# A Local Detection Approach for Named Entity Recognition and Mention Detection

**Mingbin Xu**

Joint work with **H. Jiang** and **S. Watcharawittayakul**

Lassonade School of Engineering, York University, Canada

ACL2017





A Local Detection Approach for Named Entity Recognition and Mention Detection
Mingbin Xu
Joint work with H. Jiang and S. Watcharawittayakul
Lassonade School of Engineering, York University, Canada
ACL2017

Good morning.
My name is Mingbin. I am from York University.
This presentation is to report the paper A LOCAL DETECTION APPROACH FOR NAMED ENTITY RECOGNTION AND MENTION DETECTION

## Entity Discovery

**Definition**

A sub-task of information extraction that **finds** and **classifies** entities in text.

**Example (CoNLL2003 annotation)**

[*Hinton*]$_{PER}$, a professor of [*University of Toronto*]$_{ORG}$, spends several months in [*Google*]$_{ORG}$'s [*Mountain View*]$_{LOC}$ office every year.

| | |
|---|---|
| PER | Person |
| LOC | Location |
| ORG | Organization |
| MISC | Miscellaneous |





Entity Discovery
Definition
A sub-task of information extraction that finds and classifies entities in text.
Example (CoNLL2003 annotation)
[Hinton]PER, a professor of [University of Toronto]ORG, spends several months in [Google]ORG's [Mountain View]LOC office every year.
PER Person
LOC Location
ORG Organization
MISC Miscellaneous

NER & MD sometimes is called entity discovery.
It is a sub-task of information retrieval that finds and classifies entities in text.
For example, in this sentence (read slides),
Hinton, a professor of UofT, spends several months in Google's Mountain View office every year.
we want to tell Hinton is a PER, Google is an ORG, and Mountain View is a LOC.

# Entity Discovery

**Definition**

A sub-task of information extraction that **finds** and **classifies** entities in text.

**Example (KBP EDL annotation)**

*[Hinton]*$_{PER\text{-}NAM}$, a *[professor]*$_{PER\text{-}NOM}$ of *[University of [Toronto]*$_{GPE\text{-}NAM}$*]*$_{ORG\text{-}NAM}$, spends several months in *[Google]*$_{ORG\text{-}NAM}$'s *[Mountain View]*$_{LOC\text{-}NAM}$ office every year.

PER-{NAME, NOMINAL} Person
LOC-{NAME, NOMINAL} Location
ORG-{NAME, NOMINAL} Organization
GPE-{NAME, NOMINAL} Geo-Political Entity
FAC-{NAME, NOMINAL} Facility





Entity Discovery
Definition
A sub-task of information extraction that finds and classifies entities in text.
Example (KBP EDL annotation)
[Hinton]PER-NAM, a [professor]PER-NOM of [University of [Toronto]GPE-NAM]ORG-NAM, spends several months in [Google]ORG-NAM's [Mountain View]LOC-NAM office every year.
PER-{NAME, NOMINAL} Person
LOC-{NAME, NOMINAL} Location
ORG-{NAME, NOMINAL} Organization
GPE-{NAME, NOMINAL} Geo-Political Entity
FAC-{NAME, NOMINAL} Facility

Some task is more difficult.

We may need to detect nested mentions, e.g. Toronto is embeded in UofT. We need to find out Toronto is GPE and UofT is ORG.

We may also need to detect nominal mentions, e.g. We need to know "professor" refers to a person in real world.

# Review

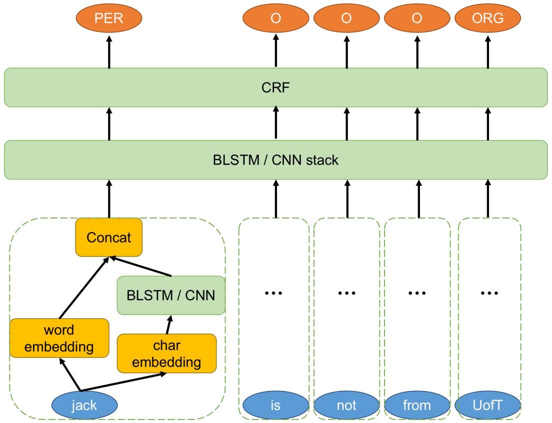
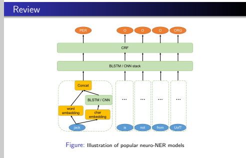



Figure: Illustration of popular neuro-NER models







A common solution to this problem is sequence labeling.
Each word in a sentence is modeled by word embedding and either CNN or LSTM.
The sentence is modeled by either CNN or LSTM, and decoded by CRF.

# Fixed-size Ordinally Forgetting Encoding

## Definition (FOFE)

- $S = w_1, w_2, ..., w_n$ is a sequence of any discrete symbols;
- $w_i$ is represented as $\boldsymbol{e_i}$ in 1-hot representation;
- the encoding of a partial sequence up to the $t$-th word is recursively defined as (Zhang et al. 2015):

$$\boldsymbol{z_t} = \begin{cases} \boldsymbol{0}, & \text{if } t = 0 \\ \alpha \cdot \boldsymbol{z_{t-1}} + \boldsymbol{e_t}, & \text{otherwise} \end{cases}$$

- $\alpha \in (0, 1)$ and $t \in \{\mathbb{Z} | 1 \leq x \leq n\}$





Fixed-size Ordinally Forgetting Encoding
Definition (FOFE)
- $S = w_1, w_2, ..., w_n$ is a sequence of any discrete symbols;
- $w_i$ is represented as $\boldsymbol{e_i}$ in 1-hot representation;
- the encoding of a partial sequence up to the $t$-th word is recursively defined as (Zhang et al. 2015):
$$\boldsymbol{z_t} = \begin{cases} \boldsymbol{0}, & \text{if } t = 0 \\ \alpha \cdot \boldsymbol{z_{t-1}} + \boldsymbol{e_t}, & \text{otherwise} \end{cases}$$
- $\alpha \in (0, 1)$ and $t \in \{\mathbb{Z} | 1 \leq x \leq n\}$

Instead of CNN and LSTM, we adpots another sequence modeling method.
It's Fixed-size Orinally Forgetting Encoding, or FOFE.
Let's say we have a sequence of $n$ symbols, $w1, w2, ..., w_n$.
Each symbol is represented in a 1-hot vector.
The encoding of a partial sequence up to the current symbol is the partial encoding up to the previous symbol times $\alpha$ plus the 1-hot vector of the current symbol.

$\alpha$ is called forgetting factor. It's usually picked between 0 and 1.

# Fixed-size Ordinally Forgetting Encoding

Any **variable length** sequence is encoded into a **fixed-size** vector.

| WORD | 1-HOT |
|---|---|
| $w_0$ | 1000000 |
| $w_1$ | 0100000 |
| $w_2$ | 0010000 |
| $w_3$ | 0001000 |
| $w_4$ | 0000100 |
| $w_5$ | 0000010 |
| $w_6$ | 0000001 |

Table: Vocab of size 7

| PARTIAL SEQUENCE | FOFE |
|---|---|
| $w_6$ | $0, 0, 0, 0, 0, 0, 1$ |
| $w_6, w_4$ | $0, 0, 0, 0, 1, 0, \alpha$ |
| $w_6, w_4, w_5$ | $0, 0, 0, 0, \alpha, 1, \alpha^2$ |
| $w_6, w_4, w_5, w_0$ | $1, 0, 0, 0, \alpha^2, \alpha, \alpha^3$ |
| $w_6, w_4, w_5, w_0, w_5$ | $\alpha, 0, 0, 0, \alpha^3, 1+\alpha^2, \alpha^4$ |
| $w_6, w_4, w_5, w_0, w_5, w_4$ | $\alpha^2, 0, 0, 0, 1+\alpha^4, \alpha+\alpha^3, \alpha^5$ |

Table: Partial encoding of $w_6, w_4, w_5, w_0, w_5, w_4$





Since the encoding is the weighted sum of each symbol, its size depends on vocabulory size only.

FOFE is able to encode any sequence into a fixed-size vector.

# Uniqueness of FOFE

**Theorem**

*If the forgetting factor $\alpha$ satisfies $0 < \alpha \leq 0.5$, FOFE is unique for any countable vocabulary $V$ and any finite value $T$.*

**Theorem**

*For $0.5 < \alpha < 1$, given any finite value $T$ and any countable vocabulary $V$, FOFE is almost unique everywhere, except only a finite set of countable choices of $\alpha$.*





Uniqueness of FOFE
Theorem
If the forgetting factor $\alpha$ satisfies $0 < \alpha \leq 0.5$, FOFE is unique for any countable vocabulary $V$ and any finite value $T$.
Theorem
For $0.5 < \alpha < 1$, given any finite value $T$ and any countable vocabulary $V$, FOFE is almost unique everywhere, except only a finite set of countable choices of $\alpha$.

FOFE has a very nice uniqueness property.
(READ SLIDES)
if $\alpha$ is between 0 and 0.5, FOFE is unique for any countable vocabulary V and any finite value T.
if $\alpha$ is between 0.5 and 1, given any finite value T and countable vocabulary V, FOFE is almost unique, except a finite set of countable choices of $\alpha$.

Therefore, FOFE is a lossless fixed-size representation for sequence modeling.

# Computational Efficiency of FOFE

## LSTM

One step at a time; each involves 4 matrix multications.

$$x = oneHot([w_1, w_2, ..., w_n]) \times W_{embed}$$
$$C_t, h_t = LSTM(x_t, C_{t-1}, h_{t_1})$$
$$enc([w_1, ..., w_n]) = C_n$$

## FOFE

A single matrix multiplication leads to the final encoding.

$$alpha = [\alpha^{n-1}, \alpha^{n-2}, ..., \alpha, 1]$$
$$enc([w_1, ..., w_n]) = (alpha \times oneHot([w_1, w_2, ..., w_n])) \times W_{embed}$$
$$= alpha \times (oneHot([w_1, w_2, ..., w_n]) \times W_{embed})$$



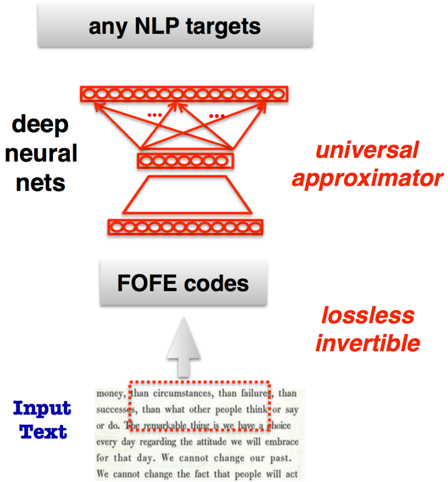


Computational Efficiency of FOFE
LSTM
One step at a time; each involves 4 matrix multications.
$x = oneHot([w_1, w_2, ..., w_n]) \times W_{embed}$
$C_t, h_t = LSTM(x_t, C_{t-1}, h_{t_1})$
$enc([w_1, ..., w_n]) = C_n$
FOFE
A single matrix multiplication leads to the final encoding.
$alpha = [\alpha^{n-1}, \alpha^{n-2}, ..., \alpha, 1]$
$enc([w_1, ..., w_n]) = (alpha \times oneHot([w_1, w_2, ..., w_n])) \times W_{embed}$
$= alpha \times (oneHot([w_1, w_2, ..., w_n]) \times W_{embed})$

FOFE is significantly faster than LSTM.
e.g. In LSTM, we first get the embedding of each word. It's conceptually a matrix multiplication. It is implemented as table lookup. Then, the encoding at each postion must be computed step by step. Each step consists of 4 matrix multiplications.

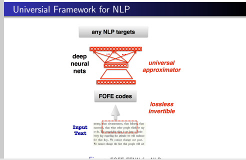
FOFE is equivalent to a vector of geometric series times the matrix of one-hot vectors. Similarly, word embedding is used to reduce dimensions. Because of associativity of matrix multiplication, we can do table lookup first to get a much smaller matrix. A single matrix multiplication leads to the final encoding of the sentence.

# Universal Framework for NLP







Because FOFE is lossless fixed-size representation and FFNN is a universial approximator,
FOFE plus FFNN serves as a universail framework for NLP.

# Local Detection

## Intuition

- People rarely conduct a global decoding over the entire sentence to pinpoint entities.
- The key to accurate local detection is to have full access to **the fragment itself**, and **its contextual information**.
- FOFE is a **lossless** representation of **fixed length**.

## Example

- $[S.E.C.]_{ORG}$ chief $[Mary\ Shapiro]_{PER}$ left $[Washington]_{LOC}$ in December.
- Do the entity types of "S.E.C" and "Washington" matter?
  How about: **Our** chief Mary Shapiro left **us** in December?





Local Detection
Intuition
People rarely conduct a global decoding over the entire sentence to pinpoint entities.
The key to accurate local detection is to have full access to the fragment itself, and its contextual information.
FOFE is a lossless representation of fixed length.
Example
[S.E.C.]ORG chief [Mary Shapiro]PER left [Washington]LOC in December.
Do the entity types of "S.E.C" and "Washington" matter? How about: Our chief Mary Shapiro left us in December?

Here's our LOCAL DETECTION algorithm.
The intution behind this idea is that:
(read slides)
People rarely conduct a global decoding over the entire sentence to find and classify entities.
The key to accurate local detection is to have full access to the text fragment itself, and its contextual information.
Let's say we have a sentence S.E.C chief Mayr Shapiro left Washington in December, and we're interested in the text fragment May Shapiro in this sentence.
As long as we know it is a cheif and it can perform an action of "left", we can tell it's a person. Whether S.E.C is an ORG or not and whether Washington is a LOC or not do not affect our decision.
(read slides)
We picked FOFE as our method of modeling these two pieces because FOFE is a lossless representation of fixed length.

## Algorithm

Our methods treats the **whole sentence** as context to make a local decision for each text fragment.

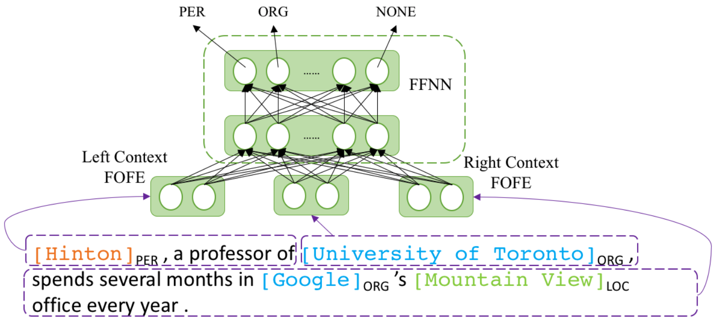
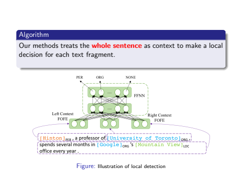



Figure: Illustration of local detection







Unlike previous local detection approach, our approach treats the entire sentence as context.
The local decision is made based on global information.
**(Next)**
Our Local Detection approach extracts features from each segment and sends them to an FFNN.
Let's say the text fragment we're interested in is "UofT".
It divides the sentence into 3 disjoint sub-sequences.
Everything to the left of UofT is called left context.
Everything to the right of UofT is called right context.
Becuase these 3 pieces are sequences of words. They can be easily modeled by FOFE.
Because FOFE is fixed-size, we pick FFNN as our classifier.

## Algorithm

- Extract features from **each segment** and send to FFNN.
- Remove overlapping / inconsistent labels.

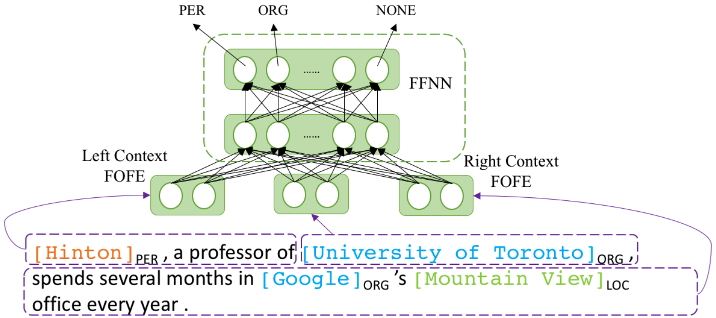
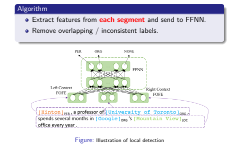



Figure: Illustration of local detection



2017-08-02
A Local Detection Approach for NER & MD
11/25
└Local Detection Algorithm
└Algorithm



Unlike previous local detection approach, our approach treats the entire sentence as context.
The local decision is made based on global information.
**(Next)**
Our LOCAL DETECTION approach extracts features from each segment and sends them to an FFNN.
Let's say the text fragment we're interested in is "UofT".
It divides the sentence into 3 disjoint sub-sequences.
Everything to the left of UofT is called left context.
Everything to the right of UofT is called right context.
Becuase these 3 pieces are sequences of words. They can be easily modeled by FOFE.
Because FOFE is fixed-size, we pick FFNN as our classifier.

# Feature Extraction

## Feature Extraction

| | text segment | context |
|---|---|---|
| word level | BoW | left FOFE excl. text fragment |
| | | right FOFE excl. text fragment |
| | left FOFE incl. text fragment | |
| | right FOFE incl. text fragment | |
| char level | Char CNN (Kim et al. 2016)<br>Char FOFE | N/A |

## Example (examining 'Mary Shapiro')

$[S.E.C.]_{ORG}$ chief $[Mary\ Shapiro]_{PER}$ left $[Washington]_{LOC}$ in December.





we model the text fragment at word level and character level.
Let's say we're interested in the text fragment "Mary Shapiro" in this sentence. **(Next)**
The text fragment is first modeled by BoW. **(Next)**
At the same time, it can be viewed as a character sequence. So we can construct a bi-directional FOFE to model its internal structure. **(Next)**
Similary, we apply character CNN as well. **(Next)**
In terms of context feature, we build 2 FOFE represnetation for left context and right context respectively. **(Next)**
Finally, in order to emphize the order of words in the text fragment and their relationship to the context, we create 2 more FOFE representation. One is from the start of the sentence to the end of the text fragment. The other one is from the end ofthe sentence to the start of the text fragment.

# Feature Extraction

## Feature Extraction

| | text segment | context |
|---|---|---|
| word level | BoW | left FOFE excl. text fragment |
| | | right FOFE excl. text fragment |
| | left FOFE incl. text fragment | |
| | right FOFE incl. text fragment | |
| char level | Char CNN (Kim et al. 2016)<br>Char FOFE | N/A |

## Example (examining 'Mary Shapiro')

$[S.E.C.]_{ORG}$ chief $[Mary\ Shapiro]_{PER}$ left $[Washington]_{LOC}$ in December.





we model the text fragment at word level and character level.
Let's say we're interested in the text fragment "Mary Shapiro" in this sentence. **(Next)**
The text fragment is first modeled by BoW. **(Next)**
At the same time, it can be viewed as a character sequence. So we can construct a bi-directional FOFE to model its internal structure. **(Next)**
Similary, we apply character CNN as well. **(Next)**
In terms of context feature, we build 2 FOFE represnetation for left context and right context respectively. **(Next)**
Finally, in order to emphize the order of words in the text fragment and their relationship to the context, we create 2 more FOFE representation.
One is from the start of the sentence to the end of the text fragment.
The other one is from the end ofthe sentence to the start of the text fragment.

# Feature Extraction

## Feature Extraction

| | text segment | context |
|---|---|---|
| word level | BoW | left FOFE excl. text fragment |
| | | right FOFE excl. text fragment |
| | left FOFE incl. text fragment | |
| | right FOFE incl. text fragment | |
| char level | Char CNN (Kim et al. 2016)<br>Char FOFE | N/A |

## Example (examining 'Mary Shapiro')

*[S.E.C.]$_{ORG}$* chief *[Mary Shapiro]$_{PER}$* left *[Washington]$_{LOC}$* in December.





we model the text fragment at word level and character level.
Let's say we're interested in the text fragment "Mary Shapiro" in this sentence. **(Next)**
The text fragment is first modeled by BoW. **(Next)**
At the same time, it can be viewed as a character sequence. So we can construct a bi-directional FOFE to model its internal structure. **(Next)**
Similary, we apply character CNN as well. **(Next)**
In terms of context feature, we build 2 FOFE represnetation for left context and right context respectively. **(Next)**
Finally, in order to emphize the order of words in the text fragment and their relationship to the context, we create 2 more FOFE representation.
One is from the start of the sentence to the end of the text fragment.
The other one is from the end ofthe sentence to the start of the text fragment.

# Feature Extraction

## Feature Extraction

| | text segment | context |
|---|---|---|
| word level | BoW | left FOFE excl. text fragment |
| | | right FOFE excl. text fragment |
| | left FOFE incl. text fragment | |
| | right FOFE incl. text fragment | |
| char level | Char CNN (Kim et al. 2016)<br>Char FOFE | N/A |

## Example (examining 'Mary Shapiro')

[*S.E.C.*]$_{ORG}$ chief [*Mary Shapiro*]$_{PER}$ left [*Washington*]$_{LOC}$ in December.





we model the text fragment at word level and character level.
Let's say we're interested in the text fragment "Mary Shapiro" in this sentence. **(Next)**
The text fragment is first modeled by BoW. **(Next)**
At the same time, it can be viewed as a character sequence. So we can construct a bi-directional FOFE to model its internal structure. **(Next)**
Similary, we apply character CNN as well. **(Next)**
In terms of context feature, we build 2 FOFE represnetation for left context and right context respectively. **(Next)**
Finally, in order to emphize the order of words in the text fragment and their relationship to the context, we create 2 more FOFE representation. One is from the start of the sentence to the end of the text fragment. The other one is from the end ofthe sentence to the start of the text fragment.

# Feature Extraction

## Feature Extraction

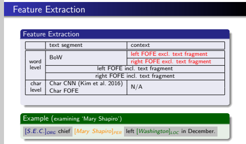
| | text segment | context |
|---|---|---|
| word level | BoW | left FOFE excl. text fragment |
| | | right FOFE excl. text fragment |
| | left FOFE incl. text fragment | |
| | right FOFE incl. text fragment | |
| char level | Char CNN (Kim et al. 2016)<br>Char FOFE | N/A |

## Example (examining ‘Mary Shapiro’)

$[S.E.C.]_{ORG}$ chief $[Mary\ Shapiro]_{PER}$ left $[Washington]_{LOC}$ in December.







we model the text fragment at word level and character level.
Let's say we're interested in the text fragment “Mary Shapiro” in this sentence. **(Next)**
The text fragment is first modeled by BoW. **(Next)**
At the same time, it can be viewed as a character sequence. So we can construct a bi-directional FOFE to model its internal structure. **(Next)**
Similary, we apply character CNN as well. **(Next)**
In terms of context feature, we build 2 FOFE represnetation for left context and right context respectively. **(Next)**
Finally, in order to emphize the order of words in the text fragment and their relationship to the context, we create 2 more FOFE representation. One is from the start of the sentence to the end of the text fragment. The other one is from the end ofthe sentence to the start of the text fragment.

# Feature Extraction

## Feature Extraction

| | text segment | context |
|---|---|---|
| word level | BoW | left FOFE excl. text fragment |
| | | right FOFE excl. text fragment |
| | left FOFE incl. text fragment | |
| | right FOFE incl. text fragment | |
| char level | Char CNN (Kim et al. 2016)<br>Char FOFE | N/A |

## Example (examining 'Mary Shapiro')

[*S.E.C.*]$_{ORG}$ chief [*Mary Shapiro*]$_{PER}$ left [*Washington*]$_{LOC}$ in December.





we model the text fragment at word level and character level.
Let's say we're interested in the text fragment "Mary Shapiro" in this sentence. **(Next)**
The text fragment is first modeled by BoW. **(Next)**
At the same time, it can be viewed as a character sequence. So we can construct a bi-directional FOFE to model its internal structure. **(Next)**
Similary, we apply character CNN as well. **(Next)**
In terms of context feature, we build 2 FOFE represnetation for left context and right context respectively. **(Next)**
Finally, in order to emphize the order of words in the text fragment and their relationship to the context, we create 2 more FOFE representation.
One is from the start of the sentence to the end of the text fragment.
The other one is from the end ofthe sentence to the start of the text fragment.

# Feature Extraction

## Feature Extraction

| | text segment | context |
|---|---|---|
| word level | BoW | left FOFE excl. text fragment |
| | | right FOFE excl. text fragment |
| | left FOFE incl. text fragment | |
| | right FOFE incl. text fragment | |
| char level | Char CNN (Kim et al. 2016)<br>Char FOFE | N/A |

## Example (examining 'Mary Shapiro')

$[S.E.C.]_{ORG}$ chief $[Mary\ Shapiro]_{PER}$ left $[Washington]_{LOC}$ in December.





we model the text fragment at word level and character level.
Let's say we're interested in the text fragment "Mary Shapiro" in this sentence. **(Next)**
The text fragment is first modeled by BoW. **(Next)**
At the same time, it can be viewed as a character sequence. So we can construct a bi-directional FOFE to model its internal structure. **(Next)**
Similary, we apply character CNN as well. **(Next)**
In terms of context feature, we build 2 FOFE represnetation for left context and right context respectively. **(Next)**
Finally, in order to emphize the order of words in the text fragment and their relationship to the context, we create 2 more FOFE representation.
One is from the start of the sentence to the end of the text fragment.
The other one is from the end ofthe sentence to the start of the text fragment.

# Model Advantages

## Data Availability

Correct annotation for the entire sentence is NOT a must.

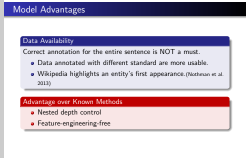
- Data annotated with different standard are more usable.
- Wikipedia highlights an entity's first appearance.(Nothman et al. 2013)

## Advantage over Known Methods

- Nested depth control
- Feature-engineering-free







Our model has several advantages over existing solutions.
Since it's local detection, correct annotation for the entire sentence is not a must. As long as one entity is correctly labeled, it can be used as a training example.
So, data annotated with different standard are more usable.
We can also generate high-quality training example easily. Wikipedia highlights an entity's first appearance in an article. The entity spanning is well-defined by the its hyper-link. This kind of machine-generated data is very accurate locally.
Our model handles nested mention. Remember that We assign scores for each text fragment. e.g. if we use the example of UofT, UofT has its own score and Toronto has its own score.
The user detects nested entity. He can easily control whether to keep nested mention based on the task definition.
Another benefit is that our word level feature and character feature are derived from data. There is no feature engineering at all.

# CoNLL2003 Shared Task

## CoNLL2003 Shared Task

- newswire from the Reuters RCV1 corpus;
- tagged with 4 types of **non-nested** named entities: Person (PER), Organization (ORG), Location (LOC), and Miscellaneous (MISC).

| | Articles | Sentences | Tokens | LOC | MISC | ORG | PER |
|---|---|---|---|---|---|---|---|
| train | 946 | 14,987 | 203,621 | 7,140 | 3,438 | 6,321 | 6,600 |
| dev | 216 | 3,466 | 51,362 | 1,837 | 922 | 1,341 | 1,842 |
| test | 231 | 3,684 | 46,435 | 1,668 | 702 | 1,661 | 1,617 |

Table: Data distribution of CoNLL2003





CoNLL2003 Shared Task
CoNLL2003 Shared Task
newswire from the Reuters RCV1 corpus;
tagged with 4 types of non-nested named entities: Person (PER), Organization (ORG), Location (LOC), and Miscellaneous (MISC).
Table: Data distribution of CoNLL2003

We first evaluate our model on CoNLL2003 shared task.

The task defines 4 non-nested named entities.

# Feature Effectiveness

| FEATURE | | | P | R | F1 |
|---|---|---|---|---|---|
| word level | case-insensitive | context FOFE incl. word fragment | 86.64 | 77.04 | 81.56 |
| | | context FOFE excl. word fragment | 53.98 | 42.17 | 47.35 |
| | | BoW of word fragment | 82.92 | 71.85 | 76.99 |
| | case-sensitive | context FOFE incl. word fragment | 88.88 | 79.83 | 84.12 |
| | | context FOFE excl. word fragment | 50.91 | 42.46 | 46.30 |
| | | BoW of word fragment | 85.41 | 74.95 | 79.84 |
| char level | Char FOFE of word fragment | | 67.67 | 52.78 | 59.31 |
| | Char CNN of word fragment | | 78.93 | 69.49 | 73.91 |
| all case-insensitive features | | | 90.11 | 82.75 | 86.28 |
| all case-sensitive features | | | 90.26 | 86.63 | 88.41 |
| all word-level features | | | 92.03 | 86.08 | 88.96 |
| all word-level & Char FOFE features | | | 91.68 | 88.54 | **90.08** |
| all word-level & Char CNN features | | | 91.80 | 88.58 | **90.16** |
| all word-level & all char-level features | | | 93.29 | 88.27 | **90.71** |

Table: Effect of various FOFE feature combinations on the CoNLL2003 test data.





Let's look at the effectivenes of various features. **(Next)**
From line2, we can see that the context along does much better than random guess. It proves that context is a deciding factor.
From line3, we can see that the text fragment alone is not strong enough. It's ambiguous. **(Next)**
From line11, we can see that when we combined all word level features of text fragment and context, it allows us to make accurate local decision. **(Next)**
Line 12 includes character FOFE. Line 13 includes character. FOFE is competitive to CNN when modeling the internal structure of the text fragment.

# Feature Effectiveness

| FEATURE | | | P | R | F1 |
|---|---|---|---|---|---|
| word level | case-insensitive | context FOFE incl. word fragment | 86.64 | 77.04 | 81.56 |
| | | context FOFE excl. word fragment | 53.98 | 42.17 | 47.35 |
| | | BoW of word fragment | 82.92 | 71.85 | 76.99 |
| | case-sensitive | context FOFE incl. word fragment | 88.88 | 79.83 | 84.12 |
| | | context FOFE excl. word fragment | 50.91 | 42.46 | 46.30 |
| | | BoW of word fragment | 85.41 | 74.95 | 79.84 |
| char level | Char FOFE of word fragment | | 67.67 | 52.78 | 59.31 |
| | Char CNN of word fragment | | 78.93 | 69.49 | 73.91 |
| all case-insensitive features | | | 90.11 | 82.75 | 86.28 |
| all case-sensitive features | | | 90.26 | 86.63 | 88.41 |
| all word-level features | | | 92.03 | 86.08 | 88.96 |
| all word-level & Char FOFE features | | | 91.68 | 88.54 | **90.08** |
| all word-level & Char CNN features | | | 91.80 | 88.58 | **90.16** |
| all word-level & all char-level features | | | 93.29 | 88.27 | **90.71** |

Table: Effect of various FOFE feature combinations on the CoNLL2003 test data.





Feature Effectiveness

| FEATURE | | | P | R | F1 |
|---|---|---|---|---|---|
| word level | case-insensitive | context FOFE incl. word fragment | 86.64 | 77.04 | 81.56 |
| | | context FOFE excl. word fragment | 53.98 | 42.17 | 47.35 |
| | | BoW of word fragment | 82.92 | 71.85 | 76.99 |
| | case-sensitive | context FOFE incl. word fragment | 88.88 | 79.83 | 84.12 |
| | | context FOFE excl. word fragment | 50.91 | 42.46 | 46.30 |
| | | BoW of word fragment | 85.41 | 74.95 | 79.84 |
| char level | Char FOFE of word fragment | | 67.67 | 52.78 | 59.31 |
| | Char CNN of word fragment | | 78.93 | 69.49 | 73.91 |
| all case-insensitive features | | | 90.11 | 82.75 | 86.28 |
| all case-sensitive features | | | 90.26 | 86.63 | 88.41 |
| all word-level features | | | 92.03 | 86.08 | 88.96 |
| all word-level & Char FOFE features | | | 91.68 | 88.54 | 90.08 |
| all word-level & Char CNN features | | | 91.80 | 88.58 | 90.16 |
| all word-level & all char-level features | | | 93.29 | 88.27 | 90.71 |

Table: Effect of various FOFE feature combinations on the CoNLL2003 test data.

Let's look at the effectivenes of various features. **(Next)**
From line2, we can see that the context along does much better than random guess. It proves that context is a deciding factor.
From line3, we can see that the text fragment alone is not strong enough. It's ambiguous. **(Next)**
From line11, we can see that when we combined all word level features of text fragment and context, it allows us to make accurate local decision. **(Next)**
Line 12 includes character FOFE. Line 13 includes character. FOFE is competitive to CNN when modeling the internal structure of the text fragment.

# Feature Effectiveness

| FEATURE | | | P | R | F1 |
|---|---|---|---|---|---|
| word level | case-insensitive | context FOFE incl. word fragment | 86.64 | 77.04 | 81.56 |
| | | context FOFE excl. word fragment | 53.98 | 42.17 | 47.35 |
| | | BoW of word fragment | 82.92 | 71.85 | 76.99 |
| | case-sensitive | context FOFE incl. word fragment | 88.88 | 79.83 | 84.12 |
| | | context FOFE excl. word fragment | 50.91 | 42.46 | 46.30 |
| | | BoW of word fragment | 85.41 | 74.95 | 79.84 |
| char level | Char FOFE of word fragment | | 67.67 | 52.78 | 59.31 |
| | Char CNN of word fragment | | 78.93 | 69.49 | 73.91 |
| all case-insensitive features | | | 90.11 | 82.75 | 86.28 |
| all case-sensitive features | | | 90.26 | 86.63 | 88.41 |
| all word-level features | | | 92.03 | 86.08 | 88.96 |
| all word-level & Char FOFE features | | | 91.68 | 88.54 | **90.08** |
| all word-level & Char CNN features | | | 91.80 | 88.58 | **90.16** |
| all word-level & all char-level features | | | 93.29 | 88.27 | **90.71** |

Table: Effect of various FOFE feature combinations on the CoNLL2003 test data.





Feature Effectiveness
Table: Effect of various FOFE feature combinations on the CoNLL2003 test data.

Let's look at the effectivenes of various features. **(Next)**
From line2, we can see that the context along does much better than random guess. It proves that context is a deciding factor.
From line3, we can see that the text fragment alone is not strong enough. It's ambiguous. **(Next)**
From line11, we can see that when we combined all word level features of text fragment and context, it allows us to make accurate local decision. **(Next)**
Line 12 includes character FOFE. Line 13 includes character. FOFE is competitive to CNN when modeling the internal structure of the text fragment.

# Feature Effectiveness

| FEATURE | | | P | R | F1 |
|---|---|---|---|---|---|
| word level | case-insensitive | context FOFE incl. word fragment | 86.64 | 77.04 | 81.56 |
| | | context FOFE excl. word fragment | 53.98 | 42.17 | 47.35 |
| | | BoW of word fragment | 82.92 | 71.85 | 76.99 |
| | case-sensitive | context FOFE incl. word fragment | 88.88 | 79.83 | 84.12 |
| | | context FOFE excl. word fragment | 50.91 | 42.46 | 46.30 |
| | | BoW of word fragment | 85.41 | 74.95 | 79.84 |
| char level | Char FOFE of word fragment | | 67.67 | 52.78 | 59.31 |
| | Char CNN of word fragment | | 78.93 | 69.49 | 73.91 |
| all case-insensitive features | | | 90.11 | 82.75 | 86.28 |
| all case-sensitive features | | | 90.26 | 86.63 | 88.41 |
| all word-level features | | | 92.03 | 86.08 | 88.96 |
| all word-level & Char FOFE features | | | 91.68 | 88.54 | **90.08** |
| all word-level & Char CNN features | | | 91.80 | 88.58 | **90.16** |
| all word-level & all char-level features | | | 93.29 | 88.27 | **90.71** |

Table: Effect of various FOFE feature combinations on the CoNLL2003 test data.





Let's look at the effectivenes of various features. **(Next)**
From line2, we can see that the context along does much better than random guess. It proves that context is a deciding factor.
From line3, we can see that the text fragment alone is not strong enough. It's ambiguous. **(Next)**
From line11, we can see that when we combined all word level features of text fragment and context, it allows us to make accurate local decision. **(Next)**
Line 12 includes character FOFE. Line 13 includes character. FOFE is competitive to CNN when modeling the internal structure of the text fragment.

# Comparison between Neural Network Models

| algorithm | word | char | gaz | cap | pos | F1 |
|---|---|---|---|---|---|---|
| CNN-BLSTM-CRF (Collobert et al. 2011) | ✓ | ✗ | ✓ | ✓ | ✗ | 89.59 |
| BLSTM-CRF (Huang, Xu, and Yu 2015) | ✓ | ✓ | ✓ | ✓ | ✓ | 90.10 |
| BLSTM-CRF (Rondeau and Su 2016) | ✓ | ✗ | ✓ | ✓ | ✓ | 89.28 |
| BLSTM-CRF, char-CNN (Chiu and Nichols 2016) | ✓ | ✓ | ✓ | ✗ | ✗ | **91.62** |
| Stack-LSTM-CRF, char-LSTM (Lample et al. 2016) | ✓ | ✓ | ✗ | ✗ | ✗ | **90.94** |
| **FOFE-NER** (single) | ✓ | ✓ | ✗ | ✗ | ✗ | **90.71** |
| **FOFE-NER** (ensemble) + dev | ✓ | ✓ | ✗ | ✗ | ✗ | **90.92** |

Table: Performance ($F_1$ score) comparison among various neural models reported on the CoNLL dataset, and the different features used in these methods.





Comparison between Neural Network Models

In this table, we comare our result with other neural network models. Except the model in line 5, all other methods involves heavy feature engineering or make use of external knowledge.

However, the model in line 5 is much more computationally expensive than ours.

# EDL in KBP2015

## EDL Track in KBP2015 (Ji, Nothman, and Hachey 2015)

- Requires to identify entities **(including nested entities)** from English, Chinese and Spanish documents.
- 5 entity types are defined, i.e. Person (PER), Geo-political Entity (GPE), Organization (ORG), Location (LOC) and Facility (FAC).
- Documents are related but **non-parallel** across languages.

| | English | Chinese | Spanish | ALL |
|---|---|---|---|---|
| Train | 168 | 147 | 129 | 444 |
| Eval | 167 | 167 | 166 | 500 |

Table: Number of Documents in KBP2015





EDL in KBP2015
EDL Track in KBP2015 (Ji, Nothman, and Hachey 2015)
Requires to identify entities (including nested entities) from English, Chinese and Spanish documents.
5 entity types are defined, i.e. Person (PER), Geo-political Entity (GPE), Organization (ORG), Location (LOC) and Facility (FAC).
Documents are related but non-parallel across languages.
Table: Number of Documents in KBP2015

We further evaluate our model using the EDL track in KBP2015.
The contest is trilingual.
Documents are related but not parallel.

## EDL in KBP2015

| | 2015 track best | | | FOFE-NER | | |
|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| Trilingual | 75.9 | 69.3 | 72.4 | 78.3 | 69.9 | **73.9** |
| English | 79.2 | 66.7 | **72.4** | 77.1 | 67.8 | 72.2 |
| Chinese | 79.2 | 74.8 | **76.9** | 79.3 | 71.7 | 75.3 |
| Spanish | 78.4 | 72.2 | 75.2 | 79.9 | 71.8 | **75.6** |

Table: Entity Discovery Performance of our method on the KBP2015 EDL evaluation data, with comparison to the best systems in KBP2015 official evaluation.





EDL in KBP2015

| | 2015 track best | | | FOFE-NER | | |
|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| Trilingual | 75.9 | 69.3 | 72.4 | 78.3 | 69.9 | **73.9** |
| English | 79.2 | 66.7 | **72.4** | 77.1 | 67.8 | 72.2 |
| Chinese | 79.2 | 74.8 | **76.9** | 79.3 | 71.7 | 75.3 |
| Spanish | 78.4 | 72.2 | 75.2 | 79.9 | 71.8 | **75.6** |

Table: Entity Discovery Performance of our method on the KBP2015 EDL evaluation data, with comparison to the best systems in KBP2015 official evaluation.

On the left hand side of the table, we have the perfrmance of the track best.
Ours is on the right hand side.
Our model performs similarly to the best in English and CHinese, and surpasses the best in Spanish and the overall performance.

# EDL in KBP2016

## EDL Track in KBP2016 (Ji, Nothman, and Dang 2016)

- The task is extended to detect **nominal mentions** of all 5 entity typess.
- We treat nominal mention types as some extra entity types and detect them along with named entities.

## Example

[*Hinton*]$_{\textit{PER-NAM}}$, a [*professor*]$_{\textit{PER-NOM}}$ of [*University of* [*Toronto*]$_{\textit{GPE-NAM}}$]$_{\textit{ORG-NAM}}$, spends several months in [*Google*]$_{\textit{ORG-NAM}}$'s [*Mountain View*]$_{\textit{LOC-NAM}}$ office every year.





We participated the EDL track of KBP2016.
The task is much harder in the sense that the particpants were asked to annotate nominal mentions.

## Training Data for KBP2016

**Training and evaluation data in KBP2015**

(described before), nominal mentions are not labeled.

**Machine-labeled Wikipedia (WIKI)**

When terms or names are first mentioned in a Wikipedia article they are often linked to the corresponding Wikipedia page by hyperlinks.

**In-house dataset**

A set of 10,000 English and Chinese documents is manually labeled using some annotation rules similar to the KBP 2016 guidelines.





Training Data for KBP2016
Training and evaluation data in KBP2015
(described before), nominal mentions are not labeled.
Machine-labeled Wikipedia (WIKI)
When terms or names are first mentioned in a Wikipedia article they are often linked to the corresponding Wikipedia page by hyperlinks.
In-house dataset
A set of 10,000 English and Chinese documents is manually labeled using some annotation rules similar to the KBP 2016 guidelines.

In 2016, NIST didn't provide any official training data.
We make use of 3 data sources (READ SLIDES)
first, training and evaluation data in 2015,
second, machine-labeled data generated from wikipedia, and
at last, our in-house data.

# Dataset Effectivness

| training data | P | R | $F_1$ |
|---|---|---|---|
| KBP2015 | 0.836 | 0.598 | 0.697 |
| KBP2015 + WIKI | 0.837 | 0.628 | **0.718** |
| KBP2015 + in-house | 0.836 | 0.680 | **0.750** |

Table: Our entity discovery official performance (English only) in KBP2016 is shown as a comparison of three models trained by different combinations of training data sets.

FOFE-NER ranks **2nd** place in **first participation**.

FOFE-NER is the **best single model** among all participants.





Dataset Effectivness

| training data | P | R | $F_1$ |
|---|---|---|---|
| KBP2015 | 0.836 | 0.598 | 0.697 |
| KBP2015 + WIKI | 0.837 | 0.628 | **0.718** |
| KBP2015 + in-house | 0.836 | 0.680 | **0.750** |

Table: Our entity discovery official performance (English only) in KBP2016 is shown as a comparison of three models trained by different combinations of training data sets.

FOFE-NER ranks 2nd place in first participation.

FOFE-NER is the best single model among all participants.

When we compare line1 with line2, we can see the performance gain from machine-generated data.
It proves that our model can be easily improved by machine-generated data.
Even though it's our first participation, we rank the 2nd place.
It is also the best single-model system among all participants.

# Conclusion

A local detection approach to NER and MD by applying FFNN on top of FOFE

Nested mention detection

Much more efficient then known solutions

No feature engineering and No external knowledge

On a par with state-of-the-art ED system





Conclusion
A local detection approach to NER and MD by applying FFNN on top of FOFE
Nested mention detection
Much more efficient then known solutions
No feature engineering and No external knowledge
On a par with state-of-the-art ED system

In conclusion, this paper proposed a local detection approach to NER & MD by applying FFNN on top of FOFE.
it's able to detect nested mention. We reached state-of-the-art performance with less computation resource and without any feature engineering.

# Thank You!
# (Q&A)

code https://github.com/xmb-cipher/fofe-ner

demo http://www.eecs.yorku.ca/~nana/ner-home.html





Thank You!
(Q&A)
code https://github.com/xmb-cipher/fofe-ner
demo http://www.eecs.yorku.ca/~nana/ner-home.html

If you're interested in our work, feel free to try our demo and implmenetation.

Thanks for your attention.

- Chiu, Jason P. C. and Eric Nichols (2016). "Named entity recognition with bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4, pp. 357–370.
- Collobert, Ronan et al. (2011). "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12.Aug, pp. 2493–2537.
- Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF models for sequence tagging". In: *arXiv preprint arXiv:1508.01991.*
- Ji, Heng, Joel Nothman, and Hoa Trang Dang (2016). "Overview of TAC-KBP2016 Tri-lingual EDL and Its Impact on End-to-End Cold-Start". In: *Proceedings of Text Analysis Conference (TAC2016).*







- Chiu, Jason P. C. and Eric Nichols (2016). "Named entity recognition with bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4, pp. 357–370.
- Collobert, Ronan et al. (2011). "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12.Aug, pp. 2493–2537.
- Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF models for sequence tagging". In: *arXiv preprint arXiv:1508.01991.*
- Ji, Heng, Joel Nothman, and Hoa Trang Dang (2016). "Overview of TAC-KBP2016 Tri-lingual EDL and Its Impact on End-to-End Cold-Start". In: *Proceedings of Text Analysis Conference (TAC2016).*

- Ji, Heng, Joel Nothman, and Ben Hachey (2015). “Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking”. In: *Proceedings of Text Analysis Conference (TAC2015)*.
- Kim, Yoon et al. (2016). “Character-aware neural language models”. In: *AAAI*. Citeseer.
- Lample, Guillaume et al. (2016). “Neural architectures for named entity recognition”. In: *arXiv preprint arXiv:1603.01360*.
- Nothman, Joel et al. (2013). “Learning multilingual named entity recognition from Wikipedia”. In: *Artificial Intelligence* 194, pp. 151–175. DOI: 10.1016/j.artint.2012.03.006. URL: https://doi.org/10.1016/j.artint.2012.03.006.
- Rondeau, Marc-Antoine and Yi Su (2016). “LSTM-Based NeuroCRFs for Named Entity Recognition”. In: *Interspeech 2016*. International Speech Communication Association, pp. 665–669. DOI: 10.21437/interspeech.2016-288.




2017-08-02




- Ji, Heng, Joel Nothman, and Ben Hachey (2015). “Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking”. In: *Proceedings of Text Analysis Conference (TAC2015)*.
- Kim, Yoon et al. (2016). “Character-aware neural language models”. In: *AAAI*. Citeseer.
- Lample, Guillaume et al. (2016). “Neural architectures for named entity recognition”. In: *arXiv preprint arXiv:1603.01360*.
- Nothman, Joel et al. (2013). “Learning multilingual named entity recognition from Wikipedia”. In: *Artificial Intelligence* 194, pp. 151–175. DOI: 10.1016/j.artint.2012.03.006. URL: https://doi.org/10.1016/j.artint.2012.03.006.
- Rondeau, Marc-Antoine and Yi Su (2016). “LSTM-Based NeuroCRFs for Named Entity Recognition”. In: *Interspeech 2016*. International Speech Communication Association, pp. 665–669. DOI: 10.21437/interspeech.2016-288.

Zhang, Shiliang et al. (2015). "The Fixed-Size Ordinally-Forgetting Encoding Method for Neural Network Language Models". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics (ACL). DOI: 10.3115/v1/p15-2081.





Zhang, Shiliang et al. (2015). "The Fixed-Size Ordinally-Forgetting Encoding Method for Neural Network Language Models". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics (ACL). DOI: 10.3115/v1/p15-2081.