Reliability-aware Dynamic Feature Composition for Name Tagging

Ying Lin[1], **Liyuan Liu**[2], **Heng Ji**[1,2], **Dong Yu**[3] **and Jiawei Han**[2]
[1] Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA
[2] Dept. of Computer Science, University of Illinois at Urbana-Champaign
Urbana, IL, USA
{yinglin8,ll2,hengji,hanj}@illinois.edu
[3] Tencent AI Lab, Bellevue, WA, USA
dyu@tencent.com

Abstract

While word embeddings are widely used for a variety of tasks and substantially improve the performance, their quality is not consistent throughout the vocabulary due to the long-tail distribution of word frequency. Without sufficient contexts, embeddings of rare words are usually less reliable than those of common words. However, current models typically trust all word embeddings equally regardless of their reliability and thus may introduce noise and hurt the performance. Since names often contain rare and unknown words, this problem is particularly critical for name tagging. In this paper, we propose a novel reliability-aware name tagging model to tackle this issue. We design a set of word frequency-based reliability signals to indicate the quality of each word embedding. Guided by the reliability signals, the model is able to dynamically select and compose features such as word embedding and character-level representation using gating mechanisms. For example, if an input word is rare, the model relies less on its word embedding and assigns higher weights to its character and contextual features. Experiments on OntoNotes 5.0 show that our model outperforms the baseline model, obtaining up to 6.2% absolute gain in F-score. In cross-genre experiments on six genres in OntoNotes, our model improves the performance for most genre pairs and achieves 2.3% absolute F-score gain on average. [1]

1 Introduction

Serving as the basic unit of the model input, word embeddings form the foundation of various natural language processing techniques using deep neural networks. Embeddings can effectively encode semantic information and have proven successful in a wide range of tasks, such as sequence labeling (Collobert et al., 2011; Chiu and Nichols, 2016; Ma and Hovy, 2016; Lample et al., 2016), text classification (Tang et al., 2014; Lai et al., 2015; Yang et al., 2016), and parsing (Chen and Manning, 2014; Dyer et al., 2015). Still, due to the long tail distribution, the quality of pre-trained word embeddings is usually inconsistent. Without sufficient contexts, the embeddings of rare words are less reliable and may introduce noise, as current models disregard their quality and consume them in the same way as well-trained embeddings for common words. This issue is particularly important for *name tagging*, the task of identifying and classifying names from unstructured texts, because names usually contain rare and unknown words, especially when we move to new domains, topics, and genres.

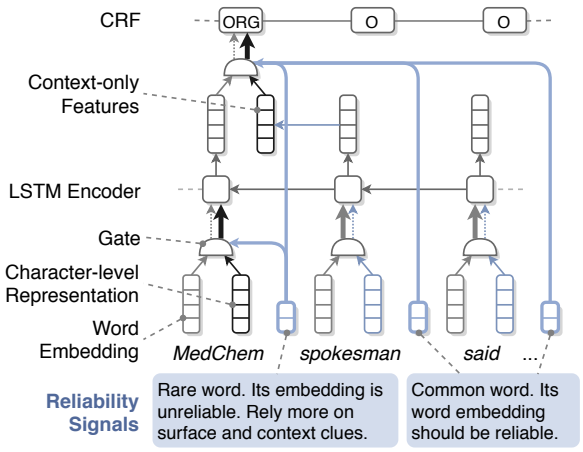

Figure 1: A simplified illustration of the proposed model. We only show the backward part in the figure.

By contrast, when encountering an unknown word, human readers usually seek other clues in the text. Similarly, when informed that an embed-

[1]Code and resources for this paper: https://github.com/limteng-rpi/neural_name_tagging

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 165–174
Florence, Italy, July 28 - August 2, 2019.

ding is noisy or uninformative, the model should rely more on other features. Therefore, we aim to make the model aware of the quality of input embeddings and guide the model to dynamically select and compose features using explicit *reliability signals*. For example, in Figure 1, since the model is informed of the relatively low quality of the word embedding of "MedChem", which only occurs 8 times in the embedding training corpus, it assigns higher weights to other features such as its character-level representation and contextual features derived from its context words (e.g., "spokesman").

The basis of this dynamic composition mechanism is the reliability signals that inform the model of the quality of each word embedding. Specifically, we assume that if a word occurs more frequently, its word embedding will be more fully trained as it has richer contexts and its embedding is updated more often during training. Thus, we design a set of reliability signals based on word frequency in the embedding training corpus and name tagging training corpus.

As Figure 1 shows, we use reliability signals to control feature composition at two levels in our model. At the word representation level, in addition to word embedding, we generate a character-level representation for each word from its compositional characters using convolutional neural networks (see Section 2.1). Such character-level representation is able to capture semantic and morphological information. For example, the character features extracted from "Med" and "Chem" may encode semantic properties related to medical and chemical industries. At the feature extraction level, we introduce *context-only* features that are derived only from the context and thus not subject to the quality of the current word representation. For rare words without reliable representations, the contexts may provide crucial information to determine whether they are part of names or not. For example, "spokesman", "products", and "sales" in the context can help the model identify "MedChem" as an organization name. Additionally, context-only features are generally more robust because most non-name tokens in the context are common words and unlikely to vary widely across topics and scenarios. To incorporate the character-level representation and context-only features, we design new gating mechanisms to mix them with the word embedding and encoder output respectively. These reliability-aware gates learn to dynamically assign weights to various types of features to obtain an optimal mixture.

Experiments on six genres in OntoNotes (see Section 3.1) show that our model outperforms the baseline model without the proposed dynamic feature composition mechanism. In the cross-genre experiments, our model improves the performance for most pairs and obtains 2.3% absolute gain in F-score on average.

2 Model

In this section, we will elaborate each component of our model. In Section 2.1, we will describe the baseline model for name tagging. After that, we will introduce the frequency-based reliability signals in Section 2.2. In Section 2.3, We will elaborate how we guide gates to dynamically compose features at the word representation level and feature extraction level.

2.1 Baseline Model

We adopt a state-of-the-art name tagging model LSTM-CNN (Long-short Term Memory - Convolutional Neural Network) (Chiu and Nichols, 2016) as our base model.

In this architecture, the input sentence is represented as a sequence of vectors $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_L\}$, where $\boldsymbol{x}_i$ is the vector representation of the i-th word, and L is the length of the sequence. Generally, $\boldsymbol{x}_i$ is a concatenation of word embedding and character-level representation generated with a group of convolutional neural networks (CNNs) with various filter sizes from compositional character embeddings of the word.

Next, the sequence $\boldsymbol{X}$ is fed into a bi-directional Recurrent Neural Network (RNN) with Long-short Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997). The bi-directional LSTM network processes the sentence in a sequential manner and encodes both contextual and non-contextual features of each word $\boldsymbol{x}_i$ into a hidden state $\boldsymbol{h}_i$, which is afterwards decoded by a linear layer into $\boldsymbol{y}_i$. Each component of $\boldsymbol{y}_i$ represents the score for the corresponding name tag category.

On top of the model, a CRF (Lafferty et al., 2001) layer is employed to capture the dependencies among predicted tags. Therefore, given an input sequence $\boldsymbol{X}$ and the output of the linear layer $\boldsymbol{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_L\}$, we define the score of a se-

quence of predictions $\hat{\boldsymbol{z}} = \{\hat{z}_1, ..., \hat{z}_L\}$ to be

$$s(\boldsymbol{X}, \hat{\boldsymbol{z}}) = \sum_{i=1}^{L+1} \boldsymbol{A}_{\hat{z}_{i-1}, \hat{z}_i} + \sum_{i=1}^{L} y_{i, \hat{z}_i},$$

where $\boldsymbol{A}_{\hat{z}_{i-1}, \hat{z}_i}$ is the score of transitioning from tag $\hat{z}_{i-1}$ to tag $\hat{z}_i$, and $y_{i, \hat{z}_i}$ is the component of $\boldsymbol{y}_i$ that corresponds to tag $\hat{z}_i$. Additionally, $\hat{z}_0$ and $\hat{z}_{L+1}$ are the `<start>` and `<end>` tags padded to the predictions.

During training, we maximize the sentence-level log-likelihood of the true tag path $\boldsymbol{z}$ given the input sequence as

$$\begin{aligned} \log p(\boldsymbol{z}|\boldsymbol{X}) &= \log \left(\frac{e^{s(\boldsymbol{X}, \boldsymbol{z})}}{\sum_{\hat{\boldsymbol{z}} \in Z} e^{s(\boldsymbol{X}, \hat{\boldsymbol{z}})}} \right) \\ &= s(\boldsymbol{X}, \boldsymbol{z}) - \log \sum_{\hat{\boldsymbol{z}} \in Z} e^{s(\boldsymbol{X}, \hat{\boldsymbol{z}})}, \end{aligned}$$

where Z is the set of all possible tag paths.

Note that in addition to word embeddings and character-level representations, (Chiu and Nichols, 2016) uses additional features such as capitalization and lexicons, which are not included in our implementation. Other similar name tagging model will be discussed in Section 4.

2.2 Reliability Signals

As the basis of the proposed dynamic feature composition mechanism, reliability signals aim to inform the model of the quality of input word embeddings. Due to the lack of evaluation methods that directly measure the reliability of a single word embedding (Bakarov, 2018), we design a set of reliability signals based on word frequency as follows:

1. Word frequency in the word embedding training corpus f^{e}. Generally, if a word has more occurrences in the corpus, it will appear in more diverse contexts, and its word embedding will be updated more times.

2. Word frequency in the name tagging training set f^{n}. By fine-tuning pre-trained word embeddings, the name tagging model can encode task-specific information (e.g., "department" is often part of an organization name) into embeddings of words in the name tagging training set and improve their quality.

Because word frequency has a broad range of values, we normalize it with $\tanh(\lambda f)$, where λ is set to 0.001 for f^{e} and 0.01 for f^{n} as the average word frequency is higher in the embedding training corpus. We do not use relative frequency because it turns low frequencies into very small numbers close to zero. Using $\tanh$ as the normalization function, the model can react more sensitively towards lower frequency values.

In addition to the above numeric signals, we introduce binary signals to give the model more explicit clues of the rarity of each word. For example, because we filter out words occurring less than 5 times during word embedding training, the following binary signal can explicitly inform the model whether a word is out-of-vocabulary or not:

$$b(f^{\mathrm{e}}, 5) = \begin{cases} 1, \text{ if } f^{\mathrm{e}} < 5 \\ 0, \text{ if } f^{\mathrm{e}} \geq 5 \end{cases}$$

We heuristically set the thresholds to 5, 10, 100, 1000, and 10000 for f^{e} and 5, 10, 50 for f^{n} based on the average word frequency in both corpora.

The reliability signals of each word are represented as a vector, of which each component is a certain numeric or binary signal. We apply a dropout layer (Srivastava et al., 2014) with probability 0.2 to the reliability signals.

2.3 Dynamic Feature Composition

Word Representation Level

It is a common practice in current name tagging models to utilize character-level representations to address the following limitations of word embeddings: 1. Word embeddings take words as atomic units and thus ignore useful subword information such as affixes; 2. Pre-trained word embddings are not available for unknown words, which are typically represented using a randomly initialized vector in current models.

Unlike previous methods that generally use the character-level representation as an additional feature under the assumption that word- and character-level representations learn disjoint features, we split the character-level representation into two segments: the first segment serves as an alternative representation to encode the same semantic information as word embedding and is mixed with word embedding using gating mechanisms; the second segment is used as an additional feature to encode morphological information that cannot be captured by word embedding.

As Figure 2 illustrates, given the i-th word in a sentence, $\boldsymbol{x}_i^{\mathrm{w}} \in \mathbb{R}^{d_{\mathrm{w}}}$ denotes its word embedding,

$\boldsymbol{x}_i^{\mathrm{c}} \in \mathbb{R}^{d_{\mathrm{c}}}$ denotes its character-level representation, and $\boldsymbol{x}_i^{\mathrm{r}} \in \mathbb{R}^{d_{\mathrm{r}}}$ denotes the reliability signals. The character-level representation $\boldsymbol{x}_i^{\mathrm{c}}$ consists of two subvectors:

$$\boldsymbol{x}_i^{\mathrm{c}} = \boldsymbol{x}_i^{\mathrm{c_a}} \oplus \boldsymbol{x}_i^{\mathrm{c_c}},$$

where $\oplus$ is the concatenation operator, $\boldsymbol{x}_i^{\mathrm{c_a}} \in \mathbb{R}_i^{d_{\mathrm{w}}}$ acts as an alternative representation to word embedding, and $\boldsymbol{x}^{\mathrm{c_c}} \in \mathbb{R}^{d_{\mathrm{c}}-d_{\mathrm{w}}}$ is concatenated as additional features.

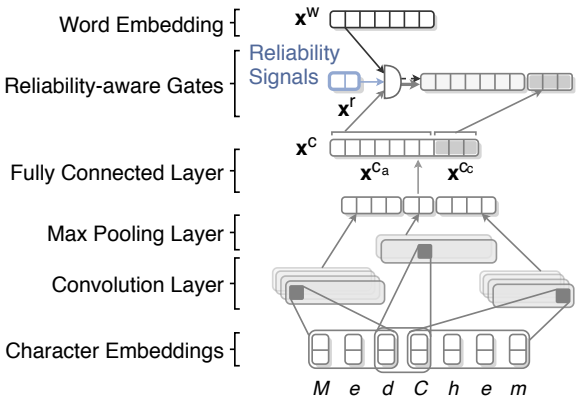

Figure 2: Dynamic feature composition at the word representation level.

In this example, because the word embedding of "MedChem" is not reliable and informative, the model should attend more to $\boldsymbol{x}_i^{\mathrm{c_a}}$. To enable the model to switch between both representations accordingly, we define a pair of reliability-aware gates $\boldsymbol{g}_i^{\mathrm{w}}$ and $\boldsymbol{g}_i^{\mathrm{c}}$ to filter $\boldsymbol{x}_i^{\mathrm{w}}$ and $\boldsymbol{x}_i^{\mathrm{c_a}}$ respectively. We refer to $\boldsymbol{g}_i^{\mathrm{w}}$ as the word-level representation gate and $\boldsymbol{g}_i^{\mathrm{c}}$ as the character-level representation gate. We calculate $\boldsymbol{g}_i^{\mathrm{w}}$ as

$$\boldsymbol{g}_i^{\mathrm{w}} = \sigma(\boldsymbol{W}^{\mathrm{w}}\boldsymbol{x}_i^{\mathrm{w}} + \boldsymbol{W}^{\mathrm{c}}\boldsymbol{x}_i^{\mathrm{c}} + \boldsymbol{W}^{\mathrm{r}}\boldsymbol{x}_i^{\mathrm{r}} + \boldsymbol{b}),$$

where $\boldsymbol{W}^{\mathrm{w}} \in \mathbb{R}^{d_{\mathrm{w}} \times d_{\mathrm{w}}}$, $\boldsymbol{W}^{\mathrm{c}} \in \mathbb{R}^{d_{\mathrm{w}} \times d_{\mathrm{w}}}$, $\boldsymbol{W}^{\mathrm{r}} \in \mathbb{R}^{d_{\mathrm{w}} \times d_{\mathrm{r}}}$, and $\boldsymbol{b} \in \mathbb{R}^{d_{\mathrm{w}}}$ are parameters of the gate. The character-level representation gate $\boldsymbol{g}_i^{\mathrm{c}}$ is defined in the same way.

Finally, the enhanced representation of the i-th word is given by

$$\boldsymbol{x}_i = (\boldsymbol{g}_i^{\mathrm{w}} \circ \boldsymbol{x}_i^{\mathrm{w}} + \boldsymbol{g}_i^{\mathrm{c}} \circ \boldsymbol{x}_i^{\mathrm{c_a}}) \oplus \boldsymbol{x}_i^{\mathrm{c_c}},$$

where $\circ$ denotes the Hadamard product.

We separately calculate $\boldsymbol{g}_i^{\mathrm{w}}$ and $\boldsymbol{g}_i^{\mathrm{c}}$ instead of setting $\boldsymbol{g}_i^{\mathrm{c}} = 1 - \boldsymbol{g}_i^{\mathrm{w}}$ because word- and character-level representations are not always exclusive.

Feature Extraction Level

Although character-level representations can encode semantic information in many cases, they cannot perfectly replace word embeddings. For example, in the following sentence:

"*How does a small town like* ***Linpien*** *come to be home to such a well-organized volunteer effort, and just how did the volunteers set about giving their town a make-over?*"

The surface information of "Linpien" does not provide sufficient clues to infer its meaning and determine whether it is a name. In this case, the model should seek other useful features from the context, such as "a small town like" in the sentence.

However, in our pilot study on OntoNotes, we observe many instances where the model fails to recognize an unseen name even with obvious context clues, along with a huge performance gap in recall between seen (92-96%) and unseen (53-73%) names. A possible reason is that the model can memorize some words without reliable representations in the training set instead of exploiting their contexts in order to reduce the training loss. As a solution to this issue, we encourage the model to leverage contextual features to reduce overfitting to seen names. Compared to names, the context usually consists of more common words. Therefore, contextual features should be more robust when we apply the model to new data.

In LSTM, each hidden state $\boldsymbol{h}_i$ is computed from the previous forward hidden state $\overrightarrow{\boldsymbol{h}}_{i-1}$, next backward hidden state $\overleftarrow{\boldsymbol{h}}_{i+1}$, and the current input $\boldsymbol{x}_i$. To obtain features that are independent of the current input and not affected by its quality, we define context-only features as

$$\boldsymbol{o}_i = \overrightarrow{\boldsymbol{o}}_i \oplus \overleftarrow{\boldsymbol{o}}_i = F(\overrightarrow{\boldsymbol{h}}_{i-1}) \oplus F'(\overleftarrow{\boldsymbol{h}}_{i+1}),$$

where F and F' are affine transformations followed by a non-linear function such that $\boldsymbol{o}_i \in \mathbb{R}^{2d_{\mathrm{h}}}$ has the same dimensionality as $\boldsymbol{h}_i$.

In order to find an optimal mixture of $\boldsymbol{h}_i$ and $\boldsymbol{o}_i$ according to the reliability of representations of the current word and its context words, we define two pairs of gates to control the composition: the forward gates $\overrightarrow{\boldsymbol{g}}_i^{\mathrm{h}}$ and $\overrightarrow{\boldsymbol{g}}_i^{\mathrm{o}}$, and the backward gates $\overleftarrow{\boldsymbol{g}}_i^{\mathrm{h}}$ and $\overleftarrow{\boldsymbol{g}}_i^{\mathrm{o}}$. Figure 3 illustrates how to obtain the forward context-only features $\overrightarrow{\boldsymbol{o}}_i$ and mix it with $\overrightarrow{\boldsymbol{h}}_i$ using reliability-aware gates.

All gates are computed in the same way. Take

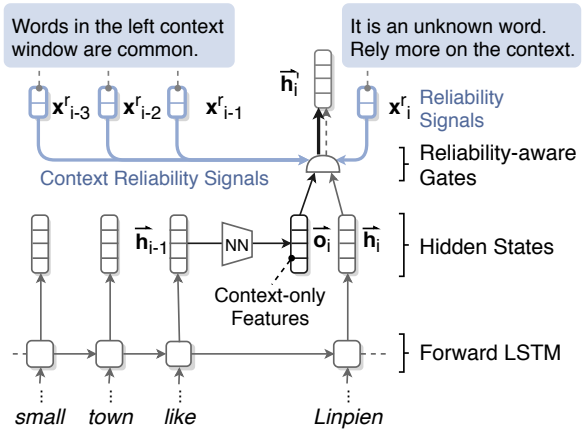

Figure 3: Dynamic feature composition at the feature extraction level. We only show the forward model for the purposes of simplicity.

the forward hidden state gate $\overrightarrow{\boldsymbol{g}}^{\mathrm{h}}_i$ as an example:

$$\overrightarrow{\boldsymbol{g}}^{\mathrm{h}}_i = \sigma(\boldsymbol{U}^{\mathrm{h}}\overrightarrow{\boldsymbol{o}}_i + \boldsymbol{U}^{\mathrm{r}}(\boldsymbol{x}^{\mathrm{r}}_i \oplus ... \oplus \boldsymbol{x}^{\mathrm{r}}_{i-C}) + \boldsymbol{b}'),$$

where $\overrightarrow{\boldsymbol{g}}^{\mathrm{h}}_i$ is parameterized by $\boldsymbol{U}^{\mathrm{h}} \in \mathbb{R}^{d_{\mathrm{h}} \times d_{\mathrm{h}}}$, $\boldsymbol{U}^{\mathrm{r}} \in \mathbb{R}^{d_{\mathrm{h}} \times d_{\mathrm{r}}}$, and $\boldsymbol{b}' \in \mathbb{R}^{d_{\mathrm{h}}}$. This gate is controlled by the previous forward context-only features $\overrightarrow{\boldsymbol{o}_i}$ and reliability signals $(\boldsymbol{x}^{\mathrm{r}}_i \oplus ... \oplus \boldsymbol{x}^{\mathrm{r}}_{i-C})$, where C is the context window size.

By contrast, the backward gates $\overleftarrow{\boldsymbol{g}}^{\mathrm{h}}_i$ and $\overleftarrow{\boldsymbol{g}}^{\mathrm{o}}_i$ take as input the backward context-only features and reliability signals of the right context. With these gates, we incorporate the context-only features by

$$\boldsymbol{h}'_i = (\overrightarrow{\boldsymbol{g}}^{\mathrm{h}}_i \circ \overrightarrow{\boldsymbol{h}}_i + \overrightarrow{\boldsymbol{g}}^{\mathrm{o}}_i \circ \overrightarrow{\boldsymbol{o}}_i) \oplus (\overleftarrow{\boldsymbol{g}}^{\mathrm{h}}_i \circ \overleftarrow{\boldsymbol{h}}_i + \overleftarrow{\boldsymbol{g}}^{\mathrm{o}}_i \circ \overleftarrow{\boldsymbol{o}}_i)$$

The enhanced hidden state $\boldsymbol{h}'_i$ is then decoded by a following linear layer as in the baseline model.

3 Experiment

3.1 Data Sets

We conduct our experiments on OntoNotes 5.0[2] (Weischedel et al., 2013), the final release of the OntoNotes project because it includes six diverse text genres for us to evaluate the robustness of our approach as Table 1 shows.

We adopt the following four common entity types that are also used in other data sets such as TAC-KBP (Ji et al., 2011): PER (person), ORG (organization), GPE (geo-political entity), and LOC (location). We pre-process the data with Pradhan et al.'s scripts[3] and therefore follow their split of training, development, and test sets.

Code	Genre Name	#Sentences		
		Train	Dev	Test
bc	Broadcast conversation	11,866	2,117	2,211
bn	Broadcast news	10,683	1,295	1,357
mz	Magazine	6,911	642	780
nw	Newswire	33,908	5,771	2,197
tc	Telephone conversation	11,162	1,634	1,366
wb	Weblogs	7,592	1,634	1,366

Table 1: OntoNotes genres.

We use the BIOES tag scheme to annotate tags. The S- prefix indicates a single-token name mention. Prefixes B-, I-, and E- mark the beginning, inside, and end of a multi-token name mention. A word that does not belong to any name mention is annotated as O.

3.2 Experimental Setup

We use 100-dimensional word embeddings trained on English Wikipedia articles (2017-12-20 dump) with word2vec, and initialize character embeddings as 50-dimensional random vectors. The character-level convolutional networks have filters of width $[2, 3, 4]$ of size 50.

For the bidirectional LSTM layer, we use a hidden state size of 100. To reduce overfitting, we attach dropout layers (Srivastava et al., 2014) with probability 0.5 to the input and output of the LSTM layer. We use an Adam optimizer with batch size of 20, learning rate of 0.001 and linear learning rate decay.

3.3 Within-genre Results

We use the LSTM-CNN model as our baseline in all experiments. We train and test models on each genre and compare the within-genre results in Table 2. We also merge all genres and show the overall scores in the last column.

Overall, with reliability-aware dynamic feature composition, our model achieves up to 6.2% absolute F-score gain on separate genres. T-test results show that the differences are considered to be statistically significant ($p < 0.05$) to statistically highly significant ($p < 0.001$).

In Figure 4, we visualize gates that control the mixture of hidden states and context-only features. Each block represents the average of output weights of a certain gate for the correspond-

[2] https://catalog.ldc.upenn.edu/LDC2013T19

[3] https://cemantix.org/data/ontonotes.html

	bc	bn	mz	nw	tc	wb	all
LSTM-CNN	83.5	89.9	86.6	92.8	65.4	**79.4**	90.1
Rei et al. (2016)	85.4	90.4	87.2	92.5	71.1	77.4	90.0
Our Model*	86.2	91.2	89.8	92.9	71.3	78.5	90.3
Our Model	**86.4**	**91.4**	**90.0**	**93.0**	**71.6**	79.1	**90.6**

Table 2: Performance on OntoNotes (F-score, %). Our Model* is a variant of our model that does not incorporate reliability signals. (Rei et al., 2016) uses a gate to control the mixture of character- and word-level representations.

ing word. The results of hidden state gates $\overrightarrow{g}^{h}$ and $\overleftarrow{g}^{h}$ show that for common words such as "a" and "to", the model mainly relies on their original hidden states. By contrast, the context-only feature gates $\overrightarrow{g}^{o}$ and $\overleftarrow{g}^{o}$ assign greater weights to the unknown word "Linpien". Meanwhile, the model barely uses any context-only features for words following "Linpien" ("come" in the forward model and "like" in the backward model) to avoid using unreliable features derived from an unknown word.

To our surprise, the model also emphasizes context-only features for the beginning and ending words. Their context-only features actually come from the zero vectors padded to the sequence during gate calculation. Our explanation is that these features may help the model distinguish the beginning and ending words that differ from other words in some aspects. For example, capitalization is usually an indicator of proper nouns for most words except for the first word of a sentence.

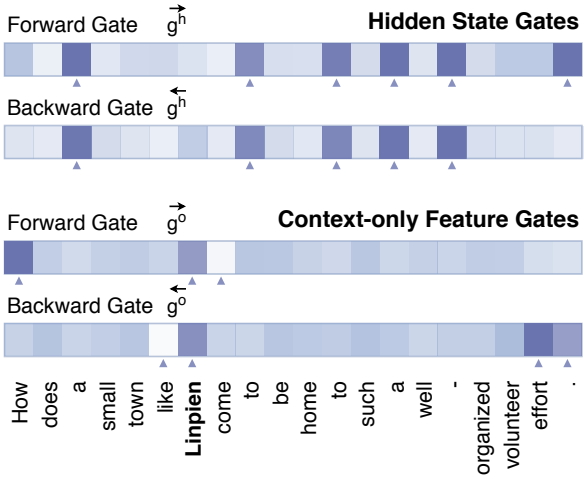

Figure 4: Visualization of reliability-aware gates. A darker color indicates a higher average weight.

3.4 Cross-genre Results

Different genres in OntoNotes not only differ in style but also cover different topics and hence different names. As Table 3 shows, when tested on another genre, the model encounters a high percentage of names that are unseen in the training genre. For example, 81.3% names are unseen when we train a model on mz and test it on bc. Therefore, through cross-genre experiments, we can evaluate the generalization capability of the model.

Train \ Test	bc	bn	mz	nw	tc	wb
bc	36.3	53.4	73.2	68.9	81.4	51.5
bn	43.9	28.5	72.8	63.6	67.8	49.9
mz	81.3	79.8	41.1	82.1	88.1	86.4
nw	40.2	43.8	70.8	33.1	55.4	55.1
tc	82.4	83.2	93.4	87.0	67.8	79.0
wb	54.6	60.6	75.4	70.8	85.3	53.4

Table 3: High percentage of unseen names (%).

Baseline Model						
Train \ Test	bc	bn	mz	nw	tc	wb
bc	83.5	82.4	70.4	67.9	**74.8**	75.2
bn	83.5	89.9	**78.7**	75.6	**76.8**	**77.1**
mz	59.2	70.7	86.6	65.9	**66.1**	58.0
nw	**82.4**	85.4	72.6	92.8	74.4	76.7
tc	53.2	51.2	34.0	38.9	65.4	44.3
wb	71.5	78.1	67.5	66.6	70.1	**79.4**
Our Model						
Train \ Test	bc	bn	mz	nw	tc	wb
bc	**86.4**	**82.5**	**76.4**	**70.6**	74.7	**76.1**
bn	**84.8**	**91.4**	**78.7**	**79.2**	76.5	76.1
mz	**64.3**	**73.8**	**90.0**	**70.5**	**57.5**	**59.3**
nw	81.5	**86.1**	**74.0**	**93.0**	**74.9**	**78.3**
tc	**58.2**	**55.6**	**43.6**	**47.1**	**71.6**	**50.4**
wb	**76.3**	**78.4**	**70.5**	**69.6**	**72.3**	79.1

Table 4: Cross-genre performance on OntoNotes (F-score, %).

In Table 4, we compare the cross-genre performance between the baseline and our model. For most cross-genre pairs, our model outperforms the baseline and obtains up to 9.6% absolute gains in F-score.

With dynamic feature composition, the cross-genre performance of our model even exceeds the within-genre performance of the baseline model in some cases. For example, when trained on the bn portion and tested on bc, our model achieves 84.8% F-score, which is 1.3% higher than the

within-genre performance of the baseline model (83.5% F-score). Such generalization capability is important for real-word applications as it is infeasible to annotate training data for all possible scenarios.

3.5 Qualitative Analysis

In Table 5, we show some typical name tagging errors corrected by our model. We highlight the difference between the outputs of the baseline model and our model in bold. We also underline words that probably have provided useful contextual clues.

Identification Errors
⋆ BASELINE:
The 50-50 joint venture, which may be dubbed **Euro-dynamics** , would have combined annual sales of at least #1.4 billion ($2.17 billion) and would be among the world's largest missile makers.
⋆ OUR MODEL:
The 50-50 joint venture, which may be dubbed [ORG **Eurodynamics**] , would have combined annual sales of at least #1.4 billion ($2.17 billion) and would be among the world's largest missile makers.
⋆ BASELINE:
The **Tanshui** of illustrations is a place of unblemished beauty, a myth that remains unshakeable.
⋆ OUR MODEL:
The [GPE **Tanshui**] of illustrations is a place of unblemished beauty, a myth that remains unshakeable.
Classification Errors
⋆ BASELINE:
As [PER **Syms**] 's "core business of off-price retailing grows, a small subsidiary that is operationally unrelated becomes a difficult distraction," said [PER Marcy Syms], president of the parent, in a statement.
⋆ OUR MODEL:
As [ORG **Syms**] 's "core business of off-price retailing grows, a small subsidiary that is operationally unrelated becomes a difficult distraction," said [PER Marcy Syms], president of the parent, in a statement.
⋆ BASELINE:
Workers at plants in [GPE Van Nuys] , [GPE Calif.] , [GPE Oklahoma City] and [ORG **Pontiac**] , [GPE Mich.] , were told their facilities are no longer being considered to build the next generation of the [ORG Pontiac] Firebird and [ORG Chevrolet] Camaro muscle cars.
⋆ OUR MODEL:
Workers at plants in [GPE Van Nuys] , [GPE Calif.] , [GPE Oklahoma City] and [GPE **Pontiac**] , [GPE Mich.] , were told their facilities are no longer being considered to build the next generation of the [ORG Pontiac] Firebird and [ORG Chevrolet] Camaro muscle cars.

Table 5: Name tagging result comparison between the baseline model and our model.

Character-level representations are particularly effective for words containing morphemes that are related to a certain type of names. For example, "Eurodynamics" in the first sentence consists of "Euro-" and "dynamic". The prefix "Euro-" often appears in European organization names such as "EuroDisney" (an entertainment resort) and "EuroAtlantic" (an airline), while "dynamic" is used in some company names such as Boston dynamics (a robotics company) and Beyerdynamic (an audio equipment manufacturer). Therefore, "Eurodynamics" is likely to be an organization rather than a person or location.

However, for words like "Tanshui" (a town) in the second example, character-level representations may not provide much useful semantic information. In this case, contextual features ("is a place") play an important role in determining the type of this name.

Contextual features can be critical even for frequent names such as "Jordan" (can be a person or a country) and "Thomson" (can be various types of entities, including person, organization, city, and river). Take the third sentence in Table 5 as an example. The name "Syms" appears twice in the sentence, referring to the Syms Corp and Marcy Syms respectively. As they share the same word- and character-level representations, context clues such as "core business" and "president" are crucial to distinguish them. Similarly, "Pontiac" in the last example can be either a city or a car brand. Cities in its context (e.g., "Van Nuys, Calif", "Oklahoma City") help the model determine that the first "Pontiac" is more likely to be a GPE instead of an ORG.

Still, the contextual information utilized by the current model is not profound enough, and our model is not capable of conducting deep reasoning as human readers. For example, in the following sentence:

"*In the middle of the 17th century the Ming dynasty loyalist* ***Zheng Chenggong*** *(also known as* ***Koxinga****) brought an influx of settlers to Taiwan from the Fujian and Guangdong regions of China.*"

Although our model successfully identifies "Zheng Chenggong" as a person, it is not able to connect this name with "Koxinga" based on the expression "also known as" to further infer that "Koxinga" should also be a person.

4 Related Work

Name Tagging Models

Most existing methods treat name tagging as a sequence labeling task. Traditional methods leverage handcrafted features to capture textual signals and employ conditional random fields (CRF) to model label dependencies (Finkel et al., 2005; Settles, 2004; Leaman et al., 2008).

Bi-LSTM-CRF (Huang et al., 2015) combines word embedding and handcrafted features, integrates neural networks with CRF, and shows performance boost over previous methods. LSTM-CNN further utilizes CNN and illustrates the potential of capturing character-level signals (Chiu and Nichols, 2016). LSTM-CRF and LSTM-CNNs-CRF are proposed to get rid of hand-crafted features and demonstrate the feasibility to fully rely on representation learning to capture textual features (Lample et al., 2016; Ma and Hovy, 2016; Liu et al., 2018b). Recently, language modeling methods are proven effective as the representation module for name tagging (Liu et al., 2018a; Peters et al., 2018; Akbik et al., 2018). At the same time, there has been extensive research about cross-genre (Peng and Dredze, 2017), cross-domain (Pan et al., 2013; He and Sun, 2017), cross-time (Mota and Grishman, 2008), cross-task (Søgaard and Goldberg, 2016; Liu et al., 2018b), and cross-lingual (Yang et al., 2017; Lin et al., 2018) adaptation for name tagging training.

Unlike these models, although we also aim to enhance the performance on new data, we achieve this by improving the generalization capability of the model so that it can work better on unknown new data instead of transferring it to a known target setting.

Word Representation Models

Recent advances on representation learning allow us to capture textual signals in a data-driven manner. Based on the distributional hypothesis (i.e., "a word is characterized by the company it keeps" (Harris, 1954)), embedding methods represent each word as a dense vector, while preserving their syntactic and semantic information in a context-agnostic manner (Mikolov et al., 2013; Pennington et al., 2014). Recent work shows that word embeddings can cover textual information of various levels (Artetxe et al., 2018) and improve name tagging performance significantly (Cherry and Guo, 2015). Still, due to the long-tail distribution of word frequency, embedding vectors usually have inconsistent reliability, and such inconsistency has been long overlooked.

Meanwhile, language models such as ELMo, Flair, and BERT have shown their effectiveness on constructing representations in a context-aware manner (Peters et al., 2018; Akbik et al., 2018; Devlin et al., 2018). These models are designed to better capture the context information by pre-training, while our model dynamically composes representations in a reliability-aware manner. Therefore, our model and these efforts have the potential to mutually enhance each other.

In addition, (Kim et al., 2016) and (Rei et al., 2016) also mix word- and character-level representations using gating mechanisms. They use a single gate to balance the representations in a reliability-agnostic way.

5 Conclusions and Future Work

We propose a name tagging model that is able to dynamically compose features depending on the quality of input word embeddings. Experiments on the benchmark data sets in both within-genre and cross-genre settings demonstrate the effectiveness of our model and verify our intuition to introduce reliability signals.

Our future work includes integrating advanced word representation methods (e.g., ELMo and BERT) and extending the proposed model to other tasks, such as event extraction and co-reference resolution. We also plan to incorporate external knowledge and common sense as additional signals into our architecture as they are important for human readers to recognize names but still absent from the current model.

Acknowledgments

This work was supported by the U.S. DARPA AIDA Program No. FA8750-18-2-0014, LORELEI Program No. HR0011-15-C-0115, Air Force No. FA8650-17-C-7715, U.S. ARL NS-CTA No. W911NF-09-2-0053, and Tencent AI Lab Rhino-Bird Gift Fund. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the International Conference on Computational Linguistics (COLING 2018)*.

Mikel Artetxe, Gorka Labaka, Inigo Lopez-Gazpio, and Eneko Agirre. 2018. Uncovering divergent linguistic information in word embeddings with lessons for intrinsic and extrinsic evaluation. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2018)*.

Amir Bakarov. 2018. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for twitter named entity recognition. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2015)*.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association of Computational Linguistics*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of The Annual Meeting of the Association for Computational Linguistics (ACL 2015)*.

J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*.

Zellig S Harris. 1954. Distributional structure. *Word*.

Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI 2017)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. An overview of the tac2011 knowledge base population track. In *Proceedings of the Text Analysis Conference (TAC 2011)*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI Conference on Artificial Intelligence (AAAI 2016)*.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on International Conference on Machine Learning (ICML 2001)*.

Siwei Lai, Liheng Xu, Kang Liu, and Jian Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI 2015)*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2016)*.

Robert Leaman, Graciela Gonzalez, et al. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.

Liyuan Liu, Xiang Ren, Jingbo Shang, Jian Peng, and Jiawei Han. 2018a. Efficient contextualized representation: Language model pruning for sequence labeling. In *EMNLP*.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018b. Empower sequence labeling with task-aware neural language model. In *AAAI Conference on Artificial Intelligence (AAAI 2018)*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of The Annual Meeting of the Association for Computational Linguistics*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Cristina Mota and Ralph Grishman. 2008. Is this NE tagger getting old? In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2008)*.

Sinno Jialin Pan, Zhiqiang Toh, and Jian Su. 2013. Transfer joint embedding for cross-domain named entity recognition. *ACM Transactions on Information Systems*.

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2018)*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2013)*.

Marek Rei, Gamal Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. In *Proceedings of International Conference on Computational Linguistics (COLING 2016)*.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of The Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

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

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2014)*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 LDC2013T19. *Linguistic Data Consortium, Philadelphia, PA*.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *Proceedings of International Conference on Learning Representations (ICLR 2017)*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*.