Code and Named Entity Recognition in StackOverflow

Jeniya Tabassum, Mounica Maddela, Alan Ritter, Wei Xu

The Ohio State University
**StackOverflow Entity Recognition**

<table>
<thead>
<tr>
<th>BRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am trying to do a search query on the SoundCloud API.</td>
</tr>
<tr>
<td>Using their JavaScript SDK, the following works:</td>
</tr>
</tbody>
</table>

**NEW NER CORPUS**

15,372 Sentences
Manually Annotated
20 Entity Types
StackOverflow Entity Recognition

<table>
<thead>
<tr>
<th>BRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am trying to do a search query on the SoundCloud API.</td>
</tr>
<tr>
<td>Using their JavaScript SDK, the following works:</td>
</tr>
</tbody>
</table>

Available at: https://github.com/jeniyat/StackOverflowNER

NEW NER CORPUS

15,372 Sentences
Manually Annotated
20 Entity Types
SoftNER extracts the software entities with 79.1% F₁

<table>
<thead>
<tr>
<th>BRAT</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am trying to do a search query on the SoundCloud API.</td>
<td></td>
</tr>
<tr>
<td>Using their JavaScript SDK, the following works:</td>
<td></td>
</tr>
</tbody>
</table>

NEW NER CORPUS

15,372 Sentences
Manually Annotated
20 Entity Types

Available at: https://github.com/jeniyat/StackOverflowNER
StackOverflow Entity Recognition

**BRAT**

I am trying to do a search query on the SoundCloud API.

Using their JavaScript SDK, the following works:

Available at: https://github.com/jeniyat/StackOverflowNER

**NEW NER CORPUS**

15,372 Sentences
Manually Annotated
20 Entity Types

**SoftNER**

extracts the software entities with 79.1% $F_1$

**NEW NER MODEL**

21.6 $F_1$

**Fine-tuned BERT off-the-self**

extracts the software entities with 57.5% $F_1$
StackOverflow Entity Recognition

**NEW NER CORPUS**
15,372 Sentences
Manually Annotated
20 Entity Types

**BRAT**
I am trying to do a search query on the SoundCloud API.
Using their JavaScript SDK, the following works:

**SoftNER**
extracts the software entities with 79.1% $F_1$

**NEW NER MODEL**
21.6 $F_1$↑
BERTOverflow
Pre-trained on 152M Sentences

**Fine-tuned BERT** off-the-self
extracts the software entities with 57.5% $F_1$

Available at: https://github.com/jeniyat/StackOverflowNER
Using mysql find the sorted list of 200 most popular links with date earlier than 2014/02/25

```
SELECT url FROM links WHERE date =
    ( SELECT date FROM links WHERE date < "2014/02/25" ORDER BY date DESC LIMIT 1)
ORDER BY date DESC, clicks DESC LIMIT 200
```

[Yao et al., 2019]
[Iyer et al. 2016]
[Giorgi and Bader 2018]
Using **mysql** find the sorted list of **200** most popular links with date earlier than **2014/02/25**

**Code Snippet**

```sql
SELECT url FROM links WHERE date =
  ( SELECT date FROM links WHERE date < "2014/02/25" ORDER BY date DESC LIMIT 1)
ORDER BY date DESC, clicks DESC LIMIT 200
```

[Yao et al., 2019]
[Iyer et al. 2016]
[Giorgi and Bader 2018]
**Code Retrieval**

**NL Query:**
Using `mysql` find the sorted list of 200 most popular links with date earlier than `2014/02/25`.

**Code Snippet**
```
SELECT url FROM links WHERE date =
  ( SELECT date FROM links WHERE date < "2014/02/25" ORDER BY date DESC LIMIT 1)
ORDER BY date DESC, clicks DESC LIMIT 200
```

---

**KB Creation**

[Movshovitz-Attias and Cohen 2015]

[Yao et al., 2019]
[Iyer et al. 2016]
[Giorgi and Bader 2018]
NL Query:
Using `mysql` find the sorted list of 200 most popular links with date earlier than 2014/02/25

Code Snippet
```
SELECT url FROM links WHERE date =
  ( SELECT date FROM links WHERE
    date < "2014/02/25" ORDER BY date
    DESC LIMIT 1)
ORDER BY date DESC, clicks DESC LIMIT 200
```

[Yao et al., 2019]
[Iyer et al. 2016]
[Giorgi and Bader 2018]
StackOverflow Entity Recognition

Challenges

Polysemy

InLine Code
StackOverflow Entity Recognition

Polysemy

Challenges

InLine Code
Polysemy

list
Polysemy

Data Structure

`itertools` can be used to flatten a nested Python list.

list
Polysemy

Data Structure
itertools can be used to flatten a nested python list

Class
The List class is the child interface of java Collection.
Polysemy

The **List** class is the child interface of java Collection.

**Data Structure**

Itertools can be used to flatten a nested python list

**Variable**

I want to iterate through the following list

```java
ArrayList<String> list = new ArrayList<String>();
list.add("http://www.google.com");
list.add("http://www.stackoverflow.com");
exchange.getOut().setHeader("endpoints", list);
```
Polysemy

Data Structure

Itertools can be used to flatten a nested python list

Class

The List class is the child interface of java Collection.

Non Entity

Here is a list of most useful ajax frameworks

Variable

I want to iterate through the following list

```java
ArrayList<String> list = new ArrayList<String>();
list.add("http://www.google.com");
list.add("http://www.stackoverflow.com");
exchange.getOut().setHeader("endpoints", list);
```
StackOverflow Entity Recognition

Polysemy

Challenges

InLine Code
check if key is numeric by is_numeric($key) function
check if key is numeric by is_numeric($key) function
check if key is numeric by `is_numeric($key)` function
Before adding element to array, check if key is numeric by `is_numeric($key)` function. If it return false, then covert key to integer using typecasting, `(int)$key.

Now, the array will have numeric keys only and can be ordered.
Before adding element to array, check if key is numeric by `is_numeric($key)` function. If it return false, then, covert key to integer using typecasting, `(int)$key.

Now, the array will have numeric keys only and can be ordered.
Contribution #1: Construct new data for Software Entities

Contribution #2: Propose a new model for Software Entities
Contribution #1: Construct **new data** for Software Entities

- Manually annotated sentence with **20 types of entities**
- **15k** StackOverflow sentence

Contribution #2: Propose a **new model** for Software Entities
Annotated StackOverflow Corpus

1237 Question-Answer Threads

- Most upvoted answer
- Accepted answer
- 2 randomly selected answers

15,372 Sentences

Archive timeline: (2008 - 2018)
Annotated StackOverflow Corpus

Archive timeline: (2008 - 2018)

1237 Question-Answer Threads

- Most upvoted answer
- Accepted answer
- 2 randomly selected answer

15,372 Sentences

20 Entity Types

Code Entity Types
- Class
- Value
- Function
- HTML/XML_Tag
- In_Line_code
- Variable
- Data_Type
- Library

Natural Language Entity
- Algorithm
- Application
- Data_Structure
- File_Type
- Version
- Language
- Website
- File_Name
- Operating_System
- User_Name
Annotated StackOverflow Corpus

1237 Question-Answer Threads

- Most upvoted answer 🔺
- Accepted answer ✔
- 2 randomly selected answers

15,372 Sentences
20 Entity Types

Code Entity Types
- Class
- Value
- Function
- HTML_XML_Tag
- In_Line_code
- Variable
- Data_Type
- Library

Natural Language Entity Types
- Algorithm
- Application
- Data_Structure
- File_Type
- Version
- Language
- Website
- File_Name
- Operating_System
- User_Name
Annotated StackOverflow Corpus

1237 Question-Answer Threads

- Most upvoted answer
- Accepted answer
- 2 randomly selected answers

15,372 Sentences
20 Entity Types

[40% double annotated]
[62% agreement]

[annotators]
Annotated StackOverflow Corpus

1237 Question-Answer Threads

- Most upvoted answer
- Accepted answer
- 2 randomly selected answer

15,372 Sentences
20 Entity Types

[40% double annotated]
[62% agreement]

- Resolves disagreement in double annotated data
- Ensures the sanity of single annotated data
Contribution #1: Construct new data for Software Entities
  • Manually annotated sentence with 20 types of entities
  • 15k StackOverflow sentence

Contribution #2: Proposed a new model for Software Entities
  • Attentive NER tagger
  • combined with in-domain contextual representation
SoftNER

Contextual Word Representation

Entity Segmenter

Code Recognizer

Polysemy

Challenges

InLine Code
cpp uses quick sort QSort()
cpp QSort() uses quick sort
BERTOverflow

152M sentences
StackOverflow 10 year archive
Word Context

BERTOverflow

152M sentences
StackOverflow 10 year archive

Excludes sentences from the annotated corpus
cpp QSort() uses quick sort
SoftNER

Code Recognizer

GigaWord Corpus

PP(G_char)

PP(G_word)

StackOverflow Code Snippet Corpus

PP(SC_word)

PP(SC_char)

Gaussian vectorization

Input Text

cpp  QSort()  uses  quick  sort
Code Recognizer

0/1 Binary Output for the input word

Feed Forward Layer - Sigmoid

Feed Forward Layer

GigaWord Corpus

FastText

Gaussian vectorization

StackOverflow Code Snippet Corpus

Input Text

cpp
QSort()
uses
quick
sort
Code Recognizer

0/1 Binary Output for the input word

Feed Forward Layer - Sigmoid

Feed Forward Layer

QSort()

GigaWord Corpus

StackOverflow Code Snippet Corpus

FastText

Gaussian vectorization
QSort()
Code Recognizer

0/1 Binary Output for the input word

Feed Forward Layer - Sigmoid

Feed Forward Layer

FastText

GigaWord Corpus

StackOverflow Code Snippet Corpus

sort

Gaussian vectorization
Input Text

cpp
QSort()
uses
quick
sort
cpp QSort() uses quick sort
cpp
QSort()
uses
quick
sort
cpp uses quick sort

Entity Segmenter

Feed Forward Layer

cpp QSort uses quick sort

Word Context

Word Markdown

Word Frequency

Entity Segmenter

Code Recognizer

Input Text
cpp uses QSort() uses quick sort
cpp uses QSort() uses quick sort
cpp uses QSort() uses quick sort
cpp

QSort()

uses

quick

sort
Embedding Level Attention

input embeddings

\[ e_{it} \]
Embedding Level Attention

Input embeddings $e_{it}$ → BiGRU $h_{it}$
Embedding Level Attention

\[ e_{it} \xrightarrow{\text{BiGRU}} h_{it} \xrightarrow{\text{tanh}} u_{it} \xrightarrow{\text{softmax}} \alpha_{it} \]

- **Input Embeddings**
- **BiGRU**
- **tanh**
- **softmax**
- **Normalized Embedding Weights**

**Weights and Biases**
- \( W_e \)
- \( b_e \)
- **Embedding Context**

**Textual Input**
- Input Text
- `cpp`
- `QSort()`
- `uses`
- `quick`
- `sort`
SoftNER

Embedding Level Attention

Attentive Embedding

Word Context

Code Recognizer

Entity Segmenter

Input Text

cpp QSort() uses quick sort
SoftNER

Embedding Level Attention

Attentive Embedding

Linear CRF

Embedding Level Attention

Word Context

Code Recognizer

Entity Segmenter

Input Text

cpp  QSort()  uses  quick  sort
cpp QSort() uses quick sort
SoftNER

Annotated Corpus

stack overflow

1237 Question-Answer Threads

- Most upvoted answer
- Accepted answer
- 2 randomly selected answer

15,372 Sentences

<table>
<thead>
<tr>
<th>Single annotated</th>
<th>Double annotated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train Set</td>
<td>Dev Set</td>
</tr>
<tr>
<td>741 Question-Answers</td>
<td>247 Question-Answers</td>
</tr>
<tr>
<td>9,315 Sentences</td>
<td>2,942 Sentences</td>
</tr>
</tbody>
</table>
Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Based CRF</td>
<td>71.77</td>
<td>73.03</td>
<td>78.42</td>
</tr>
<tr>
<td>BiLSTM CRF (ELMoVerflow)</td>
<td>68.77</td>
<td>67.47</td>
<td>68.12</td>
</tr>
<tr>
<td>Fine tuned BERT</td>
<td>77.02</td>
<td>64.82</td>
<td>68.68</td>
</tr>
<tr>
<td>Fine tuned BERTOverflow</td>
<td>51.12</td>
<td>57.54</td>
<td>68.12</td>
</tr>
</tbody>
</table>

StackOverflow
Precision

Recall

F1

Feature Based CRF

BiLSTM CRF (ELMoVerflow)

Fine tuned BERT

Fine tuned BERTOverflow

Performance

10.58 F1 over BERT\textsubscript{off-the-self}
Performance

<table>
<thead>
<tr>
<th>Feature Based CRF</th>
<th>Fine tuned BERT</th>
<th>BiLSTM CRF (ELMoOverflow)</th>
<th>Fine tuned BERTOverflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>71.77</td>
<td>77.02</td>
<td>78.42</td>
</tr>
<tr>
<td>Recall</td>
<td>68.77</td>
<td>64.82</td>
<td>67.47</td>
</tr>
<tr>
<td>F1</td>
<td>57.54</td>
<td>51.12</td>
<td>57.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>68.68</td>
<td>68.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.1</td>
<td></td>
</tr>
</tbody>
</table>
Performance

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Based CRF</td>
<td>71.77</td>
<td>73.03</td>
<td>73.03</td>
</tr>
<tr>
<td>Fine tuned BERT</td>
<td>77.02</td>
<td>68.77</td>
<td>71.77</td>
</tr>
<tr>
<td>BiLSTM CRF (ELMoVerflow)</td>
<td>68.82</td>
<td>67.47</td>
<td>68.68</td>
</tr>
<tr>
<td>SoftNER (BERTOverflow)</td>
<td>51.12</td>
<td>57.54</td>
<td>57.54</td>
</tr>
</tbody>
</table>

StackOverflow
Performance

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Based CRF</td>
<td>71.77</td>
<td>68.77</td>
<td>74.34</td>
</tr>
<tr>
<td>BiLSTM CRF (ELMoVerflow)</td>
<td>78.02</td>
<td>64.82</td>
<td>77.68</td>
</tr>
<tr>
<td>Fine tuned BERT</td>
<td>77.02</td>
<td>51.12</td>
<td>63.87</td>
</tr>
<tr>
<td>Fine tuned BERTOverflow</td>
<td>78.42</td>
<td>68.68</td>
<td>73.14</td>
</tr>
<tr>
<td>SoftNER (BERTOverflow)</td>
<td>79.79</td>
<td>57.54</td>
<td>69.36</td>
</tr>
</tbody>
</table>

10.98 F1 over fine-tuned BERTOverflow
Does SoftNER work on other social software domain text?
Performance on GitHub Test Set

GitHub Archive timeline: (2007 - 2017)

- 143 Repositories
- 143 Readme
- 9 issue events
- 3 issue comments

6,510 Sentences
Performance on GitHub Test Set

GitHub
Archive timeline:

143 Readme
9 issue events
3 issue comments

6,510 Sentences

143 Repositories

Precision
Recall
F1

- Feature Based CRF
- Fine tuned BERTOverflow
- SoftNER (BERTOverflow)

Trained on 9,315 manually annotated StackOverflow Sentences
Performance on GitHub Test Set

143 Repositories
143 Readme
9 issue events
3 issue comments
6,510 Sentences

Feature Based CRF
Fine tuned BERTOverflow
BiLSTM-CRF (ELMoHUB)
SoftNER (BERTOverflow)

43.16 61.71 61.92 64.53
58.75 60.26 60.96
39.09 60.19 61.08 62.69

Precision  Recall  F1

Trained on 9,315 manually annotated StackOverflow Sentences

GitHub
Takeaway

Task: Fine-Grained Software Entity Extraction
- Covers **20 types** of fine-grained named entities
- Works on StackOverflow and Github Data

code and data at: [https://github.com/jeniyat/StackOverflowNER](https://github.com/jeniyat/StackOverflowNER)
Takeaway

Task: Fine-Grained Software Entity Extraction
• Covers 20 types of fine-grained named entities
• Works on StackOverflow and Github Data

New Data
• 15k manually annotated sentences with fine grained software entities
• Software domain word vectors trained on 152M sentences.
  ❖ BERTOverflow, ELMoVerflow, GLoVerflow

code and data at: https://github.com/jeniyat/StackOverflowNER
## Takeaway

### Task: Fine-Grained Software Entity Extraction
- Covers **20 types** of fine-grained named entities
- Works on StackOverflow and Github Data

### New Data
- **15k** manually annotated sentences with fine-grained software entities
- Software domain word vectors trained on **152M** sentences.
  - BERTOverflow, ELMoVerflow, GLoVerflow

### New Model
- Attentive NER tagger to combine the domain knowledge with contextual knowledge
  - **10.98 F1 increase** over fine tuned BERT
- Standalone code token recognizer

---

code and data at: [https://github.com/jeniyat/StackOverflowNER](https://github.com/jeniyat/StackOverflowNER)
Takeaway

Task: Fine-Grained Software Entity Extraction
• Covers **20 types** of fine-grained named entities
• Works on StackOverflow and Github Data

New Data
• **15k** manually annotated sentences with fine-grained software entities
• Software domain word vectors trained on **152M** sentences.
  ❖ BERTOverflow, ELMoVerflow, GLoVerflow

New Model
• Attentive NER tagger to combine the domain knowledge with contextual knowledge
  ❖ **10.98 F1 increase** over fine tuned BERT
• Standalone code token recognizer

code and data at: https://github.com/jeniyat/StackOverflowNER