Semantic Text Mining

Maximilian Idahl
maximilian@idahl.de
Outline

• Some Background Knowledge
  Knowledge Graphs and Embeddings

• Paper 1
  Knowledge Graph and Text Jointly Embedding

• Paper 2
  Knowledge Base Unification via Sense Embeddings and Disambiguation
Knowledge Graphs (Knowledge Bases)

- Structured and formal representation of knowledge

- Entities (nodes) and relations (edges)
  \[ G = (E, R) \]

- Often represented as a collection of triples
  e. g. \( (\text{Obama}, \text{was born in}, \text{Hawaii}) \)

- Issue: coverage
Word Embeddings (Skip-gram, Mikolov et al.)

- Word → k-dimensional vector
- Learned from unlabeled text by predicting the context of words
- Vectors connecting pairs of certain relation are almost parallel

See also: tensorflow.org/tutorials/representation/word2vec
Knowledge Graph Embeddings

- Similar to word embeddings
- Entity $\rightarrow$ k-dimensional vector
- Learned from entities and relations in a knowledge graph
- Relations are characterized by operations in the vector space
e. g. $Obama = (1, 0, 0)$
  $was \text{ - born} \text{ - in} = (-1, 1, 1)$
  $Hawaii = (0, 1, 1)$

- Can be used to reason new facts
e. g. $is \text{ - president} \text{ - of} = (1, 1, 0)$
  $? = (2, 1, 0)$

- Requires the vectors of both entities to score a candidate fact
Paper 1:

Knowledge Graph and Text Jointly Embedding

Z. Wang, J. Zhang, J. Feng, Z. Chen
Microsoft Research & Sun Yat-sen University

2014

Empirical Methods on Natural Language Processing (EMNLP)
Key Idea: Joint Embeddings

- Embed entities and words into the same continuous vector space
  - By joining knowledge graph embeddings and word embeddings
  - While preserving the relations between entities and the co-occurrences of words

- **Goal**: Reason new facts given a knowledge graph and a text corpus
  - No longer require both entities to be in the knowledge graph
  → tackle the **problem** of knowledge graph coverage
Approach:

- **Model composed of:**
  1. Knowledge Model
  2. Text Model
  3. Alignment Model

- Maximize the likelihood $\mathcal{L} = \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A$

- Function to score a fact triple
  
  $\zeta(e_s, r, e_t) = \frac{1}{2} \| e_s + r - e_t \|^2$

  is expected to be close to 0 if the triple is true.
The Knowledge Model

• Relations are supervised
• “Fitting the fact triples”
• Maximize the conditional likelihood of existing fact triples
The Text Model

• Key Assumption to connect the two embeddings:

  **Relational Concurrence Assumption.** If two words $w$ and $v$ concur in some context, e.g., a window of text, then there is a relation $r_{wv}$ between the two words. That is, we can state the triplet of $(w, r_{wv}, v)$ is a fact.

  → we do not know what the relations are
  → $r_{wv}$ is a hidden variable

• Maximize the likelihood of word pair co-occurrences for a given text window (Similar to Skip-Gram word2vec)
The Alignment Model

- Problem: Entity embeddings and word embeddings are in different vector spaces
  → Any computation between them is meaningless

- Solved by aligning the two spaces into the same one, by using either:
  1. Wikipedia anchors
  2. Names of entities
Alignment by Wikipedia Anchors

- Most Wikipedia pages have a corresponding entry in Freebase

- Align surface phrase $\mathcal{u}$ of an Wikipedia anchor with its corresponding Freebase entity $e_u$


- Maximize likelihood of word-entity pairs

  - No ambiguity issues
  - Number of Anchors is small compared to number of word pairs
Alignment by Names of Entities

• Create a sub-graph containing names
  • If the name $w_e$ of an entity belongs to the word vocabulary:
    • Add triples to the graph with the entities being replaced by their names
      e.g. for $(e_s, r, e_t)$ add $(w_{e_s}, r, e_t), (e_s, r, w_{e_t}), (w_{e_s}, r, w_{e_t})$

• Maximize likelihood for the sub-graph triples

- Different entities can have the same name
- Entity can have different aliases
+ Not relying on additional data
+ Large number of triples generated by the names
Experiments + Results

- Using Freebase as the knowledge graph source and Wikipedia as a text source
- Task: Triple classification
- For each triple: construct a false triple by replacing one entity with a random one out of $\mathcal{E}$

Table 1: **Data**: triplets used in our experiments.

<table>
<thead>
<tr>
<th></th>
<th>$# \mathcal{R}$</th>
<th>$# \mathcal{E}$</th>
<th>#Triplet (Train/Valid/Test)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4,490</td>
<td>43,793,608</td>
<td>123,062,855 40,528,963 40,528,963</td>
</tr>
</tbody>
</table>

: **Triplet classification**: accuracy (%) over various types of triplets.

<table>
<thead>
<tr>
<th>Type</th>
<th>$e - e$</th>
<th>$w - e$</th>
<th>$e - w$</th>
<th>$w - w$</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>respectively</td>
<td>93.4</td>
<td>52.1</td>
<td>51.4</td>
<td>71.0</td>
<td>77.5</td>
</tr>
<tr>
<td>jointly (anchor)</td>
<td>94.4</td>
<td>67.0</td>
<td>66.7</td>
<td>79.8</td>
<td>81.9</td>
</tr>
<tr>
<td>jointly (name)</td>
<td>94.5</td>
<td>80.5</td>
<td>80.0</td>
<td>89.0</td>
<td>87.7</td>
</tr>
<tr>
<td>jointly (anchor+name)</td>
<td>95.0</td>
<td>82.0</td>
<td>81.5</td>
<td>90.0</td>
<td>88.8</td>
</tr>
</tbody>
</table>
Paper 2:

Knowledge Base Unification via Sense Embeddings and Disambiguation

C. Delli Bovi, L. Espinosa-Anke, R. Navigli
Sapienza University of Rome & Universitat Pompeu Fabra*

2015

Empirical Methods on Natural Language Processing (EMNLP)
Open Information Extraction (OIE)

- Extraction of relation tuples, typically binary relations, from plain text
- **Goal:** Acquire and formalize large quantities of knowledge

- **Problem:** Many different systems with different type inventories and no portable ontology

See also: nlp.stanford.edu/software/openie.html
**Key Idea: Join the outputs of different OIE systems**

- Draw knowledge from an *arbitrary number* of knowledge bases produced by different OIE systems
- Create a single, *unified* and fully *disambiguated* knowledge repository

**Challenges:**
- Knowledge bases have different structures, or no structure at all
- Shared semantic representation is needed
- Disambiguation
Knowledge Base Unification

- Input: set of knowledge bases \( K = KB_1, \ldots, KB_n \)
- Output: single, unified and disambiguated knowledge base \( KB^* \)
- Entities in an input knowledge base \( KB_i \) can be either:
  - **Linked** to an external inventory, already disambiguated
    - e.g. Entity “Washington”
  - **Unlinked**, only available as ambiguous mentions
    - e.g. Word “washington”

\[ \rightarrow \text{Partition input into } K_D \text{ (linked) and } K_U \text{ (unlinked)} \]
Knowledge Base Unification (2)

To align knowledge bases at the semantic level use:

a) An unified sense inventory $S$, that acts as an inventory superset
   • **BabelNet** - a lexicalized semantic network
     • provides wide coverage of entities and concepts
     • Still enables inter-resource mappings

b) A vector space model $V_S$
   • Enables **semantic** representation for every item in $S$
   • **SenseEmbed** - a semantically enhanced embedding approach
(a) Entities from linked resources can be directly mapped to corresponding entries in S
(b) Unlinked Entities need explicit disambiguation
(c) Compare relation pairs, identify alignments, merge relations sharing equivalent semantics
Experiments + Results

- Approach tested on a set of four very different knowledge bases
- Evaluated disambiguation and alignment using public gold standard datasets and human evaluations
- Resulting knowledge base:
  > 24M disambiguated triples
  > 1.9M dist. entities
  > 2.6M dist. relations

<table>
<thead>
<tr>
<th>KB-UNIFY</th>
<th>Dutta et al.</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>.852</td>
<td>.931</td>
</tr>
<tr>
<td>Recall</td>
<td>.875</td>
<td>.799</td>
</tr>
<tr>
<td>F-score</td>
<td>.864</td>
<td>.857</td>
</tr>
</tbody>
</table>

Table 4: Disambiguation results over NELL gold standard
Conclusion

Knowledge Graph and Text Jointly Embedding
→ Reason new facts by joining knowledge graph & word embeddings

Knowledge Base Unification via Sense Embeddings and Disambiguation
→ Unify and disambiguate output of many different information extraction systems
Thank you!
Questions?
Some Useful Links and References

[1] word2vec tutorials: www.tensorflow.org/tutorials/representation/word2vec
[4] Reading and Reasoning with knowledge graphs:
   www.cs.cmu.edu/~mg1/thesis.pdf
[5] Collection of knowledge graph embedding approaches:
   gist.github.com/mommi84/07f7c044fa18aaaaa7b5133230207d8d4
Table 8: Examples of cross resource relation alignments

<table>
<thead>
<tr>
<th><strong>NELL-PATTY</strong></th>
<th>$\zeta_{align}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>worksfor</td>
<td>was hired by</td>
</tr>
<tr>
<td>riveremptiesintoriver</td>
<td>tributary of</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NELL-WISENET</strong></th>
<th>$\zeta_{align}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>animaleatfood</td>
<td>feeds on</td>
</tr>
<tr>
<td>teamhomestadium</td>
<td>play their home games at</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>REVERB-WISENET</strong></th>
<th>$\zeta_{align}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>has a selection of</td>
<td>offers</td>
</tr>
<tr>
<td>had grown up in</td>
<td>was born and raised in</td>
</tr>
</tbody>
</table>