Relation extraction for non-standard types

Benjamin Roth, Ludwig Maximilian University Munich

work with Costanza Conforti, Nina Poerner, Sanjeev Karn, Hinrich Schütze
From unstructured to structured data

- Most information about real world is unstructured.
  - “At the age of 19, Martin Luther entered the University of Erfurt.”
  - “On 2 July 1505 he was returning to Erfurt after visiting his parents in Mansfeld.”
  - Did Martin Luther live in Erfurt?

- Turning unstructured data into structured form:
  Automated knowledge base population (KBP)
  ⇒ lived_in(M_Luther, Erfurt) 0.8942
prior to founding Instagram, Kevin Systrom was one of the startup founders...
Mike Krieger co-founded Instagram with Kevin Systrom...
reminiscent of Instagram's parent company Facebook Inc. ...
the $19 billion buyout of WhatsApp by Facebook...
Why more structured data?

• Aggregate and combine information:
  • **Computational social science:**
    Detecting real world political events and trends in society
    [O’Connor, 2013, 2017]
  • **Science, e.g. Bio-informatics:**
    Extracting genome and protein interactions from research publications [Krallinger et al., 2017]
  • **Market research:**
    Extracting typical use-cases of food and products
    [Wiegand et al., 2014].

• Query structured data in dialogue systems:
  • E.g. Flight information [Seneff et al. 1991], **In-car assistants** [Madotto et al. 2018]
Traditional relation extraction

- Complex retrieval + filtering pipelines
- **Identify entities**, then predict relation
- Problems with traditional approach
  - Tagging errors, nested entities, type granularity
  - **Non-standard entity types** ($\neq$ PER, LOC, ORG,...)
Problems with traditional approach

- “[Popular Kabul]$_{ORG}$ lawmaker [Ramazan Bashardost]$_{PER}$, who camps out in a tent near parliament ...”
  city-of-residence ?

- “[Haig]$_{PER}$ attended the [US Army]$_{ORG}$ academy at [West Point]$_{LOC}$ ...”
  school-attended ?

- “[Michael Sandy]$_{PER}$ died after being [struck by a car]$_{DEATH_CAUSE}$ as he ran from ...”
  cause-of-death ?
Recall lost in pipeline

- Document retrieval: 5.6%
- Query matching: 10.4%
- Coreference: 16.6%
- Named Entity Tag missing: 24.9%
- NE Tag inexact: 5.4%
- Classifier: 37.2%

NE-tagger responsible for 30% of lost recall

Relation extraction pipeline
Relation extraction for non-standard types

• “Neural Architectures for Open-Type Relation Argument Extraction” [Roth, Conforti, Poerner, Karn, Schütze. NLE 2018]

• **Problem:** Named entity recognition

• **Solution:**
  • Relation prediction without NE tagger
  • Any subspan can be relation argument
  • **No restriction on argument types**
Query-driven relation prediction

**Query:** „Alexander Haig“

**Context:** „Haig attended the US army academy at Westpoint.“

- **Traditional approach:**
  - “[Haig]Query attended the [US Army]Answer academy at West Point ...”
    - school-attended ➔ Yes / No?
  - “[Haig]Query attended the US Army academy at [West Point]Answer ...”
    - born-in ➔ Yes / No?
  - ...

- **Proposed approach:**
  - “[Haig]Query attended the US Army academy at West Point ...”
    - school-attended ➔ Answer?
    - born-in ➔ Answer?
    - ...

Query-driven argument extraction

Query entities
Lagos Inc.
Steve Jackson ...

Relations
org:product
per:notable_work ...

Corpus

Candidate instances

Input: Lagos is a privately held jewelry company based in Philadelphia
Query match: Lagos
Relation: org:product

Instance creation: Information Retrieval
Query Matching

Argument prediction: org:product(Lagos Inc., jewelry) ...

...
Model
Query-driven argument extraction

Lagos is a jewelry company  org:product
Query-driven argument extraction

<QUERY> is a jewelry company  org:product
Query-driven argument extraction

<QUERY> is a jewelry company

org:product
Query-driven argument extraction

<table>
<thead>
<tr>
<th>Input representation</th>
<th>Relation emb. (repeat)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position embs.</td>
<td></td>
</tr>
<tr>
<td>Suffix embs.</td>
<td></td>
</tr>
<tr>
<td>Prefix embs.</td>
<td></td>
</tr>
<tr>
<td>Word embs.</td>
<td></td>
</tr>
</tbody>
</table>

<QUERY> is a jewelry company

org:product
Query-driven argument extraction

Embedding Lookup

<QUERY> is a jewelry company

org:product
Query-driven argument extraction

<QUERY> is a jewelry company

org:product
Query-driven argument extraction

<QUERY> is a jewelry company

org:product
Query-driven argument extraction

<QUERY> is a jewelry company

org:product
Query-driven argument extraction

<QUERY> is a jewelry company

org:product
Query-driven argument extraction

Predicted argument: “jewelry”

<QUERY> is a jewelry company

org:product
Encoder stage

- Encode candidate sentence into sequence of vectors.
  
  \[ h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ \ldots \]

- Variants:
  - Bi-Directional Gated Recurrent Units (RNN) [Chung, 2014]
    - Standard for encoding sequences
    - Inductive bias: global with local bias
  - Convolutional neural networks (CNN) [Collobert, 2011]
    - Efficient processing
    - Inductive bias: local
  - Self-attention/Google Transformer (ATTN) [Vaswani, 2017]
    - Relatively recently proposed sequence encoder
    - Interaction with non-transformer layers?
    - Inductive bias: weak/global
Extractor stage

• Select subspan (relational argument)

• Variants:
  • Pointer network [Vinyals, 2015]
  • Table filling [Miwa, 2014]
  • Conditional random fields tagger (CRF) [Lample, 2016]
• Predict start position, then end position
• Predictions dependent, not joint!
• Many deep QA models are pointer networks

\[
\tilde{s} = \text{ReLU}(W^s \text{Pool}(H))
\]

\[
p(\text{start} = i) = \text{softmax}(\text{MLP}([\tilde{s}; h_i]))
\]
• Decide for all pairs of start/end positions

• ~ joint version of pointer network

• Large number of negative cells

\[
p(is\_answer = True | start = i, end = j) = \sigma([h_i; h_j]^T w^{(table)})
\]
• Mark subsequence with I-tags
• Optimize global score
  • Local label scores (s)
  • Label compatibility (A)

\[ s(H, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} s_{i, y_i} \]
Data set
Data set

• First relation extraction data set with focus on non-standard types
  • entities ➔ concepts

• Requirements for selecting relations:
  • Missing argument has non-standard type. location, person, organization,…
  • Open class. Wide range of admissible values (>1000). gender,…
  • Substantial coverage. > 10000 facts in Wikidata for relation.

• Distant supervision from WikiData and Wikipedia
  • SPARQL
  • Elasticsearch
  • Entity expansion

<table>
<thead>
<tr>
<th>relation</th>
<th>id</th>
<th>#sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>per:occupation</td>
<td>P106</td>
<td>57693</td>
</tr>
<tr>
<td>per:position_held</td>
<td>P39</td>
<td>47386</td>
</tr>
<tr>
<td>per:conflict</td>
<td>P607</td>
<td>20575</td>
</tr>
<tr>
<td>per:notable_work</td>
<td>P800</td>
<td>18826</td>
</tr>
<tr>
<td>per:participant_of</td>
<td>P1344</td>
<td>14646</td>
</tr>
<tr>
<td>per:award_received</td>
<td>P166</td>
<td>13330</td>
</tr>
<tr>
<td>per:field_of_work</td>
<td>P101</td>
<td>13059</td>
</tr>
<tr>
<td>org:industry</td>
<td>P452</td>
<td>12352</td>
</tr>
<tr>
<td>per:noble_family</td>
<td>P53</td>
<td>9260</td>
</tr>
<tr>
<td>per:ethnic_group</td>
<td>P172</td>
<td>7169</td>
</tr>
<tr>
<td>org:product</td>
<td>P1056</td>
<td>6482</td>
</tr>
<tr>
<td>gpe:office</td>
<td>P1313</td>
<td>3781</td>
</tr>
</tbody>
</table>
Relations (examples)

**per:conflict**
It is named for \([\text{Henry Knox}]_Q\), an \([\text{American Revolutionary War}]_A\) general.

**per:notable_work**
In the \([\text{Steve Jackson}]_Q\) Games card game \([\text{Munchkin}]_A\), there is a card called “Dwarf Tossing”.

**per:field_of_work**
While teaching at Berkeley, \([\text{John Harsanyi}]_Q\) did extensive research in \([\text{game theory}]_A\).

**per:noble_family**
Stefan was the son of Lazar and his wife \([\text{Milica}]_Q\), a lateral line of \([\text{Nemanjić}]_A\).

**org:product**
\([\text{Lagos}]_Q\) is a privately held American \([\text{jewelry}]_A\) company.
Experiments
Comparison: all combinations

- F1-scores
- Best encoder / extractor: RNN / CRF
- Self-attention disappoints

<table>
<thead>
<tr>
<th></th>
<th>Pointer Network</th>
<th>Table Filling</th>
<th>Neural CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN</td>
<td>78.99</td>
<td>79.64</td>
<td>81.86</td>
</tr>
<tr>
<td>CNN</td>
<td>79.41</td>
<td>79.11</td>
<td>79.61</td>
</tr>
<tr>
<td>Self-Attention</td>
<td>74.49</td>
<td>75.89</td>
<td>74.35</td>
</tr>
</tbody>
</table>
Baselines

• Bi-Directional Attention-Flow (BiDAF / AllenAi, Seo et al. 2017)
  • Neural question answering model
  • Pointer mechanism
  • For our task:
    Relation is 1-word question ("org:product ?")

• Position-aware Attention (PosAtt / Stanford, Zhang et al. 2017)
  • Neural relation classification model
  • Predicts relation given marked candidate arguments
  • For our task:
    Use answers from training data to match answer candidates in dev/test.
Comparison with baselines

<table>
<thead>
<tr>
<th>Method</th>
<th>Prec</th>
<th>Rec</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiDAF</td>
<td>70.86</td>
<td>78.76</td>
<td>74.60</td>
</tr>
<tr>
<td>PosAtt</td>
<td>83.65</td>
<td>72.11</td>
<td>77.45</td>
</tr>
<tr>
<td>CNN / CRF</td>
<td>82.59</td>
<td>76.84</td>
<td>79.61</td>
</tr>
<tr>
<td>RNN / Table</td>
<td>77.92</td>
<td>81.44</td>
<td>79.64</td>
</tr>
<tr>
<td>RNN / CRF</td>
<td>82.53</td>
<td>81.19</td>
<td>81.86</td>
</tr>
</tbody>
</table>
Ablation Analysis: Input encoding

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Word</th>
<th>Affix</th>
<th>Position</th>
<th>Query</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTN+Table</td>
<td>0.16</td>
<td>1.73</td>
<td>3.54</td>
<td>4.89</td>
<td>50.76</td>
</tr>
<tr>
<td>CNN+CRF</td>
<td>2.63</td>
<td>0.16</td>
<td>0.16</td>
<td>2.87</td>
<td>74.06</td>
</tr>
<tr>
<td>RNN+CRF</td>
<td>2.73</td>
<td>0.05</td>
<td>-0.29</td>
<td>3.71</td>
<td>79.99</td>
</tr>
</tbody>
</table>

F1 improvement when (re-)adding embedding

Input: `<QUERY>` is a jewelry company
Relation: org:product
Examples

• "The Emperor's New Clothes" is a Danish fairy tale written by [Hans Christian Andersen] and first published in 1837.
  • relation: per:notable_work
  • gold answer: The Emperor's New Clothes
  • predicted: The Emperor's New Clothes

• Lucas won the 1977 *Academy Award for Film Editing* with [Richard Chew] and Paul Hirsch for her work editing "Star Wars."
  • relation: per:award_received
  • gold answer: Academy Award for Film Editing
  • predicted: Academy Award for Film Editing
Example: wrong span

• North Star (anti-slavery newspaper) North Star was a nineteenth-century \textit{anti-slavery} newspaper published from the Talman Building in Rochester, New York by \textit{abolitionist [Frederick Douglass]}\textsubscript{query}.
  • relation: \texttt{per:field_of_work}
  • gold answer: \textit{anti-slavery}
  • predicted: \textit{abolitionist}
Example: missed answer

• Game Show Network Game Show Network ( GSN ) is an American digital cable and satellite television channel that is owned as a joint venture between Sony Pictures Television ( owning a controlling 58 % interest ) and [AT & T]query Entertainment Group ( holding a 42 % ownership stake ).
  • relation: org:product_material_produced
  • gold answer: satellite television
  • predicted: -
„End-to-end“

• We successfully removed the NE-tagger.
• OK, but what about the rest of the pipeline?
• How far is it reasonable to go? How many IR-steps to replace by deep models?
  • Embed the web for each query?
  • If deep, then
    • how wide? (how many instances)
    • how deep? (interactions modeled)
  • „Deep“ re-rankers $\leftrightarrow$ reasoning with memory networks
• Needs to be carefully explored for each task!
• Interesting: very deep reasoning on limited amount of retrieved instances
Conclusion

- First work to focus on non-standard entities in relation extraction
  - Data set with 12 relations for non-standard types
- Replaced named entity tagger by deep argument extraction model
- Competitive neural encoder-extractor architecture
  - RNN, CNN, Self-Attention
  - Pointer, Table filling, CRF
- What's the best place for deep learning in pipelined architectures?