Predicting Semantic Relations using Global Graph Properties

Yuval Pinter and Jacob Eisenstein

@yuvalpi  @jacobbeisenstein

code: github.com/yuvalpinter/m3gm
contact: uvp@gatech.edu
Semantic Graphs

- **WordNet**-like resources are curated to describe relations between word senses
- The graph is **directed**
  - Edges have form \(<S, r, T>: <zebra, is-a, equine>\)
  - Still, some relations are symmetric
- Relation types include:
  - Hypernym (is-a) \(<zebra, r, equine>\)
  - Meronym (is-part-of) \(<tree, r, forest>\)
  - Is-instance-of \(<rome, r, capital>\)
  - Derivational Relatedness \(<nice, r, nicely>\)
Semantic Graphs - Relation Prediction

- The task of predicting relations (zebra is a <BLANK>)
- **Local** models use embeddings-based composition for scoring edges
Semantic Graphs - Relation Prediction

- The task of predicting relations (*zebra is a* <BLANK>)
- **Local** models use embeddings-based composition for scoring edges

\[
S = - (||zebra|| + ||hypernym|| - ||equine||)
\]

Translational Embeddings (transE) [Bordes et al. 2013]
Semantic Graphs - Relation Prediction

- The task of predicting relations (zebra is a <BLANK>)
- Local models use embeddings-based composition for scoring edges

\[ S = \text{zebra} \times \text{hypernym} \times \text{equine} \]

Full-Bilinear (Bilin) [Nickel et al. 2011]
Semantic Graphs - Relation Prediction

- The task of predicting relations (*zebra is_a* <BLANK>)
- Local models use embeddings-based composition for scoring edges
- Problem: task-driven method can learn unreasonable graphs

![Semantic Graph Diagram](image-url)
Incorporating a Global View

● We want to avoid unreasonable graphs
● Imposing hard constraints isn’t flexible enough
  ○ Only takes care of impossible graphs
  ○ Requires domain knowledge
● We still want the local signal to matter - it’s very strong.
Incorporating a Global View

- We want to avoid unreasonable graphs
- Imposing hard constraints isn’t flexible enough
  - Only takes care of *impossible graphs*
  - Requires domain knowledge
- We still want the local signal to matter - it’s very strong.
- Our solution: an additive, learnable **global graph score**

\[
\text{Score}(\langle \text{zebra}, \text{hypernym}, \text{equine} \rangle | \text{WordNet}) = \]

\[
S_{\text{local}}(\text{edge}) + \Delta(S_{\text{global}}(\text{WN + edge}), S_{\text{global}}(\text{WN}))
\]
Global Graph Score

- Based on a framework called Exponential Random Graph Model (ERGM)
- The score $s_{global}(WN)$ is derived from a log-linear distribution across possible graphs that have a fixed number $n$ of nodes

$$p_{ERGM}(WN) \propto \exp(\theta^T \cdot \Phi(WN))$$
Global Graph Score

- Based on a framework called Exponential Random Graph Model (ERGM)
- The score $s_{\text{global}}(WN)$ is derived from a log-linear distribution across possible graphs that have a fixed number $n$ of nodes

$$p_{\text{ERGM}}(WN) \propto \exp(\theta^T \cdot \Phi(WN))$$

- OK. What are the features?
Graph Features (Motifs)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$
Graph Features (Motifs)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: \( \frac{1}{4} = 0.25 \)
Graph Features (Motifs)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$
Graph Features (Motifs)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$
Graph Features (*Motifs*)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$
Graph Motifs (multiple relations)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$

(some) joint blue/orange motifs:

- #edges {b, o}: 9
- #2-cycles {b, o}: 1
- #3-cycles (b-o-o): 1
- #3-cycles (b-b-o): 0
- #2-paths (b-b): 4
- #2-paths (b-o): 3
- #2-paths (o-b): 4
- Transitivity (b-o-b): $\frac{2}{3} = 0.67$
Graph Motifs (multiple relations)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$

(some) joint blue/orange motifs:

- #edges {b, o}: 9
- #2-cycles {b, o}: 1
- #3-cycles (b-o-o): 1
- #3-cycles (b-b-o): 0
- #2-paths (b-b): 4
- #2-paths (b-o): 3
- #2-paths (o-b): 4
- Transitivity (b-o-b): $\frac{2}{3} = 0.67$
Graph Motifs (multiple relations)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$

(some) joint blue/orange motifs:

- #edges \{b, o\}: 9
- #2-cycles \{b, o\}: 1
- #3-cycles (b-o-o): 1
- #3-cycles (b-b-o): 0
- #2-paths (b-b): 4
- #2-paths (b-o): 3
- #2-paths (o-b): 4
- Transitivity (b-o-b): $\frac{2}{3} = 0.67$
Graph Motifs (multiple relations)

- #edges: 6
- #targets: 4
- #3-cycles: 0
- #2-paths: 4
- Transitivity: $\frac{1}{4} = 0.25$

(some) joint blue/orange motifs:

- #edges {b, o}: 9
- #2-cycles {b, o}: 1
- #3-cycles (b-o-o): 1
- #3-cycles (b-b-o): 0
- #2-paths (b-b): 4
- #2-paths (b-o): 3
- #2-paths (o-b): 4
- Transitivity (b-o-b): $\frac{2}{3} = 0.67$
ERGM Training

- Estimating the scores for all possible graphs to obtain a probability distribution is **implausible**
  - Number of possible directed graphs with $n$ nodes: $O(\exp(n^2))$
  - $n$ nodes, $R$ relations: $O(\exp(R \cdot n^2))$
  - Estimation begins to be hard at $\sim n=100$ for $R=1$. In WordNet: $n = 40K$, $R = 11$. 
ERGM Training

- Estimating the scores for all possible graphs to obtain a probability distribution is **implausible**
  - Number of possible directed graphs with $n$ nodes: $O(\exp(n^2))$
  - $n$ nodes, $R$ relations: $O(\exp(R\cdot n^2))$
  - Estimation begins to be hard at $\sim n=100$ for $R=1$. In WordNet: $n = 40K$, $R = 11$.

- Unlike other structured problems, there’s no known dynamic programming algorithm either
ERGM Training

- Estimating the scores for all possible graphs to obtain a probability distribution is **implausible**
  - Number of possible directed graphs with \( n \) nodes: \( O(\exp(n^2)) \)
  - \( n \) nodes, \( R \) relations: \( O(\exp(R*n^2)) \)
  - Estimation begins to be hard at \( \sim n=100 \) for \( R=1 \). In WordNet: \( n = 40K, R = 11 \).

- Unlike other structured problems, there’s no known dynamic programming algorithm either

**What can we do?**

- Decompose score over dyads (node pairs) in graph
- Draw and score negative sample graphs
Max-Margin Markov Graph Model (M3GM)

- Sample negative graphs from the “local neighborhood” of the true WN
Max-Margin Markov Graph Model (M3GM)

- Sample negative graphs from the "local neighborhood" of the true WN
Max-Margin Markov Graph Model (M3GM)

- Sample negative graphs from the “local neighborhood” of the true WN
Max-Margin Markov Graph Model (M3GM)

- Sample negative graphs from the “local neighborhood” of the true WN
Max-Margin Markov Graph Model (M3GM)

- Sample negative graphs from the “local neighborhood” of the true WN
- Loss = \( \text{Max} \left\{ 0, 1 + \text{score(negative sample)} - \text{score(WN)} \right\} \)
Max-Margin Markov Graph Model (M3GM)

- It’s important to choose an appropriate proposal distribution (source of the negative samples)
Max-Margin Markov Graph Model (M3GM)

- It’s important to choose an appropriate **proposal distribution** (source of the negative samples)
- We want to make things **hard** for the scorer

\[
Q(v|s, r) \propto s_{\text{local}}(<s, r, v>)
\]
Evaluation

- Dataset - WN18RR
  - No reciprocal relations (hypernym \(\Leftrightarrow\) hyponym)
  - Still includes symmetric relations
- Metrics - MRR, H@10

- Rule baseline - take symmetric if exists in train
  - Used in all models as default for symmetric relations
- Local models
  - Synset embeddings - averaged from FastText
- M3GM (re-rank top 100 from local)
  - \(~3000\) motifs, \(~900\) non-zero
Evaluation

- **Dataset - WN18RR**
  - No reciprocal relations (hypernym ⇌ hyponym)
  - Still includes symmetric relations

- **Metrics - MRR, H@10**

- **Rule baseline - take symmetric if exists in train**
  - Used in all models as default for symmetric relations

- **Local models**
  - Synset embeddings - averaged from FastText

- **M3GM (re-rank top 100 from local)**
  - ~3000 motifs, ~900 non-zero
Evaluation

- **Dataset** - WN18RR
  - No reciprocal relations (hypernym ⇔ hyponym)
  - Still includes symmetric relations
- **Metrics** - MRR, H@10
- **Rule baseline** - take symmetric if exists in train
  - Used in all models as default for symmetric relations
- **Local models**
  - Synset embeddings - averaged from FastText
- **M3GM** (re-rank top 100 from local)
  - ~ 3000 motifs, ~900 non-zero
Relation Prediction (WN18RR)

- Rule-based [Trouillon et al. 2016]
- ComplEx [Dettmers et al. 2018]
- Cony* [Nguyen et al. 2018]
- transE [Bordes et al. 2013]
- w/ M3GM

MRR
H@10
Feature Analysis

- **Motifs with heavy positive weights:**
  - Targets of *has_part*
  - Two-paths *hypernym* $\rightarrow$ *derivationally_related_form*

- **Motifs with heavy negative weights:**
  - Targets of *hypernym*
  - Two-cycles of *hypernym*
  - Target of both *has_part* and *verb_group*
Feature Analysis

- **Motifs with heavy positive weights:**
  - Targets of `has_part`
  - Two-paths `hypernym` ➔ `derivationally_related_form`

- **Motifs with heavy negative weights:**
  - Targets of `hypernym`
  - Two-cycles of `hypernym`
  - Target of both `has_part` and `verb_group`

---

- **Seen in training data**
- **Local-only prediction**
- **M3GM prediction**
- **Unseen in data**

---

- vienna
- austria
- france
- germany
Feature Analysis

- Motifs with heavy positive weights:
  - Targets of *has_part*
  - Two-paths *hypernym* $\rightarrow$ *derivationally_related_form*

- Motifs with heavy **negative** weights:
  - Targets of *hypernym*
  - Two-cycles of *hypernym*
  - Target of both *has_part* and *verb_group*
Feature Analysis

- Motifs with heavy positive weights:
  - Targets of `has_part`
  - Two-paths `hypernym` → `derivationally_related_form`

- Motifs with heavy **negative** weights:
  - Targets of `hypernym`
  - **Two-cycles of hypernym**
  - Target of both `has_part` and `verb_group`
Feature Analysis

- Motifs with heavy **positive** weights:
  - Targets of *has_part*
  - *Two-paths hypernym → derivationally_related_form*

- Motifs with heavy negative weights:
  - Targets of *hypernym*
  - Two-cycles of *hypernym*
  - Target of both *has_part* and *verb_group*

"Derivations occur in the abstract parts of the graph"

(bodega / canteen vs. shop)
Feature Analysis

● Motifs with heavy positive weights:
  ○ Targets of has_part
  ○ Two-paths hypernym → derivationally_related_form

● Motifs with heavy **negative** weights:
  ○ Targets of hypernym
  ○ Two-cycles of hypernym
  ○ Target of both has_part and verb_group

Nouns  Verbs
Future Work

- Multilingual transfers of semantic graphs
Future Work

- Multilingual transfers of semantic graphs align embeddings / translate concepts
Future Work

- Multilingual transfers of semantic graphs align embeddings / translate concepts
- Can we introduce global features to help?
Conclusion

- Global reasoning of graph features is beneficial for relation prediction
- Works well on top of strong local models
- Applicable to large graphs with dozens of relation types
- Orthogonal of word / synset embedding techniques
- Finds a wide variety of linguistic patterns in semantic graphs
Thanks

- Computational Linguistics lab @Georgia Tech

code + bonus WordNet analysis tools:
[github.com/yuvalpinter/m3gm](https://github.com/yuvalpinter/m3gm)

contact: [uvp@gatech.edu](mailto:uvp@gatech.edu)
Thanks

- Computational Linguistics lab @Georgia Tech
- Bloomberg Data Science PhD. Fellowship Program

code + bonus WordNet analysis tools: 
github.com/yuvalpinter/m3gm

contact: uvp@gatech.edu
Thanks

- Computational Linguistics lab @Georgia Tech
- Bloomberg Data Science PhD. Fellowship Program
- YOU!

code + bonus WordNet analysis tools: 
github.com/yuvalpinter/m3gm

contact: uvp@gatech.edu