Chasing the Tail of Self Supervised Named Entity Disambiguation

Knowledge Graphs Seminar
April 2021

Laurel Orr,
Megan Leszczynski,
Neel Guha,
Sen Wu,
Simran Arora,
Xiao Ling (Apple),
Chris Ré
Named Entity Disambiguation

Map “strings to things” in a knowledge base

How tall is Lincoln?

Key part of assistant, search, and information extraction
The Long Tail of NED

HEAD
- Washington, DC
  - Q61

TORSO
- Chevrolet, Corvette
  - Q56166

TAIL
- Reddish Potato Beetle
  - Q14934552

UNSEEN
- Sauce! by XXXTENTACION
  - Q???

IR and BERT/GPT work great for the HEAD.

The majority of entities are rare!
13% entities have Wikipedia page.
< 1% of songs in Wikidata!
Tail Scalability Challenge

Large number of patterns needed to resolve the tail, making it difficult to scale a system that can learn the patterns.

But the majority of entities are rare (tail).

90 million entities in Wikidata → 90*100 million examples for 60 F1

Subtle reasoning clues are needed for the tail! (+40 F1 points by encoding these reasoning patterns)
How do you disambiguate the tail?

(Ans: structural information over types and relations)
Disambiguation Input & Output

**Input:** Sentence

*Where is Lincoln in Logan County?*

**Output:** Entities

- **Lincoln, IL**
- **Logan County, IL**

**Extract Candidates**

- Lincoln, IL
- Lincoln, NE
- Abraham Lincoln
- Logan County, IL
- Logan County, OK
- Logan County, OH

**Entity Profiles**

- **Entity Profiles**

**Entity Payload**

- **Entity Payload**

**Disambiguate**

- **Disambiguate**

**Bootleg**

- **Bootleg**

*Entity Payload*

```json
{
  "id": "Q292973",
  "name": "Logan County, IL",
  "types": ["county", "geographic_loc"],
  "relations": [
    [<"capital-of", "Q457134">,
    <"named-after">, "Q169067"]
}
```
Signals of NED

Want to learn **information at level** of

- Individual entities
- Relationships
- Types
Text Memorization

Where is Lincoln in Nebraska

What year did Steven Speilberg make Lincoln?

Associate phrases per entity – but this doesn’t generalize to other entities
Knowledge Graph Relations

Associate phrases per relation and use relations to disambiguate – generalizes to rare entities with the same relation.
Types Patterns

How tall is Lincoln?

What is the cheapest Lincoln?

How many people are in Lincoln?

People have heights, not places or brands

Brands have prices, not places or people

Places have populations, not people or brands

Associate phrases per entity type – generalizes to rare entities of the same type
Signals of NED

Want to learn **information at level** of

- Individual entities
- Relationships
- Types

In Wikipedia sample

- 22% of entities have a relationship
  - 126 relationships occur more than 100 times
- 96% of entities have types
  - 2.8K occur more than 100 times

**Takeaway:** learning patterns of types and relations will help generalize to the tail
How do you learn the entity, type, and KG signals?

(Ans: using embeddings)
Where is Lincoln in Logan County?
Disambiguation Input & Output

Entity Profiles

For each candidate, we use the entity profile to extract (learned) embeddings.
For each candidate, we use the entity profile to extract (learned) embeddings.
Disambiguation Input & Output

Entity Payload

Entity Profiles

KG metadata

relation

combined

Relation Embedding

Entity Profiles

Entity Payload

Logan County, IL

Relation Embedding

Entity Profiles

Entity Payload

Logan County, IL

{  
  id: "Q292973", name: "Logan County, IL"
  types: ["county", "geographic_loc"],
  relations: [{"capital-of": "Q457134"},
    {"named-after": "Q169067"}]
}
Disambiguation Input & Output

Entity Payload

Entity Profiles

Fine Type Embedding

```json
{
  "id": "Q292973",
  "name": "Logan County, IL",
  "types": ["county", "geographic_loc"],
  "relations": [<"capital-of", "Q457134">, <"named-after", "Q169067"]
}
```

Logan County, IL
The entity payload has embeddings mapping to each level of the hierarchy.
Bootleg Architecture

Use simple Transformer building blocks

Architecture supports reasoning over each pattern

KG module allows for related entities to transfer representation

“Where is Lincoln in Logan County?”

.entity embs relation embs type embs
Disambiguation Input & Output

**Input**: Sentence

Where is **Lincoln** in **Logan County**?

**Output**: Entities

- Lincoln, IL
- Logan County, IL

---

**Extract Candidates**

- Lincoln, IL
- Lincoln, NE
- Abraham Lincoln
- Logan County, IL
- Logan County, OK
- Logan County, OH

**Entity Profiles**

- **Logan County, IL**
  - `id`: “Q292973”
  - `name`: “Logan County, IL”
  - `types`: [“county”, “geographic_loc”]
  - `relations`: [<“capital-of”, ”Q457134”>, <“named-after”>, “Q169067”]

**Entity Payload**

- **BoTLEG**

---

**Disambiguate**
Training: how do you encourage learning structural patterns?

(Ans: regularization and training set refinement)
Inverse Hierarchical Regularization

Want an inductive bias to push toward the tail and more general signals.

But we do not want to lose performance over popular entities.
Inverse Hierarchical Regularization

Want an inductive bias to push toward the tail and more general signals.

But we do not want to lose performance over popular entities.

Regularize the **entire** entity embedding proportional to inverse popularity

*Regularization scheme gains 13.6 F1 points on unseen entities*
“Hands-Free” Training

Need self-supervised training data that does not have hand tuned features.

Although they toured together, Lincoln did not take its name from They Might Be Giants' album Lincoln.

Using Wikipedia and Wikidata is easy to maintain with limited engineer effort and is vastly easier to extend to new languages.
Use Weak Supervision to Refine Labels

Wikipedia is known to be sparsely labeled and missing mentions. However, Wikipedia has internal structure: most sentences on an entity’s Wikipedia page are referring to that entity.

Although they toured together, Lincoln did not take its name from They Might Be Giants' album *Lincoln*.
Use Weak Supervision to Refine Labels

Wikipedia is known to be sparsely labeled and missing mentions. However, Wikipedia has internal structure: most sentences on an entity’s Wikipedia page are referring to that entity.

Although they toured together, Lincoln did not take its name from They Might Be Giants' album, *Lincoln*.

Add missing labels by heuristic label functions

Simple heuristics add ~1.7x training data and give 2 F1 point improvement over unseen entities.
Experiments

Bootleg outperforms existing SotA systems on NED benchmarks.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>SotA system</th>
<th>SotA F1</th>
<th>Bootleg</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSS500</td>
<td>Phan et al., 2019</td>
<td>82.3</td>
<td>82.5</td>
</tr>
<tr>
<td>KORE50</td>
<td>Hu et al., 2019</td>
<td>79.9</td>
<td>85.7</td>
</tr>
<tr>
<td>AIDA CoNLL YAGO</td>
<td>Fevry et al., 2020</td>
<td>96.7*</td>
<td>96.7*</td>
</tr>
</tbody>
</table>

*SotA model reports test accuracy (v. F1), so for comparison we evaluate Bootleg on test accuracy for AIDA CoNLL YAGO.
But: Tail Performance is the game!

Wikipedia Dataset.

On the head, BERT-based baseline performs \( \sim 5 \) F1 points of Bootleg. On the tail, Bootleg outperforms baseline by > 40 F1 points!

<table>
<thead>
<tr>
<th>Evaluation Set</th>
<th>BERT NED Baseline</th>
<th>Bootleg</th>
<th># Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>85.9</td>
<td>91.3</td>
<td>4,066K</td>
</tr>
<tr>
<td>Torso Entities</td>
<td>79.3</td>
<td>87.3</td>
<td>1,912K</td>
</tr>
<tr>
<td>Tail Entities</td>
<td>27.8</td>
<td>69.0</td>
<td>163K</td>
</tr>
<tr>
<td>Unseen Entities</td>
<td>18.5</td>
<td>68.5</td>
<td>10K</td>
</tr>
</tbody>
</table>

Overall F1 can be really misleading!
Understanding the Performance Lift

Mined for subpopulations of examples where a reasoning pattern is present for the correct entities.

*On tail, Bootleg demonstrates ability to capture the patterns.*

<table>
<thead>
<tr>
<th>Subpopulation where Signal is Present</th>
<th>BERT NED Baseline (Head/Tail)</th>
<th>Bootleg (Head/Tail)</th>
<th>Coverage (Head/Tail)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>59 / 29</td>
<td>66 / 47</td>
<td>0.7% / 3.3%</td>
</tr>
<tr>
<td>Type</td>
<td>87 / 28</td>
<td>93 / 73</td>
<td>84% / 76%</td>
</tr>
<tr>
<td>KG</td>
<td>91 / 30</td>
<td>98 / 92</td>
<td>27% / 23%</td>
</tr>
</tbody>
</table>

The reasoning patterns are how to improve tail performance.
Production Task

Included Bootleg embeddings into an Overton production task answering millions of users’ factoid queries. We report relative lift.

<table>
<thead>
<tr>
<th>Evaluation Set</th>
<th>English</th>
<th>Spanish</th>
<th>French</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Entities</td>
<td>1.08</td>
<td>1.03</td>
<td>1.02</td>
<td>1.00</td>
</tr>
<tr>
<td>Tail Entities</td>
<td>1.08</td>
<td>1.17</td>
<td>1.05</td>
<td>1.03</td>
</tr>
</tbody>
</table>

See 8-17% lift in production task, even over multiple languages.
Using Bootleg Downstream: SoTA on the TACRED Benchmark

**Goal:** extract the relationship between a subject and object pair.

- Gold relation: per:employee_of

  (subject) Mays worked with several other companies aside from Media Enterprises in his career.

- (object)

**Bootleg resolves errors made in the prior SoTA model.**

- Gold relation: per:cause_of_death

  Vincent Astor, like Marshall, died unexpectedly of a heart attack in 1959...

- Wikidata relation: ['cause of death']

  - SpanBERT no_relation
  - Bootleg per:cause_of_death

**Micro-Avg. F1 on TACRED Revised test dataset:**

<table>
<thead>
<tr>
<th>Model</th>
<th>Test F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpanBERT</td>
<td>78.0</td>
</tr>
<tr>
<td>KnowBERT</td>
<td>79.3</td>
</tr>
<tr>
<td>Bootleg+SpanBERT</td>
<td>80.2 (SoTA)</td>
</tr>
</tbody>
</table>

Zhang et al., 2017 and Hennig et al., 2020.

- Gold relation: org:alternate_names

  The International Water Management Institute or IWMI study said both...

- Wikidata same entity

  - SpanBERT no_relation
  - Bootleg org:alternate_names

**Leveraging type and relation information downstream**

**Understand that sub-strings relate to the same entity**
Bootleg:
  • Clean slate, simple, open source NED

Key ideas:
  • Leverages a simple hierarchy to learn patterns that generalize to the tail
  • Patch errors in the self-supervised model

https://hazyresearch.stanford.edu/bootleg/
lorr1@cs.stanford.edu