Connecting Large-Scale Knowledge Bases and Natural Language

Antoine Bordes & Jason Weston

CNRS - Univ. Tech. Compiègne & Google

UW - MSR Summer Institute 2013,
Alderbrook Resort, July 23, 2013
Knowledge Bases (KBs)

- Data is structured as a graph.
- Each node = an entity.
- Each edge = a relation.
- A relation = \((\text{sub}, \text{rel}, \text{obj})\):
  - \(\text{sub} = \text{subject}\),
  - \(\text{rel} = \text{relation type}\),
  - \(\text{obj} = \text{object}\).
- Nodes w/o features.
Example of KB: WordNet

- **WordNet**: dictionary where each entity is a sense (synset).

- Popular in NLP.

- Statistics:
  - 117k entities;
  - 20 relation types;
  - 500k relations.

- Examples:
  - (car\_NN\_1, _has_part, _wheel\_NN\_1)
  - (score\_NN\_1, _is_a, _rating\_NN\_1)
  - (score\_NN\_2, _is_a, _sheet\_music\_NN\_1)
Example of KB: Freebase

- **Freebase**: huge collaborative (hence noisy) KB.

- Part of the Google Knowledge Graph.

- Statistics:
  - > 80M of entities;
  - > 20k relation types;
  - > 1.2B relations.

- Examples:
  - (Lil Wayne, _born_in, New Orleans)
  - (Seattle, _contained_by, USA)
  - (Machine Learning, _subdiscipline, Artificial Intelligence)
Connecting KBs and Natural Language

Why?
- **Text → KB**: information extraction;
- **KB → Text**: interpretation (NER, semantic parsing), summary.

Main issue: KBs are hard to manipulate.
- **Very large dimensions**: $10^5 - 10^8$ entities, $10^7 - 10^9$ rel. types;
- **Sparse**: few valid links;
- **Noisy/incomplete**: missing/wrong relations/entities.

How?
1. Encode KBs into low-dimensional vector spaces;
2. Use these representations as KB data in text applications.
Menu

- **Context**
- **Modeling Knowledge Bases**
  - Embedding-based Models
  - Experiments on Freebase
- **Connecting KBs & Natural Language**
  - Wordnet+Text for Word-Sense Disambiguation
  - Freebase+Text for Relation Extraction
- **Conclusion**
  - What now?
Statistical Relational Learning

- **Framework:**
  - $n_s$ subjects $\{sub_i\}_{i \in [1; n_s]}$
  - $n_r$ relation types $\{rel_k\}_{k \in [1; n_r]}$
  - $n_o$ objects $\{obj_j\}_{j \in [1; n_o]}$
  - For us, $n_s = n_o = n_e$ and $\forall i \in [1; n_e], sub_i = obj_i$.  

- A relation exists for $(sub_i, rel_k, obj_j)$ if $rel_k(sub_i, obj_j) = 1$

- **Goal:** We want to model, from data,

$$P[rel_k(sub_i, obj_j) = 1]$$

(equivalent to approximate the binary tensor $X \in \{0, 1\}^{n_s \times n_o \times n_r}$)
Energy-based Learning

Two main ideas:

1. Models based on low-dimensional continuous vector embeddings for entities and relation types, learned to define a similarity criterion.

2. Stochastic training with sub-sampling of unknown relations.
Learning Representations

- Subjects and objects are represented by vectors in $\mathbb{R}^d$.
  - $\{\text{sub}_i\}_{i \in [1; n_s]} \rightarrow [s^1, \ldots, s^{n_s}] \in \mathbb{R}^{d \times n_s}$
  - $\{\text{obj}_j\}_{j \in [1; n_o]} \rightarrow [o^1, \ldots, o^{n_o}] \in \mathbb{R}^{d \times n_o}$

  For us, $n_s = n_o = n_e$ and $\forall i \in [1; n_e], \text{s}_i = \text{o}_i$.

- Rel. types = similarity operators between subjects/objects.
  - $\{\text{rel}_k\}_{k \in [1; n_r]} \rightarrow \text{operateurs} \{r_k\}_{k \in [1; n_r]}$

- Learning similarities depending on $\text{rel} \rightarrow d(\text{sub}, \text{rel}, \text{obj})$.
  (we can retrieve a probability using a transfer function)
Modeling Relations as Translations

**Intuition:** we would like that $s + r \approx o$. 

![Diagram showing relationships between entities such as Miami, Mom, Jane, John, Patti, and Austin with relations like born_in and child_of.]
Intuition: we would like that $s + r \approx o$. 

[Diagram showing relations between entities such as Austin, Mom, Patti, Miami, John, Jane with relations _born_in and _child_of.]
Modeling Relations as Translations

**Intuition:** we would like that \( s + r \approx o \).

We define the similarity measure:

\[
    d(sub, rel, obj) = \|s + r - o\|_2^2
\]

We learn \( s, r \) and \( o \) that verify that.
Stochastic Training

- Learning by **stochastic gradient descent**: one observed (true?) relation after the other.

- For each relation from the training set:
  1. we sub-sample unobserved relations (false?).
  2. we check if the similarity of the true relation is lower.
  3. if not, we update parameters of the considered relations.

- **Stopping criterion**: performance on a validation set.
Menu

- Context

- Modeling Knowledge Bases
  - Embedding-based Models
  - Experiments on Freebase

- Connecting KBs & Natural Language
  - Wordnet+Text for Word-Sense Disambiguation
  - Freebase+Text for Relation Extraction

- Conclusion
  - What now?
Chunks of Freebase

- **Data statistics:**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase15k</td>
<td>14,951</td>
<td>1,345</td>
<td>483,142</td>
<td>50,000</td>
<td>59,071</td>
</tr>
<tr>
<td>Freebase1M</td>
<td>$1 \times 10^9$</td>
<td>23,382</td>
<td>$17.5 \times 10^9$</td>
<td>50,000</td>
<td>177,404</td>
</tr>
</tbody>
</table>

- **Experimental setup:**
  - Embedding dimension: 50.
  - Training time:
    - on Freebase15k: $\approx 5h$ (on 1 CPU),
    - on Freebase1M: $\approx 1j$ (on 16 cores).
Visualization of 1,000 Entities
Visualization of 1,000 Entities - Zoom 1
Visualization of 1,000 Entities - Zoom 2
Visualization of 1,000 Entities - Zoom 3
Link Prediction

"Who influenced J.K. Rowling?"

J. K. Rowling _influenced_by ?
Link Prediction

"Who influenced J.K. Rowling?"

J. K. Rowling _influenced_by G. K. Chesterton
J. R. R. Tolkien
C. S. Lewis
Lloyd Alexander
Terry Pratchett
Roald Dahl
Jorge Luis Borges
Stephen King
Ian Fleming
Link Prediction

"Which genre is the movie WALL-E?"

WALL-E _has_genre ?
"Which genre is the movie WALL-E?"

WALL-E  \_has\_genre  Animation
Computer animation
Comedy film
Adventure film
Science Fiction
Fantasy
Stop motion
Satire
Drama
Link Prediction

On Freebase15k:

- Translation-based Model
- Tensor Factorization (RESCAL)
- Bilinear Model (SME)
- Projection Matrices (SE)

On Freebase1M, our model predicts 34% in the Top-10.
Learn Unknown Relation Types

Learning embeddings of **40 unknown relation types**.
Menu

- Context

- Modeling Knowledge Bases
  - Embedding-based Models
  - Experiments on Freebase

- Connecting KBs & Natural Language
  - Wordnet+Text for Word-Sense Disambiguation
  - Freebase+Text for Relation Extraction

- Conclusion
  - What now?
Disambiguation within a Specific Framework

Disambiguation → connect free text and the KB WordNet.

Towards open-text semantic parsing:

```
```

"A musical score accompanies a television program."

\[ \text{Semantic Role Labeling} \]

\[ ("A musical score", "accompanies", "a television program") \]

\[ \text{Preprocessing (POS, Chunking, ...)} \]

\[ (\_\text{musical\_JJ} \, \text{score\_NN} \,), \_\text{accompany\_VB} \, \_\text{television\_program\_NN} \, ) \]

\[ \text{Word-sense Disambiguation} \]

\[ (\_\text{musical\_JJ\_1} \, \text{score\_NN\_2} \,), \_\text{accompany\_VB\_1} \, \_\text{television\_program\_NN\_1} \) \]
Joint Modeling of Text and WordNet

- Text is converted into relations \((sub, rel, obj)\).
- We learn a vector for any symbol: words, entities and relation types from WordNet.
- Our system can label 37,141 words with 40,943 synsets.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Train. Ex.</th>
<th>Test Ex.</th>
<th>Labeled?</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordNet</td>
<td>146,442</td>
<td>5,000</td>
<td>No</td>
<td>synsets</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>2,146,131</td>
<td>10,000</td>
<td>No</td>
<td>words</td>
</tr>
<tr>
<td>ConceptNet</td>
<td>11,332</td>
<td>0</td>
<td>Non</td>
<td>words</td>
</tr>
<tr>
<td>Ext. WordNet</td>
<td>42,957</td>
<td>5,000</td>
<td>Yes</td>
<td>words+synsets</td>
</tr>
<tr>
<td>Unamb. Wikip.</td>
<td>981,841</td>
<td>0</td>
<td>Yes</td>
<td>words+synsets</td>
</tr>
<tr>
<td>TOTAL</td>
<td>3,328,703</td>
<td>20,000</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Experimental Results

F1-score on 5,000 test sentences to disambiguate.

![Bar chart showing F1 scores for different methods]

- Random: 26.7%
- MFS: 67.2%
- Lesk: 70.2%
- Us: 67.1%
- Us+MFS: 72.1%
Enrich WordNet

We create similarities going beyond WordNet.

"what does an army attack?"

army_NN_1  attack_VB_1  ?
Enrich WordNet

We create similarities going beyond WordNet.

"what does an army attack?"

army_NN_1  attack_VB_1    troop_NN_4
armed_service_NN_1
ship_NN_1
territory_NN_1
military_unit_NN_1
Enrich WordNet

We create similarities going beyond WordNet.

"Who or what earns money"

? earn_VB_1 money_NN_1
Enrich WordNet

We create similarities going beyond WordNet.

"Who or what earns money"

person_NN_1  earn_VB_1  money_NN_1
business_firm_NN_1
family_NN_1
payoff_NN_3
card_game_NN_1
Menu

● Context

● Modeling Knowledge Bases
  ○ Embedding-based Models
  ○ Experiments on Freebase

● Connecting KBs & Natural Language
  ○ Wordnet+Text for Word-Sense Disambiguation
  ○ Freebase+Text for Relation Extraction

● Conclusion
  ○ What now?
Relation Extraction

Given a bunch of sentences.

Text: "Alfred Hitchcock, who wrote and directed, The Birds"
"M. Hitchcock, on the set of the movie The Birds"
"Sir A. Hitchcock, the famous director of The Birds"
Relation Extraction

Given a bunch of sentences concerning the same pair of entities.

Freebase: /m/2d3rf

Text: "Alfred Hitchcock, who wrote and directed, on the set of, the famous director of The Birds" the movie The Birds" The Birds"
Relation Extraction

Goal: identify if there is a relation between them to add to the KB.

Freebase: /m/2d3rf

Text: "Alfred Hitchcock" "M. Hitchcock" "Sir A. Hitchcock" "The Birds" who wrote and directed, on the set of the famous director of the movie The Birds"
Relation Extraction

And from which type, to enrich an existing KB.

Freebase: /m/2d3rf

Text:
"Alfred Hitchcock, "M. Hitchcock, "Sir A. Hitchcock, who wrote and directed, on the set of, the famous director of The Birds" the movie The Birds" The Birds"
Jointly use Text and Freebase

- **Standard Method**: a classifier is trained to predict the relation type, given `txts` and `(sub, obj)`: 

  \[ r(txts, sub, obj) = \arg \max_{rel'} S_{txt2rel}(txts, rel') \]

- **Idea**: extract relations by using both text + available knowledge (= current KB).

- Our model of the KB forces extracted relations to agree with it: 

  \[ r(txts, sub, obj) = \arg \max_{rel'} (S_{txt2rel}(txts, rel') - d_{BC}(sub, rel', obj)) \]
Experiments on NYT+Freebase

We learn on New York Times papers and on Freebase.

Precision/recall curve for predicting relations.
Menu

- Context
- Modeling Knowledge Bases
  - Embedding-based Models
  - Experiments on Freebase
- Connecting KBs & Natural Language
  - Wordnet+Text for Word-Sense Disambiguation
  - Freebase+Text for Relation Extraction
- Conclusion
  - What now?
Encode KBs into vector spaces

- KBs are rich but need attention.

- Learn to project into vector spaces:
  - Ease their visualization;
  - Allows for link prediction (with/without ext. data);
  - Facilitate their use in other systems;
  - Compact format.

- Is that all?
Challenges

We’re just getting started:

- How to **reason**: combine logic, deduction.
- Evaluate **confidence** in predictions.
- **Summarize** KBs.
- **Fusion** KBs.
- **Connect text and KBs**: mutual interactions.
- etc.
Data/code available from my webpage.

Thanks!

antoine.bordes@hds.utc.fr
http://www.hds.utc.fr/~bordesan
References

1. **Learning Structured Embeddings of Knowledge Bases.**

2. **Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing.**

3. **A Latent Factor Model for Highly Multi-relational Data.**

4. **A Semantic Matching Energy Function for Learning with Multi-relational Data.**

5. **Irreflexive and Hierarchical Relations as Translations.**
   A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston & O. Yakhnenko. *ICML Workshop on Structured Learning, 2013*