPtr (+BERT)	83.13	-
Seq2SeqPtr (+RoBERTa)	86.67	-
Ours (base)†	83.06	94.23
Ours (+Split)†	83.97	94.55
Ours (+RoBERTa)	85.77	95.24
Our lexicon-injected parser:		
w/o Slot Disambiguation†	81.83	93.87
w/ Slot Disambiguation†	85.63	96.13
w/ SD + GR†	86.80	96.34
w/ SD + GR + RoBERTa	**87.62**	**96.60**

Table 2: Comparison of complete match accuracy and labeled bracket F1 of different methods on TOP test set. SD, GR and † denote to slot disambiguation, generalized representation and use BERT-base model.

Method	Acc	F1
Non-lexicon-injected parser:		
Pasupat	70.90	-
Ours (+Regex substitution)†	69.94	-
Ours (+RoBERTa)	74.35	92.10
Seq2SeqPtr (+RoBERTa)	74.91	92.42
Using the original lexicon:		
w/ Slot Disambiguation†	71.75	93.23
w/ SD + GR†	70.95	93.19
w/ SD + GR + RoBERTa	71.87	93.49
w/ SD + RoBERTa	72.78	93.73
Altering the lexicon:		
w/ Slot Disambiguation†	84.80	95.92
w/ SD + GR†	86.30	96.25
w/ SD + GR + RoBERTa	**87.27**	**96.54**

Table 3: Comparison of complete match accuracy and labeled bracket F1 of different methods on the modified TOP test set with introducing unseen slot values. SD, GR and † denote to slot disambiguation, generalized representation and using BERT-base respectively.

Content

Summary

Thank you!

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI