Knowledge Base Population from Text and Graphs

Lucas Sterckx
About Me

- Fourth-year PhD Student
- Research Topics:
  - Relation Extraction
  - Keyphrase Extraction
  - Sequence-to-Sequence Models
- Currently visiting Machine Intelligence Lab under supervision of **prof. Bill Byrne** and **dr. Jason Naradowsky**
NLP @ Ghent University

- Part of Internet Technology and Data Science Lab
- Initial focus: Information Retrieval
- NLP and Information Extraction since 2013
- Projects with Flanders’ major media providers
  - Named Entity Recognition
  - Named Entity Linking
  - Text Classification
  - Keyphrase extraction
Outline

1. Knowledge Bases

2. Knowledge Base Population
   a. Knowledge Extraction from Text - TAC KBP
   b. Link Prediction

3. Research Topic at Cambridge
Knowledge Bases as Labeled Graphs

Comprehensive and semantically organized **machine-readable** collection of universally relevant or domain-specific **entities, classes**, and **SPO facts** (attributes, relations)
Modern Knowledge Graphs
Applications

When did the Beatles break up?

Input Interpretation:
The Beatles (music set) end date

Result:
Friday, April 10, 1970
Knowledge Bases: Research Challenges

- **Population: Knowledge bases are incomplete**
  - Knowledge extraction from text
  - Link prediction

- **Validation: Knowledge bases contain errors**
  - Entity resolution
  - Error detection, trustworthiness

- **Interface: How to easily access knowledge**
  - Semantic parsing
  - Question answering

- **General AI: can AI emerge from Knowledge Graphs?**
  - Automatic reasoning and planning
  - Generalization and abstraction
Knowledge Bases: Research Challenges

- **Population: Knowledge bases are incomplete**
  - Knowledge extraction from text
  - Link prediction

- **Validation: Knowledge bases contain errors**
  - Entity resolution
  - Error detection, trustworthiness

- **Interface: How to easily access knowledge**
  - Semantic parsing
  - Question answering

- **General AI: can AI emerge from Knowledge Graphs?**
  - Automatic reasoning and planning
  - Generalization and abstraction
Outline

1. Knowledge Bases
2. Knowledge Base Population
   a. Knowledge Extraction from Text
   b. Link Prediction
3. Research Topic at Cambridge
Text Analysis Conference (TAC)

- **Benchmark to promote research** on automated systems that discover information about named entities and incorporate this information in a knowledge source, or database
- Continuation of previous conferences and evaluations, such as MUC and ACE
- Organised by **NIST**, sponsored by US department of Defense
- Extract **41 relations** for **50 persons** and **50 organizations** from **4 million documents**

```xml
<query id="SF_002">
  <name>PhillyInquirer</name>
  <docid>eng-NG-31-141808-9966244</docid>
  <beg>757</beg>
  <end>770</end>
  <enttype>ORG</enttype>
</query>
```

<table>
<thead>
<tr>
<th>Person Slots</th>
<th>Type</th>
<th>List?</th>
</tr>
</thead>
<tbody>
<tr>
<td>per:alternate_names</td>
<td>Name</td>
<td>Yes</td>
</tr>
<tr>
<td>per:date_of_birth</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>per:age</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>per:country_of_birth</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>per:stateorprovince_of_birth</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>per:city_of_birth</td>
<td>Name</td>
<td></td>
</tr>
</tbody>
</table>
Ghent University at TAC KBP

1. Query (Entity name + Document)
2. Alias
3. Document retrieval
4. Entity Linking
5. Documents about entities
6. Named Entity Recognition
7. Relation Extraction
8. Postprocessing
9. Output
Supervised Relation Extraction

- Extracting semantic relations between sets of grounded entities
- Train classifiers from +/- Examples

< person, city_of_birth, location >

X was born on DDDD in Y
- DEP: X←nsubjpass←on→pobj→ date→prep_in→ Y
- NER: X=PER, Y=LOC
- POS: X = NOUN, NNP; Y=NOUN, NNP
- Context: born, on, in, “born_on”

✓ Pro’s
  ○ High quality training data
  ○ Explicit negative examples

- Con’s
  ○ Expensive!
  ○ Can’t generalize to other relations and domains
Distant Supervision

- Existing Knowledge base + Unlabeled text
  1. Collect many pairs of entities **co-occurring** in sentences from the corpus (Mintz, 2009)
    - Noise

<table>
<thead>
<tr>
<th>Relation $(r)$</th>
<th>Entity 1 $(e_1)$</th>
<th>Entity 2 $(e_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>born_in</td>
<td>Barrack Obama</td>
<td>U.S.</td>
</tr>
<tr>
<td>spouse</td>
<td>Barrack Obama</td>
<td>Michelle</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mentions in free text</th>
<th>True/False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack was born in Honolulu, Hawaii, U.S.</td>
<td>✓</td>
</tr>
<tr>
<td>Barrack Obama ended U.S. military involvement in the Iraq War.</td>
<td>✗</td>
</tr>
<tr>
<td>Michelle and Barack are visiting Cuba.</td>
<td>✗</td>
</tr>
<tr>
<td>Barack and his wife Michelle are meeting with Xi Jinping</td>
<td>✓</td>
</tr>
</tbody>
</table>
Guiding Bootstrapped Relation Extractors

Knowledge Base Population using Semantic Label Propagation
(Knowledge Based Systems, Sterckx, 2016)
Guiding Bootstrapped Relation Extractors

![Diagram showing training process and relation extractor examples.]

- **(1) distant supervision**
  - Knowledge Base
  - Documents
  - Training set (labeled relation instances)

- **(2), training**
  - Relation Extractor for relation $R_1$
  - Relation Extractor for relation $R_2$
## Guiding Bootstrapped Relation Extractors

<table>
<thead>
<tr>
<th>Relation</th>
<th>Top SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>top membres employees</td>
<td>PER \textsuperscript{appos} executive \textsuperscript{prep_of} ORG</td>
</tr>
<tr>
<td></td>
<td>PER \textsuperscript{appos} chairman \textsuperscript{appos} ORG</td>
</tr>
<tr>
<td></td>
<td>ORG \textsuperscript{nn} founder \textsuperscript{prep_of} PER</td>
</tr>
<tr>
<td>children</td>
<td>PER-2 \textsuperscript{appos} son \textsuperscript{prep_of} PER-1</td>
</tr>
<tr>
<td></td>
<td>PER-1 \textsuperscript{appos} father \textsuperscript{prep_of} PER-2</td>
</tr>
<tr>
<td></td>
<td>PER-2 \textsuperscript{nn} grandson \textsuperscript{prep_of} PER-1</td>
</tr>
<tr>
<td>city of birth</td>
<td>PER \textsuperscript{rcmod} born \textsuperscript{prep_in} LOC</td>
</tr>
<tr>
<td></td>
<td>PER \textsuperscript{nsbj} mayor \textsuperscript{prep_of} LOC</td>
</tr>
<tr>
<td></td>
<td>PER \textsuperscript{appos} historian \textsuperscript{prep_from} LOC</td>
</tr>
</tbody>
</table>

### Training set

- **Knowledge Base**: \( \{ X_{11}, Y_{11} \}, \{ X_{12}, Y_{12} \}, \ldots \)
- **Documents**: \( \{ X_{21}, Y_{21} \}, \{ X_{22}, Y_{22} \}, \ldots \)

### Training

1. **(1) distant supervision**: Given labeled relation instances \( \{ R_1, R_2, \ldots \} \)
2. **(2), training**: Use relation extractors for relation \( R_1 \) and \( R_2 \)}. 


Guiding Bootstrapped Relation Extractors

<table>
<thead>
<tr>
<th>Relation</th>
<th>Top SDP</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>top_members_employees</strong></td>
<td><strong>PER  \mathsmaller{\text{\textsc{appos}}} executive  \text{\textsc{prep.of}} \rightarrow ORG</strong></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td><strong>PER  \mathsmaller{\text{\textsc{appos}}} chairman  \text{\textsc{appos}} \rightarrow ORG</strong></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td><strong>ORG  \mathsmaller{\text{\textsc{nn}}} founder  \text{\textsc{prep.of}} \rightarrow PER</strong></td>
<td>✗</td>
</tr>
<tr>
<td><strong>children</strong></td>
<td><strong>PER-2  \mathsmaller{\text{\textsc{appos}}} son  \text{\textsc{prep.of}} \rightarrow PER-1</strong></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td><strong>PER-1  \mathsmaller{\text{\textsc{appos}}} father  \text{\textsc{prep.of}} \rightarrow PER-2</strong></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td><strong>PER-2  \mathsmaller{\text{\textsc{nn}}} grandson  \text{\textsc{prep.of}} \rightarrow PER-1</strong></td>
<td>✗</td>
</tr>
<tr>
<td><strong>city_of_birth</strong></td>
<td><strong>PER  \mathsmaller{\text{\textsc{rcmod}}} born  \text{\textsc{prep.in}} \rightarrow LOC</strong></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td><strong>PER  \mathsmaller{\text{\textsc{nsbj}}} mayor  \text{\textsc{prep.of}} \rightarrow LOC</strong></td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td><strong>PER  \mathsmaller{\text{\textsc{appos}}} historian  \text{\textsc{prep.from}} \rightarrow LOC</strong></td>
<td>✗</td>
</tr>
</tbody>
</table>
Guiding Bootstrapped Relation Extractors

- Filter non-labeled patterns
  - Include weaker features in log-linear classifiers
  - Regularize
- Recall ↓↓
Guiding Bootstrapped Relation Extractors

Labeled Shortest Dependency Path

Embedding

Cosine Similarity

\[ \bar{C} = \text{CBOW}(\text{executive,of}) \]

\[ \text{Sim}(\bar{C}_t, \bar{C}_{DS}) = \frac{\bar{C}_t \cdot \bar{C}_{DS}}{|\bar{C}_t| |\bar{C}_{DS}|} \]

Training set (labeled relation instances)

(1) distant supervision

(2), training

(3) feature annotation

(4) filtering based on labeled features

(5) label propagation in semantic feature space

Relation Extractor for relation \( R_1 \)

Relation Extractor for relation \( R_2 \)
Guiding Bootstrapped Relation Extractors

Training set (labeled relation instances)

1. distant supervision

\[
R_1 \leq (x_{11}, y_{11})
\]
\[
R_2 \leq (x_{12}, y_{12})
\]
\[
\vdots
\]
\[
R_i \leq (x_{1i}, y_{1i})
\]

(2), (6) training

Relation Extractor for relation \( R_1 \)

Relation Extractor for relation \( R_2 \)

(3) feature annotation

(4) filtering based on labeled features

(5) label propagation in semantic feature space

Knowledge Base

Documents
Ghent University at TAC KBP

- **Recall ↑↑**
- **Minimal** supervision (5 min. per relation, 2u30 for TAC-KBP)
Ghent University at TAC KBP

- Human
- DeepDive (Stanford)
  - ~40 hours total writing patterns
  - Supercomputer (786 Gig RAM)
- Ensemble of feature-based classifiers neural architectures, patterns,...

(Demo)
Outline

1. Knowledge Bases
2. Knowledge Base Population
   a. Knowledge Extraction from Text - TAC KBP
   b. Link Prediction
3. Research Topic at Cambridge
Relation Extraction by Matrix Factorization (Riedel, 2013)

\[ p(\text{fact}) = p(x_{ij} = 1 | Graph) \]
[neural] vector representations,
+ Similarity, approximate inference
- Fails for little alignment, hard to fix mistakes

[symbolic] efficient ("lifted") injection of prior knowledge
+ Easy to modify
- Brittle, no generalization

combine **neural** and **symbolic** representations
to leverage advantages of both
Injecting Logical Formulae (Rocktäschel, 2015)

- Inject general 1st order formulae
  - expressed in terms of probabilities of all training facts
  - e.g. model for \( r_p \Rightarrow r_q \) : by grounding over entities

\[
p(r_p, e) \Rightarrow (r_q, e) \approx 1 - p(r_p, e)(1 - p(r_q, e))
\]

- Lessons learned:
  + joint training of facts and rules works best
  - due to grounding, only practical for few rules
Lifted implication rules (Demeester, 2016)

When is rule “\( \text{prof\_at} \Rightarrow \text{works\_for} \)” satisfied?

\[
\forall e \in \mathcal{E} : p(\text{prof\_at}(e)) \leq p(\text{works\_for}(e))
\]

\[
\sigma(v_{\text{prof\_at}} \cdot v_e) \leq \sigma(v_{\text{works\_for}} \cdot v_e)
\]

\[
v_{\text{prof\_at}} \cdot v_e \leq v_{\text{works\_for}} \cdot v_e
\]

Sufficient (even stricter) condition:

\[
\left\{ \begin{array}{l}
    v_{\text{prof\_at}} \leq v_{\text{works\_for}} \\
    \forall e \in \mathcal{E} : v_e \in \mathbb{R}^{k,+}
\end{array} \right.
\]

ordered relation embeddings

non-negative tuple embeddings

“compatibility”
Lifted implication rules - illustration

rule \ prof_at \Rightarrow \ works_for

becomes: \ \forall v_e : v_{\ prof_at} \cdot v_e \leq v_{\ works_for} \cdot v_e
Lifted implication rules - illustration

rule \( \text{prof\_at} \Rightarrow \text{works\_for} \)

becomes:

\[
\forall \nu_e : \nu_{\text{prof\_at}} \cdot \nu_e \leq \nu_{\text{works\_for}} \cdot \nu_e
\]

Given: training facts

\( \text{works\_for}(\text{Clinton, US-Gov}) \)

\( \text{prof\_at}(\text{Riedel, UCL}) \)
Given: training facts

- \text{works\_for}(\text{Clinton}, \text{US-Gov})
- \text{prof\_at}(\text{Riedel}, \text{UCL})

The rule \text{prof\_at} \Rightarrow \text{works\_for} becomes:

\[ \forall \mathbf{v}_e : \mathbf{v}_{\text{prof\_at}} \cdot \mathbf{v}_e \leq \mathbf{v}_{\text{works\_for}} \cdot \mathbf{v}_e \]
Lifted implication rules - illustration

rule prof_at ⇒ works_for
becomes: \( \forall \mathbf{v}_e : \mathbf{v}_{\text{prof_at}} \cdot \mathbf{v}_e \leq \mathbf{v}_{\text{works_for}} \cdot \mathbf{v}_e \)

Given: training facts
works_for(Clinton, US-Gov)
prof_at(Riedel, UCL)
Lifted implication rules - illustration

rule \( \text{prof@t} \Rightarrow \text{works@for} \)
becomes:

\[ \forall \nu_e : \nu_{\text{prof@t}} \cdot \nu_e \leq \nu_{\text{works@for}} \cdot \nu_e \]
Lifted implication rules - illustration

rule $\text{prof\_at} \Rightarrow \text{works\_for}$
becomes: $\forall \nu_e : \nu_{\text{prof\_at}} \cdot \nu_e \leq \nu_{\text{works\_for}} \cdot \nu_e$

\[\nu_{\text{prof\_at}} \leq \nu_{\text{works\_for}}\]

implied by $\text{prof\_at}$
implied by $\text{works\_for}$

(Riedel,UCL)

(Clinton,US-gov)

order-embeddings in relation space
Lifted implication rules - illustration

rule $\text{prof\_at} \Rightarrow \text{works\_for}$
becomes: $\forall \mathbf{v}_e : \mathbf{v}_{\text{prof\_at}} \cdot \mathbf{v}_e \leq \mathbf{v}_{\text{works\_for}} \cdot \mathbf{v}_e$

needs non-negative entity embedding space!
Lifted implication rules - illustration

rule \( \text{prof} \_	ext{at} \Rightarrow \text{works} \_	ext{for} \)

becomes:

\[ \forall \mathbf{v}_e : \mathbf{v}_{\text{prof} \_	ext{at}} \cdot \mathbf{v}_e \leq \mathbf{v}_{\text{works} \_	ext{for}} \cdot \mathbf{v}_e \]

\( \mathbf{v}_{\text{prof} \_	ext{at}} \leq \mathbf{v}_{\text{works} \_	ext{for}} \) order-embeddings in relation space

needs non-negative entity embedding space!

in practice: even better with \textbf{approximately boolean embeddings}
Lifted implication rules - in practice

Non-negative entity embeddings?
Differentiable mapping of $e \in \mathbb{R}^k$ to $\tilde{e} \in \mathbb{R}^{k,+}$

Options:
\[
\tilde{e} := \exp(e) \in \mathbb{R}^{k,+} \\
\tilde{e} := \text{ReLU}(e) \in \mathbb{R}^{k,+} \\
\tilde{e} := \sigma(e) \in (0, 1)^k
\]

strongest restriction, but works best!
“Approximately Boolean embeddings”

Ordered relation embeddings?
1 additional “lifted” loss term per implication rule:
\[
\text{minimize } \mathcal{L}_{\text{rule}} = \sum_i \max(0, [\nu_{\text{prof_at}} - \nu_{\text{prof_at}}]_i)
\]

upper bound to “grounded” loss
Experiments: Grounded versus Lifted?

More efficient:

Higher precision:

<table>
<thead>
<tr>
<th></th>
<th>0 rules</th>
<th>36 rules</th>
<th>427 rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 epoch (single CPU)</td>
<td>6.33s</td>
<td>6.76</td>
<td>6.97s</td>
</tr>
</tbody>
</table>

only 10% overhead due to rules
Outline

1. Knowledge Bases
2. Knowledge Base Population
   a. Knowledge Extraction from Text
   b. Link Prediction
3. Research Topic at Cambridge
At Cambridge: Learning to Annotate

Learn to explain hard-to-interpret text
- Genius.com
- 950.000 lyrics-annotation pairs
- English-to-English text generation
  - (Summarizing, Simplification, Paraphrasing,...)

- Statistical and Neural Machine Translation?
- Translation vs. Retrieval based?
- Evaluation ?
- Paraphrasing vs. External Knowledge ?

Remember when I used to eat sardines for dinner
Peace to Ron G, Brucey B, Kid Capri
Funkmaster Flex, Lovebug Starski (wassup?)
I’m blowing up like you thought I would
Call the crib, same number, same hood (that’s right)
It’s all good (it’s all good)
And if you don’t know, now you know, nigga

[Hook: Total]
You know very well who you are
Don’t let ’em hold you down, reach for the stars
You had a goal, but not that many
Cause you’re the only one
I’ll give you good and plenty

[Verse 2: The Notorious B.I.G.]
I made the change from a common thief
To up close and personal with Robin Leach
And I’m far from cheap, I smoke peeps all day
Spread love, it’s the Brooklyn way

Genius Annotation: Contributor
This iconic hook, sung by girl-group Total, is a
flip on the chorus of Mjumwa’s “Juicy Fruit”, which
is sampled for the beat. The lyrics of the original
are as follows:
- You know very well what you are
  - You’re my sugar thing, my chocolate star
    - I’ve had a few, but not that many
    - But you’re the only one, that gives me good
      and plenty

Biggie’s version flips the meaning, addressing his
aspiration to fame rather than romantic love, but
keeps the reference to Good & Plenty candy,
which B.I.G. presumably ate by the handful.
## Learning to Annotate

<table>
<thead>
<tr>
<th>Lyrics:</th>
<th>roilly on my arm</th>
</tr>
</thead>
<tbody>
<tr>
<td>True:</td>
<td>he’s always rocking a rolex</td>
</tr>
<tr>
<td>Retr.:</td>
<td>(5) he wears a rolex, rolex manufactures expensive wristwatches.</td>
</tr>
<tr>
<td>LSTM:</td>
<td>(4) he wears his rolex</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lyrics:</th>
<th>i blast ya cabbage</th>
</tr>
</thead>
<tbody>
<tr>
<td>True:</td>
<td>think “head of cabbage”. he’s gonna shoot you in the head.</td>
</tr>
<tr>
<td>Retr.:</td>
<td>(3) brains, post-headshot</td>
</tr>
<tr>
<td>LSTM:</td>
<td>(4) he’s gonna shoot you.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lyrics:</th>
<th>i used to have to pack a mack in back of the ac</th>
</tr>
</thead>
<tbody>
<tr>
<td>True:</td>
<td>skits called “packin a mac in the back of the ac” appeared on pun’s first two albums</td>
</tr>
<tr>
<td>Retr.:</td>
<td>(5) he’s ”packin a mac [pakinamac] in the back of the ac [meaning acura]</td>
</tr>
<tr>
<td></td>
<td>vinnie is saying that he’ll put you in the back of an accura right next to big pun’s mac.</td>
</tr>
<tr>
<td>LSTM:</td>
<td>(4) he used to have a gun in the back of the car</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lyrics:</th>
<th>behold the flyest; bentley drivers louis vuitton buyers, jet fuel abusers driving bentleys,</th>
</tr>
</thead>
<tbody>
<tr>
<td>True:</td>
<td>buying louis vuitton swag, and burning fuel in private jets are ways to show the cash that biggie and jay-z have to spend.</td>
</tr>
<tr>
<td>Retr.:</td>
<td>(2) for rich people/ballers only, prestigious (expensive as hell) name brand.</td>
</tr>
<tr>
<td>LSTM:</td>
<td>(4) he’s got a lot of money and expensive cars, and he’s got a lot of expensive brands.</td>
</tr>
</tbody>
</table>
Thank you! Questions, Comments?

Lucas Sterckx
lusterck.github.io
@lusterck
Information Extraction

Text

Entity Extraction

Relation Extraction

Resolution (Coref)

KB
Query
(Entity name + starting point)
Ghent University at TAC KBP

- **Query (Entity name + starting point)**
  - **Alias component**
  - **Information retrieval component**
  - **Documents about entities**
  - **Entity Linking Component**
Ghent University at TAC KBP

Query
(Entity name + starting point)

Alias component

Information retrieval component

Documents about entities

Entity Linking Component

Relation Extraction Component

Sentence Extraction

Filler Extraction
Ghent University at TAC KBP

Query (Entity name + starting point)

Alias component

Information retrieval component

Documents about entities

Entity Linking Component

Relation Extraction Component

Sentence Extraction

Filler Extraction

Possible slot fillers

Slot Filler Classification Component

Postprocessing component

Output
Semi-Supervised Relation Extraction

→ **Bootstrapping** (Hearst, DIPRE, Snowball, BRED)

---

Thelen and Riloff, 2002
Supervised Relation Extraction

- **LSTM or CNN-based** sentence classifiers
Link Prediction in Knowledge Graphs

$$\Pr(x_{ijk} = 1 \mid \text{Graph})$$
Knowledge Bases enable AI

- “Knowledge is Power”
  - “If a program is to perform a complex task well, it must know a great deal about the world in which it operates.”
- Graphs can be processed **efficiently** and offer a convenient abstraction
- Enabling technology for:
  - Machine reasoning
  - Disambiguation in written and spoken data
  - Semantic search in terms of entities & relations (not keywords & pages)
**Knowledge representations**

<table>
<thead>
<tr>
<th></th>
<th>is_cat</th>
<th>is_dog</th>
<th>is_animal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whisky</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Tarzan</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Snoopy</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Knowledge representations

- **neural**
  - low-dimensional representations
  - can capture similarity / hierarchy
  - can be trained from raw facts
  - Difficult to incorporate prior knowledge!

- **symbolic**

\[
\mathbf{v}_{\text{Whisky}}, \mathbf{v}_{\text{is\_cat}}, \mathbf{v}_{\text{is\_animal}} \in \mathbb{R}^k
\]

\[
\mathbf{v}_{\text{Whisky}} \approx \mathbf{v}_{\text{Tarzan}}
\]

\[
p(\text{fact}) := \sigma(\mathbf{v}_{\text{predicate}} \cdot \mathbf{v}_{\text{entity}})
\]

\[
\sigma(\mathbf{v}_{\text{is\_cat}} \cdot \mathbf{v}_{\text{Tarzan}}) \approx 1
\]

\[
\sigma(\mathbf{v}_{\text{is\_cat}} \cdot \mathbf{v}_{\text{Whisky}}) \approx 1
\]

\[
\sigma(\mathbf{v}_{\text{is\_animal}} \cdot \mathbf{v}_{\text{Tarzan}}) \approx 1
\]

\[
\sigma(\mathbf{v}_{\text{is\_animal}} \cdot \mathbf{v}_{\text{Whisky}}) \approx 1
\]

"all cats are animals"

**grounded in all entities!**
Knowledge representations

**neural**  
**symbolic**

symbols as representations (e.g. entities, predicates)

easy to integrate domain knowledge: add rules

- powerful logic reasoning tools (prolog)

  ```prolog
  is_cat(Tarzan).  % Tarzan is a cat

  is_animal(X) :- is_cat(X)  % rule: all cats are animals

  ?- is_animal(Tarzan)  % is Tarzan an animal?
  yes
  ```

- No notion of similarity (Whisky ≠ Tarzan)

  ```prolog
  ?- is_animal(Whisky)  % is Whisky an animal?
  no idea :(
  ```

not suited for approximate inference
Guiding Bootstrapped Relation Extractors

Fraction of All SDP's

Normalized sorted rank

(1) distant supervision

Training set (labeled relation instances)

Knowledge Base

Documents

56