DyLex: Incorporating Dynamic Lexicons into BERT for Sequence Labeling

Baojun Wang[1*], **Zhao Zhang**[2*], **Kun Xu**[2*], **Guang-Yuan Hao**[3], **Yuyang Zhang**[1]
Lifeng Shang[1], **Linlin Li**[2], **Xiao Chen**[1], **Xin Jiang**[1], **Qun Liu**[1]
[1]Huawei Noah's Ark Lab [2]Huawei Technologies Co., Ltd.
[3]The Hong Kong University of Science and Technology
{puking.w,zhangzhao54,xukun24,zhangyuyang4}@huawei.com
{Shang.Lifeng,lynn.lilinlin,chen.xiao2,Jiang.Xin,qun.liu}@huawei.com
guangyuanhao@outlook.com

Abstract

Incorporating lexical knowledge into deep learning models has been proved to be very effective for sequence labeling tasks. However, previous works commonly have difficulty dealing with large-scale dynamic lexicons which often cause excessive matching noise and problems of frequent updates. In this paper, we propose DyLex, a plug-in lexicon incorporation approach for BERT based sequence labeling tasks. Instead of leveraging embeddings of words in the lexicon as in conventional methods, we adopt *word-agnostic tag embeddings* to avoid re-training the representation while updating the lexicon. Moreover, we employ an effective supervised lexical knowledge denoising method to smooth out matching noise. Finally, we introduce a *col-wise attention* based knowledge fusion mechanism to guarantee the pluggability of the proposed framework. Experiments on ten datasets of three tasks show that the proposed framework achieves new SOTA, even with very large scale lexicons[1].

1 Introduction

Sequence labeling is the task of assigning categorical labels to a text sequence. Many conventional NLP tasks, such as named entity recognition (NER), Chinese word segmentation (CWS), and slot-filling based natural language understanding (NLU), can be formalized as the sequence labeling problem. The deep learning methods, especially the recently proposed BERT and its variants, have achieved great success in such sequence labeling tasks. However, the BERT-based methods are generally built based on word-piece or character embeddings. The word information (e.g., word *boundary* or *type*) is not fully exploited, which makes it difficult to accurately determine the entity boundary or correctly predict entity type.

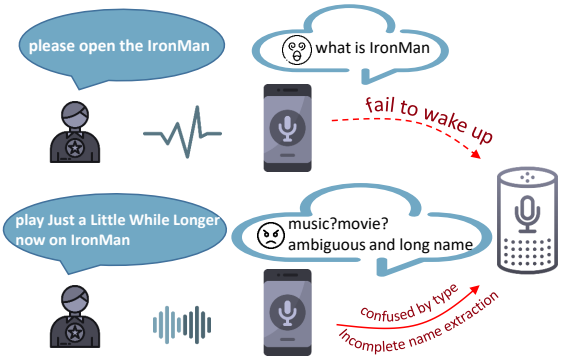


Figure 1: Iron Man can be a name of a smart device or a movie and the system would be unable to react properly upon "Please play Iron Man" from a user. Another case as "Play just a little while longer now on Iron Man" requires the system to classify "Play" between music and movie domains, and whether "now" should be combined with "just a little while longer" as a whole.

As shown in Figure 1, it is infeasible to understand user's utterance correctly without using deterministic domain knowledge that "*Iron Man*" is the alias of a Smart Speaker or "*just a little while longer*" is a famous song. In commercial systems, the lexicon is widely used as an effective way to store various domain knowledge. In practice, the size of a lexicon can range from ten to a few million, and we usually need to update the contents of lexicons frequently, which dramatically increases the difficulty of incorporating lexicons into deep models. In this work, we will study how to *effectively incorporate large-scale dynamic lexicons* into BERT-based sequence labeling models.

Recent works on incorporating lexicon knowledge (Zhang and Yang, 2018; Ding et al., 2019; Mu et al., 2020; Li et al., 2020) can be summarized as follows. First, they match an input sentence with several lexicons to obtain all matched items. Second, leveraging the matched item information through modifying the structure of the transformer layer or the feature representation layer. However,

*Equal contribution.

[1]https://github.com/huawei-noah/noah-research/Dylex

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2679–2693
November 7–11, 2021.

1) current methods normally learn additional embeddings of the words in the lexicons, which bring us a challenge - if the lexicons get updated, the model must be re-trained; 2) they only use the words in the lexicon but ignore the category of words, which is important for many tasks.

In this paper, we propose a general framework DyLex for incorporating frequently updated lexicons into sequence labeling models. The matching results of the input are reconstructed as a word-agnostic tag sequence. Then we design a supervised knowledge denoising module to smooth out noisy matches, and the remaining matches are further used as additional feature input for knowledge fusion. This step is based on a *col-wise attention* to seamlessly fuse word-piece embeddings of input sentence and the lexicon features. Moreover, since we do not explicitly learn embeddings of the words in lexicons, there is no need to retrain the entire model when updating the lexicons.

We conduct extensive experiments with the CWS, NER, and NLU tasks on various datasets. The results show that our model consistently outperforms the strong baselines and achieves new state-of-the-art results.

We summarize the contribution of this work as follows:

1) 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.

2) We devise a novel knowledge denoising module to make full use of large-scale lexicons.

3) Our framework outperforms strong baselines and achieves SOTA results on three different sequence labeling tasks.

2 Approach

In this section, we will present how to incorporate large-scale lexicons into BERT. As illustrated in Figure 2, the proposed DyLex framework contains two parts, namely the *BERT-based sequence tagger* and *Lexicon Knowledge extractor*. The Lexicon Knowledge (LexKg) extractor has three submodules: Matching, Denoising and Fusing.

2.1 BERT as Encoder

Devlin et al. (2019) introduces a new language representation model called BERT, which has become the building blocks of modern NLP systems. BERT is constructed based on transformer (Vaswani et al., 2017) layer, which employs multi-head attention to perform self-attention over a sequence individually and finally applies concatenation and linear transformation to the results from each head. Every single head attention in multi-head attention is calculated in a scaled dot product form:

$$\mathrm{Att}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V, \quad (1)$$

where Q, K, V are input matrices, respectively. Then self-attention can be formalized as:

$$\mathrm{SelfAtt}(X) = \mathrm{Att}(XW_Q, XW_K, XW_V), \quad (2)$$

where W_Q, W_K, W_V are parameter matrices to be learned.

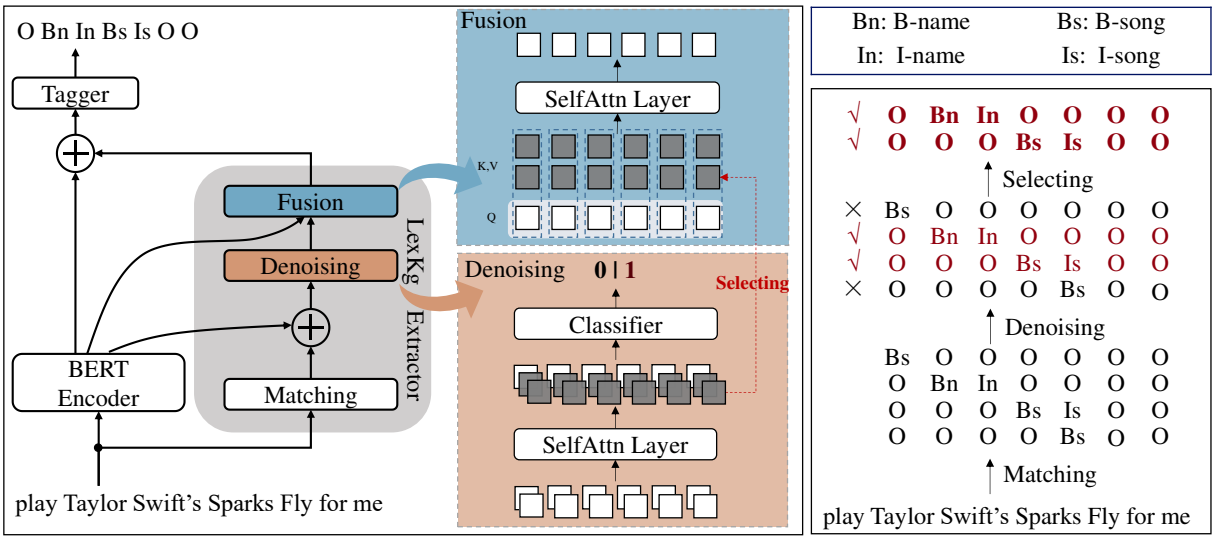
Algorithm 1: Fast Matching

Input: Trie Tree Tr built from Lexicon D, utterance U
Output: Candidate tag sequence T
T = [];
for $i = 0; i \leq length(U)$ **do**
  for $j = i; j \leq length(U)$ **do**
    if $U[i:j]$ *in* T **then**
      `// reconstruct tags`
      $tags \leftarrow$ get_tags(i, j, U, Tr)
      $tags$ append to T;
  end
end
return T;

Function `get_tags` *(i, j, u, Tr)* **:**
  `// get lexical class`
  $class \leftarrow Tr$.match$(u[i:j])$
  $tags \leftarrow$ label $u[i:j]$ with $class$, other position char label O
  return tags

2.2 The LexKg Extractor

Matching Conventional methods normally learn additional word embeddings of lexicons to incorporate lexicon knowledge, thus it is required to retrain the entire model once the lexicons are updated. Our method is independent of the lexicon size and lexicon word content by designing a word-agnostic representation. Specifically, the Matching module takes a word sequence as input, then uses a prefix tree-based fast matching algorithm (see algorithm 1) to quickly retrieve the lexicons, and finally



(a) (b)

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.

produces multiple *word-agnostic tag* sequences. Figure 2 (b) shows a concrete example.

To be detailed, we use the prefix Trie tree (Brass, 2008) to store and retrieve the lexicons. The non-leaf nodes of Trie are made up of word-pieces of lexicon words tokenized by BERT tokenizer, while the leaf nodes are made up by the types of the lexicon words, namely tag name (e.g. 'B-song' and 'I-song' as shown in Figure 2 (b)). For each subsequence of the input, the Trie may match several different candidates. Every single match can be categorized by a tag attached with a leaf node, the rest of the sequence will be filled with 'O' tags.

Formally, we denote the input sequence as U. A tag sequence obtained by fast matching is $T^{(i)}$, and superscript i represents the index of the tag sequence. The *Matching* submodule can be formalized as:

$$E_u = \text{BERT}(U) \tag{3}$$

$$E_t^{(i)} = \text{Embedding}(T^{(i)}) \tag{4}$$

$$E_d^{(i)} = E_t^{(i)} + E_u, \tag{5}$$

where $E_u \in \mathbb{R}^{l \times hz}$ (here l is sequence length and hz is hidden size) is the representation produced by BERT encoder, $E_t^{(i)} \in \mathbb{R}^{l \times hz}$ represents the embedding of i-th tag sequence, and $E_d^{(i)} \in \mathbb{R}^{l \times hz}$ is the corresponding output of this module.

Denoising The proposed fast matching algorithm can quickly obtain all potential matched subsequences with the lexicons. However, due to the large scale size of the lexicon, even for an input sequence with only a few words, there may be dozens of incorrect matches. Using Figure 2 (b) as an example, only Row 2 (i.e. the matching to singer *Taylor Swift's*) and Row 3 (i.e. the matching to song *Sparks Fly*) are expected matchings, whereas all the other tag sequences contain incorrect matchings, namely the ***matching nosie*** mentioned in this work, which will inevitably decrease final performance. Thus we devise a novel supervised knowledge denoising module to smooth them out.

The supervising signal can be automatically derived from the golden sequence labels of the training dataset. In the example of Figure 2 (b), each row corresponds to a single matching tag sequences, and Row 2 and Row 3 are used as positive training samples whereas negative for the other two. Note that, our method can still work even if the category of lexicon (e.g. name or song) is not provided, in that case, a tag sequence degenerates to mark out a lexicon word boundary.

Formally, we first get the representation of i-th tag sequence from its embedding $E_d^{(i)}$ with self-attention

$$R_d^{(i)} = \text{SelfAtt}(E_d^{(i)}). \tag{6}$$

When classifying each tag sequence, we also need to consider relationships among them. For example, Row 3 and Row 4 in Figure 2 (b) can not be True at the same time since they share some contradicting spans. Taking that into consideration, we first

concatenate the `[cls]` in R_d (i.e. first column) of all tag sequences to form a matrix $R_{cls} \in \mathbb{R}^{nd \times hz}$, where nd denotes the number of tag sequences (e.g. its value is 4 in the example of Figure 2 (b)) and hz denotes the hidden representation size. Then we pass the matrix R_{cls} to a self attention layer to model the interrelation among them,

$$Y = \text{SelfAttn}(R_{cls}) \tag{7}$$
$$\text{P}(Z = \text{True} \mid R_d) = \sigma(\text{Linear}(Y)), \tag{8}$$

where σ represents the sigmoid function, and $Z \in \mathbb{R}^{nd}$ is the classification result. The representation of a positively classified tag sequence is denoted as $R_d^{(i)+}$. These selected positive representations will be fused with the original BERT embedding E_u.

Knowledge Fusing In this stage, our framework aims to produce a lexical knowledge enhanced representation E_k by fusing BERT-based encoding E_u with several selected tag sequences R_d^+ via the proposed *col-wise attention*. Use the j-th token of input sequence as an example, we take its BERT-based representation $E_u^{(j)}$ to act as Query, and its corresponding tag representation $R_d^{(i,j)+}$ as Key and Value, then col-wise attention can formulate:

$$K = V = [R_d^{(1,j)+}; \ldots; R_d^{(m,j)+}] \tag{9}$$
$$E_k^{(i)} = \text{Att}(E_u^{(j)}, K, V), \tag{10}$$

where $m = |R_d^+|$. Then concatenate $E_k^{(i)}$ for all l positions to get E_k.

2.3 The Tagger

At last E_O is produced by combining E_u with E_k, and here we use a linear classification layer, as used by BERT tagger.

$$E_O = E_u + E_k \tag{11}$$
$$O = \sigma(\text{Linear}(E_O)) \tag{12}$$

where O is the classification result for each token.

We can see that the proposed framework is not an intrusive method but rather pluggable. As we take the encoder's output as input and return a knowledge enhanced text representation, the original model structure is not modified.

3 Experiments

We conduct experiments on several NLP tasks, including CWS (Chinese word segmentation), NER (named entity recognition), and NLU (natural language understanding). The experimental hyperparameter settings are listed in appendix F.

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
NLU	words	Location name	B-loc: Begining of a location name I-loc: Continuation of a location name

Table 1: Examples of lexicon's content in different tasks.

3.1 Primary Baselines

BERT-based Sequence Tagger The framework uses BERT as an encoder to represent the input sequence. As can be seen in Figure 2, we can get this baseline by removing the LexKg extractor part of DyLex.

Glyce (Meng et al., 2019) Glyce is the glyph-vectors for Chinese character representations. With the lexicon, it has achieved the best performance on Chinese word segmentation so far.

FLAT and HSCRF+Softdict (Li et al., 2020; Liu et al., 2019a) Named entity recognition can benefit greatly from lexicons. FLAT utilizes lexicons with the Lattice structure for Chinese entity recognition, and HSCRF with softdict is used for English named entity recognition, both of them have achieved strong results.

3.2 Lexicon Construction

The lexicon mentioned in this article refers to a collection, the entry of which contains item and Category. The item corresponds to the words, and the category corresponds to the type of the words. The category of words is customized according to the task. For example, the category in the NER task can be the song name. Tag is a BIO format that marks the type of a word. Table 1 shows notation and appendix E is a detailed lexicon fragment.

The lexicon tag mentioned above is used to mark word categories, namely the value in the lexicon, which is strongly related to the task. Figure 2(b) and the 'Tag' column in Table 1 display some examples.

The lexicons used in our experiments are consistent with the ones used in baseline methods. In the NLU task, since there has not been any related work with using lexicons, we extract labeled spans from the training corpus and merge them with the lexicon used in NER task. The lexicon sizes used in our experiments are listed in appendix B.

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.

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**

Table 4: F1 Score on PKU and CITYU datasets.

3.3 Task1: Chinese Word Segmentation

CWS aims to divide a sentence into meaningful chunks. It is a primary task for Chinese text processing. Using lexicons in CWS tasks is a commonly used operation. Brand new words and internet buzzwords emerge every day, and it is essential to add these words into lexicons for better performance.

In this work, we experiment on two popular CWS datasets, i.e., PKU and CITYU(Emerson, 2005). The lexicon used in this experiment is consistent with jieba word segmentation lexicon[2], which consists of a simplified Chinese lexicon from jieba and an extra traditional Chinese lexicon from Taiwan version of jieba. We converted all traditional Chinese into simplified Chinese for all lexicons and datasets.

To fairly compare our model with the SOTA models, we use the same settings on dataset split with Meng et al. (2019).

As shown in Table 4, our method outperforms all the other compared baselines. Compared with Glyce, which is a strong baseline, our method obtains improvement of 0.44% and 0.7% on PKU and CITYU respectively.

3.4 Task2: Named Entity Recognition

Named entity recognition is a typical sequence labeling task, and it heavily relies on external knowledge. Incorporating lexicon as external knowledge can help determine the span and type of entities. To fully verify the capability of the proposed framework in NER, we evaluate our framework on Ontonotes (Weischedel and Consortium, 2013), MSRA (Levow, 2006), Resume (Zhang and Yang, 2018), and Weibo (Peng and Dredze, 2015; He and Sun, 2017) for Chinese, and Conll2003 (Sang

[2] https://github.com/fxsjy/jieba

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.

and Meulder, 2003) and Ontonotes (Pradhan and Ramshaw, 2017) for English. The statistics of these datasets are detailed in Table C1. The lexicon used in Chinese NER tasks is the same as Li et al. (2020), and the one in English is the same as Liu et al. (2019a).

We first evaluate our framework on the Chinese datasets, and the results are shown in Table 2. Except for the Ontonotes, our approach achieves the best results over all methods with lexicons, averagely 0.69% higher than FLAT. Compared with BERT, which is the best method without using lexicon, our approach improves even more dramatically, with 1.57% higher.

We evaluate our framework on two English datasets (i.e., Conll2003, OnotNotes5.0). The conclusion is similar to Chinese Datasets, as shown in Table 3. Comparing with the HSCRF and CSE, our method is 0.91% and 0.85% higher on average, with and without lexicon respectively. LUKE (Yamada et al., 2020) scores the same as our method on the conll 2003 data set, and also uses information related to entities. they achieved it through pre-training, which is orthogonal to our method.

3.5 Task3: Natural Language Understanding

NLU is a more challenging sequence labeling task, which aims to recognize the intent of spoken language and extract slots. As shown in Figure 1, in many practical application scenarios, one cannot tell the real intent unless the entity is provided as prior knowledge.

We evaluate the framework on an industrial data set and two public data sets. The chinese industrial data set is a commercial dataset for mobile phone assistant. The public datasets are Snips[3] and ATIS (Tür et al., 2010). The details of the three datasets are shown in appendix D.

The overall performance of our framework on the industrial dataset is listed in Table 5. For the test set, there are 0.76% and 1.53% improvements in intent detection and slot filling, respectively. Specifically, the gain is more obvious in the SINGLE and MULTI set. The BERT can not distinguish intent between "play music" and "play video" since the model lacks the prior knowledge of whether "Love Story" is a song or a movie. In the MEDIA set, all sentences contain demonstrative words, such as "play music [xxx]" and "play video [xxx]". This type of sentence does not depend on the type of xxx. It is easy to make judgments through the demonstrative words (i.e., music and video), but there is still a 0.5% increase in intent detection, and the increment in the slot filling is even more obvious, reaching 2.21%.

The experimental results on Snips and ATIS are shown in Table 6, the setting follows previous works (E et al., 2019; Goo et al., 2018). It can be seen that our framework outperforms the other methods in all three metrics (except slot of

[3]https://github.com/snipsco/nlubenchmark/

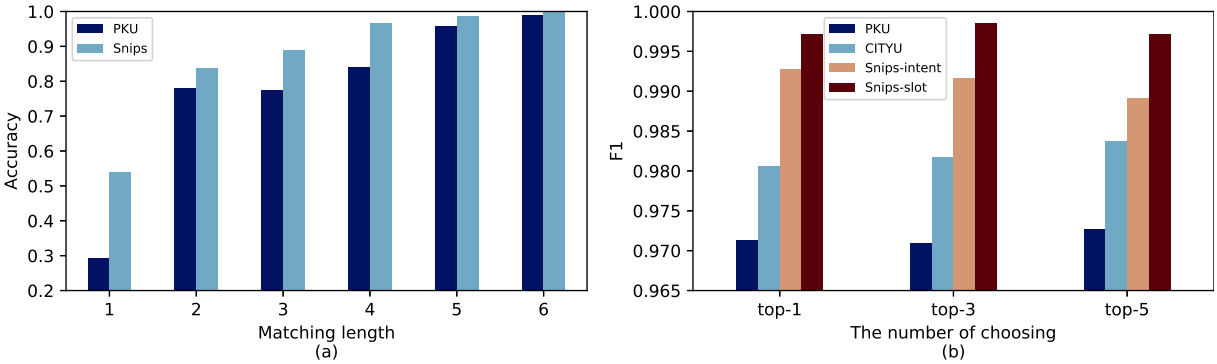


Figure 3: (a) The Influence of matching length (the x-axis represents the matched word's length, and the y-axis represents the proportion of correct results in all matching results). (b) F1 score on top-n candidates by reverse order of match length. (top-n means fetching n the longest matching results).

ATIS): slot filling (F1), intent detection (Acc), and sentence accuracy (Acc), with 1.27% higher on average than the previous best method. For ATIS, the improvement is not as much as other methods. This is mainly because the dataset is relatively small and the slot is sparse, lexicons are underutilized.

4 Discussion

4.1 The Study of Match Length

Given an utterance, the FM(algorithm 1) often produces numerous matching results for each position. On the one hand, we are not sure which result is correct. To retain the correct result, we should keep as many results as possible. On the other hand, most matching results are invalid, bringing a lot of matching noise and increasing computation cost. We have to make a balance between them. As shown in the Figure 3(a), the longer the length is, the higher the accuracy is. Based on this observation, we should select matching results by reverse order of match length.

We also studied the number of selected results for each position in the sentence. It is more likely to keep the right matches with a larger number, but it brings more noise. From Figure 3(b), F1 on the three data sets do not increase as the number grows. Taking efficiency into account, we generally select $n = 1$ or $n = 2$.

4.2 Effect of Dynamic Lexicon

One advantage of our proposed method is the ability to load lexicons dynamically. Instead of using the embedding of updated lexicon entries, we only use the lexicon words' category tags. Thus we can expand the scale of lexicons arbitrarily without retraining. We studied the effect of lexicon size on

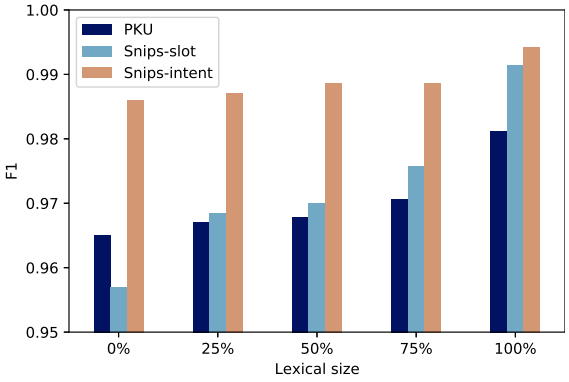


Figure 4: The F1 of different task with different lexical size. When the size is 100%, it means using the entire lexicon in the corresponding experiment above.

performance. From Figure 4, we can see that without using a lexicon, the performance are close to the results of BERT base. With the increasing size of lexicon, the performance will also be improved.

4.3 Look Back on Denoising

Indistinguishable lexicon matching will bring huge noise. The quality of denoising will affect the performance of the model. From Table 7, we can see that whether it is Exp-Dict or Sp-Dict, the more precise the denoising, the more improvement will be achieved compared to BERT without using a lexicon. The Sp-Dict here is a specialized collection of domain lexicons. For example, the lexicon only contains entities of the relevant category in the NER task, and the scale is relatively small. In this case, the matching noise brought by Sp-Dict is much smaller. From the Table 7, we can observe that the accuracy of denoising in Sp-dict is better than that in Exp-dict, which directly leads to impressive improvement in the experiment. This also confirms the importance of denoising.

Task	Datasets	BERT	Exp-Dict		Sp-Dict	
			Denoising	Dylex	Denoising	Dylex
CWS	PKU	96.50	97.90	97.14(+0.64)	99.26	98.11(+1.61)
	CITYU	97.60	97.91	98.06(+0.46)	99.14	98.72(+1.12)
NER-Chinese	Ontonotes	80.14	97.83	81.48(+1.34)	98.37	82.31(+2.17)
	MSRA	94.65	98.10	96.40(+1.75)	98.74	96.85(+2.20)
	Resume	95.53	97.92	95.99(+0.46)	98.82	96.40(+0.87)
	Weibo	68.20	96.93	71.12(+2.92)	97.83	71.53(+3.33)
NER-English	Conll2003	92.40	98.66	94.30(+1.90)	98.81	94.44(+2.04)
	Ontonotes5.0	89.13	97.24	90.19(+1.06)	98.19	91.40(+2.27)

Table 7: Column BERT represents the F1 on each task, the Denoising column represents the accuracy of the denoising module, and the Dylex column is the F1 of our method and its increment versus BERT. Exp-dict is the lexicon corresponding to each experiment above, and Sp-Dict indicates specialized domain-related lexicons.

4.4 Fusion in Hard or Soft Way

After Denoising, the results R_d should be fused with E_u for downstream tasks. The fused methods can be soft or hard. In the soft setting, all of the R_d are weighted summed before fusing with E_u. The advantage of this is we can use gradient back-propagation to train the model. Different from the soft method, E_u in the hard method is selected according to the threshold. As shown in Table 8, the overall performance of hard fusion is better since it mainly fuses more accurate results. Besides, we also adopt Teacher Forcing (Williams and Zipser, 1989) in soft/hard methods, but it does not yield promising accuracies.

Methods	MSRA		Resume	
	Exp-Dict	Sp-Dict	Exp-Dict	Exp-Dict
Soft	95.26	95.51	95.13	94.91
Hard	**96.40**	**96.85**	**95.99**	**96.40**

Table 8: The F1 of two selecting strategies.

5 Related Work

With the advance of deep learning, sequence labelling tasks, such as segmentation and NER, have achieved excellent performance. More and more methods tend to be character-based(Chen et al., 2006; Lu et al., 2016; Dong et al., 2016), especially in languages, such as Chinese, Japanese, Korean, etc., that require word segmentation. These languages do not have a natural segmentation delimiter as white space in Latin languages. Character-based input in these languages can avoid accumulation of word segmentation errors, then get better performances(He and Wang, 2008; Liu et al., 2010; Li et al., 2014). However, the downside of the purely character-based method is that the word information is not fully exploited.

To make full use of word information, incorporating a lexicon is an effective method. Existing works on incorporating lexicon can be categorized as feature based, lattice based and graph based methods according to implementation complexity.

Feature based Feature based method is a simpler way. Some works directly use lexical information with simple matching features and the others use auxiliary tasks to leverage the lexical information. Zhang et al. (2018) builds the template first and uses the template matching lexicon to build features, which help word segmentation tasks. Mu et al. (2020) uses a simple lexicon matching location information as features. Li et al. (2014) and Peters et al. (2017) adopt word-level language modeling objective and multi-task to use word information implicitly. Yang et al. (2017b) transfer cross-domain and cross-lingual knowledge via multi-task learning.

Lattice based Lattice based method is to use lattice structure. Zhang and Yang (2018) proposes Lattice-LSTM for incorporating word lexicons into the character-based NER model. Rather than heuristically choosing a word for the character when matching multiple words in the lexicon, they also introduce an elaborate modification to the sequence modeling layer of the LSTM-CRF model(Huang et al., 2015). Considering that the short path in the lattice structure will cause the word-based structure to degenerate into a character-based structure, Liu et al. (2019b) propose a novel word character LSTM(WC-LSTM) model to add

word information via four strategies. Since the lattice structure is complex and dynamic, most existing lattice-based models cannot fully utilize GPUs' parallel computation and usually have a low inference-speed. Li et al. (2020) propose a Transformer-based model for Chinese NER, which converts the lattice structure into a flat structure.

Graph based Graph based method uses a directed graph structure to fuse lexiconal information. Gui et al. (2019b) uses a GNN-based method to explore multiple graph-based interactions among characters, potential words, and the whole-sentence semantics to effectively alleviate the word ambiguity. Sui et al. (2019) employ a collaborative graph network to assign both self-matched and the nearest contextual lexical terms. To automatically learn how to incorporate multiple gazetteers into a NER system, Ding et al. (2019) propose a novel approach based on graph neural networks with a multidigraph structure. The structure captures the information the gazetteers offer.

6 Conclusion and Future Work

In this paper, we propose DyLex, a framework incorporating dynamic lexicon to improve BERT-like models' performance in sequence labeling tasks. To alleviate the problems caused by large-scale dynamic lexicons, we introduce word-agnostic tag embeddings and a knowledge denoising module. As a result, our framework outperforms the state-of-the-art works on many sequence labeling tasks. In future, how to extend it to text classification is a challenge, since denoising corpus cannot be automatically constructed at this time.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. We also appreciate the helpful discussion with the colleagues in our team.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics.

Peter Brass. 2008. *Advanced data structures*, volume 193. Cambridge University Press Cambridge.

Hui Chen, Zijia Lin, Guiguang Ding, Jianguang Lou, Yusen Zhang, and Börje Karlsson. 2019a. GRN: gated relation network to enhance convolutional neural network for named entity recognition. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6236–6243. AAAI Press.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019b. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. 2006. Chinese named entity recognition with conditional random fields. In *Proceedings of the Fifth Workshop on Chinese Language Processing, SIGHAN@COLING/ACL 2006, Sydney, Australia, July 22-23, 2006*, pages 118–121. Association for Computational Linguistics.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Trans. Assoc. Comput. Linguistics*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for chinese NER with gazetteers. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1462–1467. Association for Computational Linguistics.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based LSTM-CRF with radical-level features for chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications - 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016, Kunming, China, December 2-6, 2016, Proceedings*, volume 10102 of *Lecture Notes in Computer Science*, pages 239–250. Springer.

Haihong E, 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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5467–5471. Association for Computational Linguistics.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2005, Jeju Island, Korea, 14-15, 2005*. ACL.

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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 753–757. Association for Computational Linguistics.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019a. Cnn-based chinese NER with lexicon rethinking. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4982–4988. ijcai.org.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019b. A lexicon-based graph neural network for chinese NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1040–1050. Association for Computational Linguistics.

Hangfeng He and Xu Sun. 2017. F-score driven max margin neural network for named entity recognition in chinese social media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 713–718. Association for Computational Linguistics.

Jingzhou He and Houfeng Wang. 2008. Chinese named entity recognition and word segmentation based on character. In *Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008*, pages 128–132. The Association for Computer Linguistics.

Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu. 2020a. Towards fast and accurate neural chinese word segmentation with multi-criteria learning. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2062–2072. International Committee on Computational Linguistics.

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

Zhiqi Huang, Fenglin Liu, and Yuexian Zou. 2020b. Federated learning for spoken language understanding. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3467–3478. International Committee on Computational Linguistics.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth Workshop on Chinese Language Processing, SIGHAN@COLING/ACL 2006, Sydney, Australia, July 22-23, 2006*, pages 108–117. Association for Computational Linguistics.

Haibo Li, Masato Hagiwara, Qi Li, and Heng Ji. 2014. Comparison of the impact of word segmentation on name tagging for chinese and japanese. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 2532–2536. European Language Resources Association (ELRA).

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6836–6842. Association for Computational Linguistics.

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 685–689. ISCA.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019a. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5301–5307. Association for Computational Linguistics.

Wei Liu, Tongge Xu, QingHua Xu, Jiayu Song, and Yueran Zu. 2019b. An encoding strategy based word-character LSTM for chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2379–2389. Association for Computational Linguistics.

Zhangxun Liu, Conghui Zhu, and Tiejun Zhao. 2010. Chinese named entity recognition with a sequence labeling approach: Based on characters, or based on words? In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, 6th International Conference on Intelligent Computing, ICIC 2010, Changsha, China, August 18-21, 2010. Proceedings*, volume 6216 of *Lecture Notes in Computer Science*, pages 634–640. Springer.

Yanan Lu, Yue Zhang, and Dong-Hong Ji. 2016. Multi-prototype chinese character embedding. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 879–888. The Association for Computational Linguistics.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art chinese word segmentation with bi-lstms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4902–4908. Association for Computational Linguistics.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the usage of lexicon in chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5951–5960. Association for Computational Linguistics.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *Advances in Neural Information Processing Systems*, pages 2746–2757.

Xiaofeng Mu, Wang Wei, and Xu Aiping. 2020. Incorporating token-level dictionary feature into neural model for named entity recognition. *Neurocomputing*.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 548–554. The Association for Computational Linguistics.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1756–1765. Association for Computational Linguistics.

Sameer Pradhan and Lance Ramshaw. 2017. Ontonotes: Large scale multi-layer, multi-lingual, distributed annotation. In *Handbook of Linguistic Annotation*, pages 521–554. Springer.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2670–2680. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for chinese named entity recognition via collaborative graph network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3828–3838. Association for Computational Linguistics.

Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop, SLT 2010, Berkeley, California, USA, December 12-15, 2010*, pages 19–24. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Ralph M Weischedel and Linguistic Data Consortium. 2013. Ontonotes release 5.0. Title from disc label.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: adapting transformer encoder for named entity recognition. *CoRR*, abs/1911.04474.

Jie Yang, Yue Zhang, and Fei Dong. 2017a. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 839–849. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017b. Transfer learning for sequence tagging with hierarchical recurrent networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for chinese word segmentation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5682–5689. AAAI Press.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1554–1564. Association for Computational Linguistics.

A Case Study

Input(1)	Play	this	is	colour	by	panda	Bear.											
In dict	o	track	track	track	o	artist	artist											
Baseline	o	album	album	album	o	artist	artist											
DyLex	o	track	track	track	o	artist	artist											
Input(2)	Use	netflix	to	play	bizzy	bone	kiss	me	good-night	Serge-ant	major							
In dict	o	service	o	o	artist	artist	track	track	track	track	track							
Baseline	o	service	o	o	track	track	track	track	track	track	track							
DyLex	o	service	o	o	artist	artist	track	track	track	track	track							
Input(3)	I	want	to	add	hind	etin	to	my	la	mejor	musica	dance	2017	playlist				
In dict	o	o	o	o	entity	entity	o	owner	plst	plst	plst	plst	plst	plst				
Baseline	o	o	o	o	artist	artist	o	owner	plst	plst	plst	plst	plst	plst				
DyLex	o	o	o	o	artist	artist	o	owner	plst	plst	plst	plst	plst	plst				
Input(4)	what	is	the	weather	like	in	north	salt	lake	and	afghanistan							
In dict	o	o	o	o	o	o	city	city	city	o	country							
Baseline	o	o	o	o	o	o	country	country	country	o	country							
DyLex	o	o	o	o	o	o	city	city	city	o	country							
Input(5)	I	want	to	book	a	cafe	for	3	in	fargo								
In dict	o	o	o	o	o	res_type	o	o	o	city								
Baseline	o	o	o	o	o	res_type	o	o	o	country								
DyLex	o	o	o	o	o	res_type	o	o	o	city								
Input(6)	play	the	new	noise	theology	ep	intent											
In dict	o	object	object	object	object	object												
Baseline	o	plst	plst	plst	plst	plst	PlayMusic											
DyLex	o	object	object	object	object	object	SearchCreativeWork											
Input(7)	Find	a	man	needs	a	maid	Bear.	Intent										
In dict	o	object	object	object	object	object	object											
Baseline	o	movie	movie	o	movie	movie	movie	SearchScreeningEvent										
DyLex	o	object	object	object	object	object	object	SearchCreativeWork										
Input(8)	播	放	林	星	辰	的	音	乐	盒	Intent								
In dict	o	o	artist	artist	artist	o	track	track	track									
Baseline	o	o	artist	artist	track	track	track	track	track	PlayMusic								
DyLex	o	o	artist	artist	artist	o	track	track	track	PlayMusic								
Input(9)	播	放	林	星	辰	的	音	乐	盒	Intent								
In dict	o	o	track	track	track	track	track	track	track									
Baseline	o	o	artist	artist	track	track	track	track	track	PlayMusic								
DyLex	o	o	track	track	track	track	track	track	track	PlayMusic								
Input(10)	外	国	政	要	发	表	新	年	贺	词	满	怀	信	心	应	对	挑	战
In dict	B	I	B	I	B	I	B	I	B	I	B	I	I	I	B	I	B	I
Baseline	B	I	B	I	B	I	B	I	B	I	B	I	I	I	B	B	B	I
DyLex	B	I	B	I	B	I	B	I	B	I	B	I	I	I	B	I	B	I
Input(11)	环	南	中	国	海	自	行	车	赛	落	幕	澳	门					
In dict	B	B	I	I	I	B	I	I	I	B	I	B	I					
Baseline	B	B	B	I	I	B	I	I	I	B	I	B	I					
DyLex	B	B	I	I	I	B	I	I	I	B	I	B	I					
Input(12)	这	起	发	生	在	校	园	内	的	重	大	安	全	责	任	事	故	
In dict	B	B	B	I	B	B	I	B	B	B	I	B	I	B	I	I	I	
Baseline	B	B	B	I	B	B	I	B	B	B	I	B	I	B	I	B	I	
DyLex	B	B	B	I	B	B	I	B	B	B	I	B	I	B	I	I	I	

As showed in above, we randomly select some examples of inconsistent predictions before and after adding the lexicons, example [1-5] is from NER, example [6-9] is from NLU, and example [10-12] is from CWS . Each example contains the input sentence, the related matching result, the baseline prediction, and **DyLex** prediction. Highlighted parts indicate inconsistent results. We make some interesting observations.

CASE I Different type of entities can be placed under a same context. For example [1], "play" can be followed by TRACK or ALBUM (play [XX]). Model would be confused of whether XX is a TRACK or a ALBUM. In this case lexicons can provide enough type information to acquire a correct result.

CASE II Chinese word segmentation granularity is flexible according to the context. “南中国海(South China Sea)” can be segmented into “南(South)” and “中国海(China Sea)”, or it can be regarded as a single word [11]. At this point, an external lexicon will be benefit for controlling the granularity.

CASE III It happens that the word combination in slot have different interpretations, usually when the length of a slot is too long. That may cause the discontinuity of slot extraction. For example, we can see an improper O is inserted in the baseline prediction [7]. By incorporating lexicons, the boundary information can enhance the integrity of slot extraction.

CASE IV Dylex can adapt its prediction to updating lexicons. As example[8-9] illustrated, given different lexicon entries, our framework can understand what “林星辰的音乐盒” is, then dynamicly provide correct slot.

B Lexicon size uesd in different experiment

Task	Datasets	Exp-Dict	Sp-Dict
CWS	PKU	570K	57.7K
	CITYU	579K	70.5K
NER-CN	Ontonotes	97.2K	68.6K
	MSRA	98.1K	80.5K
	Resume	97.9K	68.9K
	Weibo	96.9K	62.9K
NER-EN	Conll2003	1.3M	33K
	Ontonotes5.0	1.3M	47K
NLU	ATIS	1.3K	1.3K
	Snips	12K	12K
	Industrial NLU	16M	16M

Table B1: Lexicon size(number of term) uesd in different experiment

C Overiew of NER dataset

	Ontones	MSRA	Resume	Weibo	Conll2003	OntoNotes5.0
train	15,470	46,675	3,821	1,350	14,987	115,812
char_{avg}	36.92	45.87	32.15	54.37	-	-
word_{avg}	17.59	22.38	24.99	21.49	13.5	9.40
entity_{avg}	1.15	1.58	3.48	1.42	1.56	0.71

Table C1: Overiew of NER dataset

D Overiew of NLU dataset

Type	Dataset	Train	Dev	Test	Intents	Slots
Industrial	-	80,000	30,000	30,000	500	400
Public	Snips	13,084	700	700	7	72
	ATIS	4,478	500	893	21	120

Table D1: The stastics of NLU datasets.

The Chinese industrial NLU dataset is a corpus specially used to train mobile phone assistants. The data set includes 80k Training set, 30k Dev set and 30k Test set. The annotation contains 500 types of intentions commonly used by mobile assistants , which are divided into 8 categories such as setting and control. There are 400 slots categories in total. The data is labeled using crowdsourcing. The cost is about 1 dollar per sentence. Each sentence was marked by 3 people, and finally the result was determined by voting. At last, there is a acceptance sampling, and professionals will spot check the quality of each batch, and the error is controlled within 1%.

E A concrete example of a lexicon

Item	Category
cathy mu ∼no ##z	PER
pieter pieter ##sz barbie ##rs	PER
bell high school	ORG
fredrik ri ##sp	PER
liverpool	ORG
venice gardens	LOC
brant ##ford golden eagles	ORG
jerry and ##rus	PER
taylor leon	PER
kata ##rina e ##wer ##lo ##f	PER
anne finch	PER
hanna st ##yre ##ll	PER
the big blue	MISC
math ##are united	ORG
var ##ana ##si college of pharmacy	ORG
gilbert ##sville	LOC

Table E1: A fragment of the lexicon used in this article. The Item on the left is the wordpiece of the words, and the corresponding category on the right.

F Hyperparameters

batch_size	[32, 64]	
learning_rate	2e-5	
optimizer	Adam	
weight_decay	0.01	
dropout	0.1	
max_seq_length	128	
dict_candidate	16	#the maximum number of matches per sentence
top_n	1	#number of matches reserved for each position
warmup_proportion	0.1	
epochs	20	
use_first	True	#only the first character category is used to predict the entity type

Table F1: The hyperparameters used in the experiment. Other hyperparameters default are consistent with BERT.