PIDGIN: Ontology Alignment using Web-text as Interlingua

Derry Wijaya    Partha Talukdar    Tom Mitchell

Carnegie Mellon University
Ontology Alignment
Ontology Alignment

Automatically constructed Knowledge Bases (KBs)

- ontologies with hundreds of predicates (relations, categories)
- millions of $relation(entity, entity)$ triples
- examples: NELL, YAGO, Freebase, etc.
Ontology Alignment

Automatically constructed Knowledge Bases (KBs)

- ontologies with hundreds of predicates (relations, categories)
- millions of relation(entity, entity) triples
- examples: NELL, YAGO, Freebase, etc.

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)
Ontology Alignment

Automatically constructed Knowledge Bases (KBs)
• ontologies with hundreds of predicates (relations, categories)
• millions of $relation(entity, entity)$ triples
• examples: NELL, YAGO, Freebase, etc.

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

Ontology Alignment
Ontology Alignment

Automatically constructed Knowledge Bases (KBs)
• ontologies with hundreds of predicates (relations, categories)
• millions of relation(entity, entity) triples
• examples: NELL, YAGO, Freebase, etc.

<table>
<thead>
<tr>
<th>KB1</th>
<th>KB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>bornIn(Rihanna, St. Michael)</td>
<td>personBornIn(Reagan, Tampico)</td>
</tr>
<tr>
<td>bornIn(Bill Clinton, Hope)</td>
<td>personBornIn(Obama, Honolulu)</td>
</tr>
</tbody>
</table>

Ontology Alignment
• connect predicates from ontologies from different KBs
  (e.g., KB1:bornIn == KB2:personBornIn above)
Ontology Alignment

Automatically constructed Knowledge Bases (KBs)

- ontologies with hundreds of predicates (relations, categories)
- millions of relation(entity, entity) triples
- examples: NELL, YAGO, Freebase, etc.

KB1

bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2

personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

Ontology Alignment

- connect predicates from ontologies from different KBs
  (e.g., KB1:bornIn == KB2:personBornIn above)
- lots of prior work: PARIS (2011),...
Ontology Alignment

Automatically constructed Knowledge Bases (KBs)

• ontologies with hundreds of predicates (relations, categories)
• millions of relation(entity, entity) triples
• examples: NELL, YAGO, Freebase, etc.

<table>
<thead>
<tr>
<th>KB1</th>
<th>KB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>bornIn(Rihanna, St. Michael)</td>
<td>personBornIn(Reagan, Tampico)</td>
</tr>
<tr>
<td>bornIn(Bill Clinton, Hope)</td>
<td>personBornIn(Obama, Honolulu)</td>
</tr>
</tbody>
</table>

Ontology Alignment

• connect predicates from ontologies from different KBs (e.g., KB1:bornIn == KB2:personBornIn above)
• lots of prior work: PARIS (2011), ...
  • exploits predicate instance overlap
Challenges with Overlap Constraint
Challenges with Overlap Constraint

KB1: bornIn triples

KB2: personBornInCity triples

Fully overlapping (ideal)
Challenges with Overlap Constraint

KB1: bornIn triples

KB2: personBornInCity triples

Fully overlapping (ideal)

Partially overlapping
Challenges with Overlap Constraint

- Fully overlapping (ideal)
- Partially overlapping
- Non overlapping

KB1: bornIn triples
KB2: personBornInCity triples
Challenges with Overlap Constraint

Real world KBs are incomplete and biased, making instance overlap harder to enforce!

- Fully overlapping (ideal)
- Partially overlapping
- Non overlapping
Motivation: Going beyond Overlap
Motivation: Going beyond Overlap

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)
Motivation: Going beyond Overlap

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)
Motivation: Going beyond Overlap

KB1
- bornIn(Rihanna, St. Michael)
- bornIn(Bill Clinton, Hope)

KB2
- personBornIn(Reagan, Tampico)
- personBornIn(Obama, Honolulu)

Former President Bill Clinton was born in Hope ...
Motivation: Going beyond Overlap

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

Former President Bill Clinton was born in Hope ...

President Obama was born in Honolulu, while ...
Motivation: Going beyond Overlap

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

Former President Bill Clinton was born in Hope ...

President Obama was born in Honolulu, while ...

Web
Motivation: Going beyond Overlap

**KB1**
- bornIn(Rihanna, St. Michael)
- bornIn(Bill Clinton, Hope)

**KB2**
- personBornIn(Reagan, Tampico)
- personBornIn(Obama, Honolulu)

Key Idea: Use this regularity of expression in web text to connect relations from different KBs.

Former President **Bill Clinton** was born in **Hope** ...

President **Obama** was born in **Honolulu**, while ...
Motivation

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

Former President **Bill Clinton** was born in **Hope** ...

President **Obama** was born in **Honolulu**, while ...
Motivation

KB1

bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2

personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

Former President **Bill Clinton** was born in **Hope** ...

President **Obama** was born in **Honolulu**, while ...
Motivation

KB1
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

KB2
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

Former President **Bill Clinton** was born in **Hope** ...

President **Obama** was born in **Honolulu**, while ...
Motivation

**KB1**
bornIn(Rihanna, St. Michael)
bornIn(Bill Clinton, Hope)

**KB2**
personBornIn(Reagan, Tampico)
personBornIn(Obama, Honolulu)

**SVO (Interlingua)**
“Bill Clinton”, “was born in”, “Hope”
“Obama”, “was born in”, “Honolulu”

**Web**

Extract 600m Subject-Verb-Object (SVO) triples from a parsed web corpus of 230 billion tokens
Putting it all Together
PIDGIN: Alignment as Classification over Graph
PIDGIN: Alignment as Classification over Graph

Seed label specific to $KB_1$:bornIn
PIDGIN: Alignment as Classification over Graph

Seed label specific to \( KB_1: \text{bornIn} \)
PIDGIN: Alignment as Classification over Graph

Seed label specific to $KB_1$:bornIn

$(\triangle, 0.9)$

$KB_1$:<Bill_Clinton, Hope>

$(\triangle, 0.9)$

$KB_2$:<Obama, Honolulu>

$(\triangle, 0.9)$

$KB_1$:<Rihanna, St. Michael>

$(\triangle, 0.9)$

$KB_2$:<Reagan, Tampico>

$(\triangle, 0.7)$

$KB_2$:personBornInCity

$(\triangle, 0.8)$

Interlingua (SVO Data)

$(\triangle, 0.9)$

V: "was born in"

$(\triangle, 0.9)$

$(KB_1$:bornIn == $KB_2$:personBornInCity, 0.8)
PIDGIN: Alignment as Classification over Graph

Seed label specific to $KB_1$:$bornIn$

Graph-based self-supervised learning (MAD)
PIDGIN: Alignment as Classification over Graph

Seed label specific to $KB_1\text{:bornIn}$

Graph-based self-supervised learning (MAD)
- no need for human labeled data
PIDGIN: Alignment as Classification over Graph

Seed label specific to $KB_1: bornIn$

Graph-based **self**-supervised learning (MAD)
- no need for human labeled data
- transitive inference
PIDGIN: Alignment as Classification over Graph

Seed label specific to $KB_1:bornIn$

Graph-based self-supervised learning (MAD)
- no need for human labeled data
- transitive inference
- implementable in Hadoop, scalable to large graphs
Category Alignment from Relation Alignment
Category Alignment from Relation Alignment

\[ H_{\equiv}^{\text{Dom}}(c_1, c_2) = \sum_{r_1 \in R_1, c_1 = \text{Dom}(r_1), r_2 \in R_2, c_2 = \text{Dom}(r_2)} \hat{Y}_{\equiv}(r_1, r_2) \]

\[ H_{\equiv}^{\text{Ran}}(c_1, c_2) = \sum_{r_1 \in R_1, c_1 = \text{Ran}(r_1), r_2 \in R_2, c_2 = \text{Ran}(r_2)} \hat{Y}_{\equiv}(r_1, r_2) \]
Category Alignment from Relation Alignment

\[ H_{\Xi}^{\text{Dom}}(c_1, c_2) = \sum_{\substack{r_1 \in R_1, c_1 = \text{Dom}(r_1), \\ r_2 \in R_2, c_2 = \text{Dom}(r_2)}} \hat{Y}_{\Xi}(r_1, r_2) \]

\[ H_{\Xi}^{\text{Ran}}(c_1, c_2) = \sum_{\substack{r_1 \in R_1, c_1 = \text{Ran}(r_1), \\ r_2 \in R_2, c_2 = \text{Ran}(r_2)}} \hat{Y}_{\Xi}(r_1, r_2) \]

\[ A(C_1, C_2) = \{(c_1, \Xi, c_2, H_{\Xi}^{\text{Dom}}(c_1, c_2) + H_{\Xi}^{\text{Ran}}(c_1, c_2)) \mid c_1 \in C_1, \ c_2 \in C_2\} \]
## Experiments

<table>
<thead>
<tr>
<th></th>
<th>Instance Overlap</th>
<th>Interlingua</th>
<th>Joint Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARIS (PVLDB 2011)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIDGIN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments

<table>
<thead>
<tr>
<th></th>
<th>Instance Overlap</th>
<th>Interlingua</th>
<th>Joint Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PARIS</strong> (PVLDB 2011)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PIDGIN</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Main Baseline (current state-of-the-art)
# Experiments

Main Baseline (current state-of-the-art)

<table>
<thead>
<tr>
<th></th>
<th>Instance Overlap</th>
<th>Interlingua</th>
<th>Joint Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PARIS</strong> (PVLDB 2011)</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PIDGIN</strong></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Experiments

<table>
<thead>
<tr>
<th></th>
<th>Instance Overlap</th>
<th>Interlingua</th>
<th>Joint Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PARIS</strong></td>
<td>✓</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td><strong>PIDGIN</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Main Baseline (current state-of-the-art)
Experiments

<table>
<thead>
<tr>
<th></th>
<th>Instance Overlap</th>
<th>Interlingua</th>
<th>Joint Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PARIS</strong> (PVLDB 2011)</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td><strong>PIDGIN</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Main Baseline (current state-of-the-art)
Experiments

Main Baseline
(current state-of-the-art)

<table>
<thead>
<tr>
<th></th>
<th>Instance Overlap</th>
<th>Interlingua</th>
<th>Joint Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARIS (PVLDB 2011)</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>PIDGIN</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: Statistics of KBs used in experiments. We use NELL as a common target to align other KBs to, and consider only those relations in other KBs that have alignments to NELL relations (as decided by human annotators).
Relation Equivalence
## Relation Equivalence

<table>
<thead>
<tr>
<th>KB Pair</th>
<th>System</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase &amp; NELL</td>
<td>JACCARD (inst)</td>
<td>0.61</td>
<td>0.51</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>PARIS</td>
<td>0.47</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.65</td>
<td>0.61</td>
<td>0.63</td>
</tr>
<tr>
<td>Yago2 &amp; NELL</td>
<td>JACCARD (inst)</td>
<td>0.56</td>
<td>0.43</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>PARIS</td>
<td>0.67</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.52</td>
<td>0.52</td>
<td>0.52</td>
</tr>
<tr>
<td>KBP &amp; NELL</td>
<td>JACCARD (inst)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PARIS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.07</td>
<td>0.06</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Table 2: Precision, Recall and F1 scores @k=1 of Relation equivalence alignments comparing overlap based approach such as JACCARD and PARIS with PIDGIN. For each KB pair, best performance is marked in bold (See Section 5.3.1 for details)
Relation Equivalence

<table>
<thead>
<tr>
<th>KB Pair</th>
<th>System</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase &amp; NELL</td>
<td>JACCARD (inst)</td>
<td>0.61</td>
<td>0.51</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>PARIS</td>
<td>0.47</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.65</td>
<td>0.61</td>
<td>0.63</td>
</tr>
<tr>
<td>Yago2 &amp; NELL</td>
<td>JACCARD (inst)</td>
<td>0.56</td>
<td>0.43</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>PARIS</td>
<td>0.67</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.52</td>
<td>0.52</td>
<td>0.52</td>
</tr>
<tr>
<td>KBP &amp; NELL</td>
<td>JACCARD (inst)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PARIS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.07</td>
<td>0.06</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Table 2: Precision, Recall and F1 scores @k=1 of Relation equivalence alignments comparing overlap based approach such as JACCARD and PARIS with PIDGIN. For each KB pair, best performance is marked in bold (See Section 5.3.1 for details)

PIDGIN outperforms simple overlap-based baselines (JACCARD) and PARIS in all settings
Relation Subsumption
Relation Subsumption

<table>
<thead>
<tr>
<th>KB Pair</th>
<th>System</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase &amp; NELL</td>
<td>PARIS</td>
<td>0.36</td>
<td>0.08</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.8</td>
<td>0.77</td>
<td>0.79</td>
</tr>
<tr>
<td>Yago2 &amp; NELL</td>
<td>PARIS</td>
<td>0.33</td>
<td>0.06</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.65</td>
<td>0.61</td>
<td>0.63</td>
</tr>
<tr>
<td>KBP &amp; NELL</td>
<td>PARIS</td>
<td>1.0</td>
<td>0.13</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 3: Precision, Recall and F1 scores @k=1 of relation subsumption alignments comparing PARIS with PIDGIN. For each KB pair, best performance is marked in bold. (See Section 5.3.2 for details)
Relation Subsumption

<table>
<thead>
<tr>
<th>KB Pair</th>
<th>System</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase &amp; NELL</td>
<td>PARIS</td>
<td>0.36</td>
<td>0.08</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.8</td>
<td>0.77</td>
<td>0.79</td>
</tr>
<tr>
<td>Yago2 &amp; NELL</td>
<td>PARIS</td>
<td>0.33</td>
<td>0.06</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.65</td>
<td>0.61</td>
<td>0.63</td>
</tr>
<tr>
<td>KBP &amp; NELL</td>
<td>PARIS</td>
<td>1.0</td>
<td>0.13</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>PIDGIN</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 3: Precision, Recall and F1 scores @k=1 of relation subsumption alignments comparing PARIS with PIDGIN. For each KB pair, best performance is marked in bold. (See Section 5.3.2 for details)

PIDGIN infers more accurate subsumption relationships
Category Equivalence
Category Equivalence

<table>
<thead>
<tr>
<th>System</th>
<th>F1 @k=1</th>
<th>F1 @k=3</th>
<th>F1 @k=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARIS</td>
<td>0.0</td>
<td>0.47</td>
<td>0.68</td>
</tr>
<tr>
<td>PIDGIN</td>
<td>0.53</td>
<td>0.68</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 4: F1 scores @k = 1, 3 and 5 of Category Equivalence Alignments between Yago2 and NELL, comparing PARIS and PIDGIN. For each k, best performance is marked in bold. (See Section 5.3.3 for details)
Category Equivalence

<table>
<thead>
<tr>
<th>System</th>
<th>F1 @k=1</th>
<th>F1 @k=3</th>
<th>F1 @k=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARIS</td>
<td>0.0</td>
<td>0.47</td>
<td>0.68</td>
</tr>
<tr>
<td>PIDGIN</td>
<td>0.53</td>
<td>0.68</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 4: F1 scores @k = 1, 3 and 5 of Category Equivalence Alignments between Yago2 and NELL, comparing PARIS and PIDGIN. For each k, best performance is marked in bold. (See Section 5.3.3 for details)

PIDGIN discovers significantly more accurate category equivalences
Is Graph-based Joint Inference Essential?
Is Graph-based Joint Inference Essential?

Figure 4: F1 scores @\(k = 1\) and 5 of relation equivalence alignments comparing overlap-based approach (black) and PIDGIN (grey) both using the same set of resources. This demonstrates the benefit of transitivity of inference which is exploited by PIDGIN. (See Section 5.5 for details)
Is Graph-based Joint Inference Essential?

Transitivity in graph-based joint inference results in more accurate alignments

Figure 4: F1 scores @k = 1 and 5 of relation equivalence alignments comparing overlap-based approach (black) and PIDGIN (grey) both using the same set of resources. This demonstrates the benefit of transitivity of inference which is exploited by PIDGIN. (See Section 5.5 for details)
Noise Tolerance
Noise Tolerance

Noise is introduced in KB by randomly moving instances from one relation into another.
Noise Tolerance

Noise is introduced in KB by randomly moving instances from one relation into another.

Figure 5: F1 scores @k = 5 of relation equivalence alignments when varying amount of noise is introduced into the KB, comparing performance of PARIS (light grey) and PIDGIN (grey). PARIS performance drops drastically to zero as more noise is added, while PIDGIN is more robust to noise. (See Section 5.6 for details)
Noise Tolerance

PIDGIN is more noise tolerant

Noise is introduced in KB by randomly moving instances from one relation into another.

Figure 5: F1 scores @k = 5 of relation equivalence alignment when varying amount of noise is introduced into the KB. Comparing performance of PARIS (light grey) and PIDGIN (grey). PARIS performance drops drastically to zero as more noise is added, while PIDGIN is more robust to noise. (See Section for details)
By-product 1: Typing Freebase Relations using NELL Categories
By-product 1: Typing Freebase Relations using NELL Categories

Figure 6: F1 scores @k = 1, 3 and 5 when typing relations in Freebase, Yago2, and KBP with NELL categories, comparing performance of PARIS (light grey) and PIDGIN (grey). (See Section 5.7.1 for details)
By-product 1: Typing Freebase Relations using NELL Categories

Freebase:/business/industry/name is typed with NELL <company, economicSector>

Figure 6: F1 scores @k = 1, 3 and 5 when typing relations in Freebase, Yago2, and KBP with NELL categories, comparing performance of PARIS (light grey) and PIDGIN (grey). (See Section 5.7.1 for details)
By-product 1: Typing Freebase Relations using NELL Categories

Freebase:/business/industry/name is typed with NELL <company, economicSector>

PIDGIN results in more accurate cross-KB category typing of relations

Figure 6: F1 scores @k = 1, 3 and 5 when typing relations in Freebase, Yago2, and KBP with NELL categories, comparing performance of PARIS (light grey) and PIDGIN (grey). (See Section 5.7.1 for details)
By-product 2: Learned Relation-Verb Mappings
## By-product 2: Learned Relation-Verb Mappings

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Relation</th>
<th>Learned Verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase</td>
<td>/sports/sports_team/arena_stadium</td>
<td>played at, played in, defeated at, will host at, beaten at</td>
</tr>
<tr>
<td></td>
<td>/medicine/medical_treatment/side_effects</td>
<td>may promote, can cause, may produce, is worsen, exacerbate</td>
</tr>
<tr>
<td>NELL</td>
<td>drugPossiblyTreats</td>
<td>treat, relieve, reduce, help with, can help alleviate</td>
</tr>
<tr>
<td></td>
<td>PhysiologicalCondition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>politicianHoldsOffice</td>
<td>serves as, run for, became, was elected</td>
</tr>
<tr>
<td>Yago2</td>
<td>actedIn</td>
<td>played in, starred in, starred, played, portrayed in</td>
</tr>
<tr>
<td></td>
<td>isMarriedTo</td>
<td>married, met, date, wed, divorce</td>
</tr>
</tbody>
</table>

Note: These mappings are By-products of the PIDGIN system, highlighting relations learned from different knowledge bases.
Summary
Summary

• PIDGIN
Summary

• **PIDGIN**
  • uses large web-text as interlingua, overcomes limitations of previous approaches
Summary

• **PIDGIN**
  
  • uses large web-text as interlingua, overcomes limitations of previous approaches
  
  • self-supervised, scales to large data
Summary

• **PIDGIN**
  • uses large web-text as interlingua, overcomes limitations of previous approaches
  • self-supervised, scales to large data
  • effective in practice
Summary

- **PIDGIN**
  - uses large web-text as interlingua, overcomes limitations of previous approaches
  - self-supervised, scales to large data
  - effective in practice
  - code and data resources available
Tenkyu!*

Resources @
http://rtw.ml.cmu.edu/cikm2013_pidgin/

*“Thank You” in Papua New Guinean Pidgin