Combining Word and Entity Embeddings for Entity Linking

Jose G. MORENO Romaric Besançon Romain Beaumont Eva D’hondt
Anne-Laure Ligozat Sophie Rosset Xavier Tannier Brigitte Grau

Institut de Recherche en Informatique de Toulouse (IRIT)
CEA, LIST, Vision and Content Engineering Lab
Laboratoire d’informatique pour la mécanique et les sciences de l’ingénieur (LIMSI)
jose.moreno@irit.fr

31th May 2017
Entity Linking

- Entity Linking consists of connecting an entity mention that has been identified in a text to one of the known entities in a knowledge base.

“As soon as he landed at the Bangkok airport, Koirala saw Modi’s tweets on the quake, Nepal’s Minister for Foreign Affairs Mahendra Bahadur Pandey said on Tuesday.”
...or Entity Disambiguation

- A simple keyword search will return several related entities:
  - 1763 for “Bangkok”, 131 for “Koirala”, etc.
- Disambiguating between candidates is a challenging task!
Outline

• Introduction
• From word to entity embeddings
• EAT model
• Integrating EAT features into an E.L. system
• Results and Conclusion
Preliminaries

• Entity linking task
  – Annotated corpus such as Wikipedia
  – Knowledge base
  – Annotated documents

• A document is a Web page

• Query
  – Mention (offsets) + document
What we can use?

• Information from the input (query):

  “As soon as he landed at the Bangkok airport, Koirala saw Modi’s tweets on the quake, Nepal’s Minister for Foreign Affairs Mahendra Bahadur Pandey said on Tuesday.”

• Information from the target (knowledge base):

  These parallel sources are frequently used as independent similarities

• Our hypothesis is that the combination of both sources of information in a unique space contributes to disambiguate entities

\[ \varphi("quake", \text{ )} = ? \]
From word to entity embeddings

- You shall know a word by the company it keeps.

...at the Bangkok airport, Koirala saw Modi's tweets...

\[ p(w_o | w_i) = \frac{\exp(v_{w_o}^T v_{w_i})}{\sum_{w=1}^{W} \exp(v_{w}^T v_{w_i})} \]

- You shall know an ENTITY by the company it keeps in a ...

\[ \phi("quake", \text{flags}) = ? \]

\[ \phi(\text{flags}, \text{flags}) = \cdot \]
Our Extended Anchor Text model

- EAT uses an annotated corpus of mentions and their associated entities, such as Wikipedia
- EAT is based on the widely used ‘word embeddings’ techniques to jointly represent words and entities

Devghat is a religious and cultural center in Nepal. The town is located at the junction of the Seti Gandaki and Krishna Gandaki rivers, and is one of the holiest places in Hindu mythology.
Devghat is a religious and cultural center in Nepal. The town...
Main advantages

- Skip-gram or CBOW configuration are allowed into our model

- Anchor text information is taken into account to jointly represent words and entities

- Entity vectors are learned using their context, instead of indirect relations (concatenation or alignment)
Common Entity Linking architecture

- Query expansion and Candidate Generation
  - Comparison between query string and entity mentions (KB, alias, etc.)
  - Levenshtein distance <=2
  - Query is used in a full text index

- Candidate Ranking
  - Baseline features: textual, generation and popularity
  - EAT features

Ji et al
TAC'14
Baseline features

• Generation
  – Equality between the lexical forms:
    • entity mention and an entity label in the KB
    • entity mention and a variation (alias or translation) of an entity label in the KB
  – Inclusion of the entity mention in the name or one of the forms of the variations of an entity in the KB
  – Levenshtein distance (<=2) between an entity mention and an entity form or any of its variations

• Global context (document level)
  – Cosine similarity between the Wikipedia page and the global context
  – Cosine similarity between the Wikipedia page of related entities and the global context

• Popularity
  – The number of incoming links to a Wikipedia page
  – The number of visits of the Wikipedia page (in 2014)
EAT features

- We propose to use four EAT features:

  \[ EAT_1(e, p(q)) = \frac{\sum_{w_i \in p(q)} \cos(e, w_i)}{||p(q)||} \]

  Average local context similarity

  \[ EAT_2(e, p(q)) = \cos(e, \frac{\sum_{w_i \in p(q)} w_i}{||p(q)||}) \]

  Similarity with average local context

  \[ EAT_3(e, p(q)) = \frac{\sum_{i=1...k} \text{argmax}_{w_i \in p(q)} \cos(e, w_i)}{k} \]

  most-k similar words similarity

  \[ EAT_4(e, w_m) = \cos(e, w_m) \]

  Mention similarity
"As soon as he landed at the Bangkok airport, Koirala saw Modi’s tweets on the quake, Nepal’s Minister for Foreign Affairs Mahendra Bahadur Pandey said on Tuesday."
Candidate Ranking (Training)

- For each mention
  - Candidates are generated
  - Correct candidate is marked as the positive example
  - 10 candidates are randomly selected and marked as negative examples

- A model is learned using a supervised algorithm
Candidate Ranking (Prediction)

- For each query
  - Candidates are generated
  - Each individual candidate is classified using the learned model
  - The candidate with highest prediction score for the positive class is returned as the linked entity
- If all candidates are predicted as negative then the mention is considered as out of the Knowledge Base (NIL)
Evaluation setup

- TAC EDL 2015 dataset
  - Train:
    - 168 documents
    - 12175 queries (mentions)
    - 73.6% in KB vs 26.4% NIL
  - Test:
    - 167 documents
    - 13175 queries (mentions)
    - 74.4% in KB vs 25.6% NIL

- Evaluation metrics (\(t\) for the systems and \(r\) for the gold standard)

\[
    P(nil) = \frac{N(e_t = NIL \land e_r = NIL)}{N(e_t = NIL)}
\]
\[
    R(nil) = \frac{N(e_t = NIL \land e_r = NIL)}{N(e_r = NIL)}
\]
\[
    P(link) = \frac{N(e_t = e_r \land e_t \neq NIL)}{N(e_t \neq NIL)}
\]
\[
    R(link) = \frac{N(e_t = e_r \land e_t \neq NIL)}{N(e_r \neq NIL)}
\]
\[
    P(all) = \frac{N(e_t = e_r)}{N(e_t)}
\]
Entity linking results

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>$+EAT_1$</th>
<th>$+EAT_2$</th>
<th>$+EAT_3$</th>
<th>$+EAT_4$</th>
<th>$+EAT_{1/2/3/4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(nil)</td>
<td>0.598</td>
<td>0.604</td>
<td>0.608</td>
<td>0.605</td>
<td>0.605</td>
<td>0.606</td>
</tr>
<tr>
<td>R(nil)</td>
<td>0.815</td>
<td>0.830</td>
<td>0.825</td>
<td>0.828</td>
<td>0.830</td>
<td>0.838</td>
</tr>
<tr>
<td>F(nil)</td>
<td>0.690</td>
<td>0.699</td>
<td>0.700</td>
<td>0.700</td>
<td>0.700</td>
<td>0.704</td>
</tr>
<tr>
<td>P(link)</td>
<td>0.796</td>
<td>0.806</td>
<td>0.800</td>
<td>0.804</td>
<td>0.806</td>
<td>0.814</td>
</tr>
<tr>
<td>R(link)</td>
<td>0.699</td>
<td>0.706</td>
<td>0.706</td>
<td>0.706</td>
<td>0.707</td>
<td>0.710</td>
</tr>
<tr>
<td>F(link)</td>
<td>0.745</td>
<td>0.752</td>
<td>0.750</td>
<td>0.752</td>
<td>0.753</td>
<td>0.759</td>
</tr>
<tr>
<td>P(all)</td>
<td>0.728</td>
<td>0.737</td>
<td>0.735</td>
<td>0.737</td>
<td>0.737</td>
<td>0.742</td>
</tr>
</tbody>
</table>

- Each single EAT feature improves the used baseline in every metric.
- The combination of all EAT features deals with the best result and outperforms the state-of-the-art.
Entity linking results

<table>
<thead>
<tr>
<th>Classifier</th>
<th>P(all)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaboost</td>
<td>0.742</td>
</tr>
<tr>
<td>Top-1 TAC EDL 15</td>
<td>0.737</td>
</tr>
<tr>
<td>Linear SVM</td>
<td>0.689</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.672</td>
</tr>
<tr>
<td>Top-2 TAC EDL 15</td>
<td>0.671</td>
</tr>
</tbody>
</table>

- We have tested three classification algorithms into our EL system: Adaboost, Linear Support vector machines and Random Forest
- Best result is obtained with Adaboost, but other classifiers rank between top 1 and top 2 participants in the challenge
Conclusion

• The EAT model is capable of jointly represent words and entities into a unique space:
  - without loss in accuracy
  - using only their context making it easy to implement and to integrate into new embedding developments

• Four different features have been proposed to encode similarities between the context (or mention) and the candidates using EAT

• Results with the TAC EDL 2015 dataset show:
  - that individual EAT features as well as their combination improves baseline features
  - our final result (P(all) = 0.742) hypothetically achieves the first position in the campaign
Thank you for your attention!

Learning the EAT model

Devghat is a religious and cultural center in Nepal. The town is located at the junction of the Seti Gandaki and Krishna Gandaki rivers, and is one of the holiest places in Hindu mythology.

Training a classifier for EL

Similarities with the EAT model

Linking with the prediction score
Embeddings evaluation

- We trained the EAT model using Wikipedia and evaluated with the analogy task.

- Word representations were evaluated using the analogy dataset and entities using a ‘translated’ version.

- The original dataset is composed of syntactic and semantic relations.

- Only four of five semantic relations (subgroups) were used: capital, common countries, capital world, currency and city in state.
Results of the analogy task

- Adapted versions of two popular implementations for word embeddings were evaluated: Hyperwords and Gensim

<table>
<thead>
<tr>
<th>Subgroup</th>
<th>EAT-hyperwords</th>
<th>EAT-Gensim</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>words</td>
<td>entities</td>
</tr>
<tr>
<td>capital-com-countries</td>
<td>95.7%</td>
<td>63.0%</td>
</tr>
<tr>
<td>capital-world</td>
<td>77.0%</td>
<td>37.3%</td>
</tr>
<tr>
<td>currency</td>
<td>8.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>city-in-state</td>
<td>72.3%</td>
<td>25.8%</td>
</tr>
</tbody>
</table>

- For words, the best results are obtained with EAT-hyperwords which achieves comparable performances to the state-of-the-art
- However, EAT-Gensim represents better the entities in the embedded space
- In both cases, the entity embeddings manage to capture semantic regularities
Statistics on candidate generation