CogALex-V Shared Task - **LexNET**: Integrated Path-based and Distributional Method for the Identification of Semantic Relations

Vered Shwartz and Ido Dagan

Bar-Ilan University

December 12, 2016
CogALex Shared Task

- Corpus-based identification of semantic relations
- Given two words $x$ and $y$:
  - **Subtask 1**: decide whether they are related or not:
    - e.g. related: *(misery, sadness)*, unrelated: *(misery, school)*
  - **Subtask 2**: decide what is the semantic relation that holds between them:
    - e.g. ANT: *(child, parent)*, HYPER: *(child, human)*,
      PART_OF: *(child, family)*, SYN: *(child, kid)*,
      RANDOM: *(child, mix)*
Outline

LexNET Architecture

Subtask 1 - Word Relatedness

Subtask 2 - Semantic Relation Classification
LexNET Architecture
LexNET Architecture (1)

- $(x, y)$ is represented as a feature vector, a concatenation of:
  - **Path-based features** - averaged path embedding: $\vec{v}_{paths(x,y)}$
  - **Distributional features** - $x$ and $y$'s word embeddings: $\vec{v}_w x$, $\vec{v}_w y$

An MLP classifies $(x, y)$ to the semantic relation that holds between them.
LexNET Architecture (1)

- \((x, y)\) is represented as a feature vector, a concatenation of:
  - **Path-based features** - averaged path embedding: \(\vec{v}_{paths(x,y)}\)
  - **Distributional features** - \(x\) and \(y\)’s word embeddings: \(\vec{v}_w^x, \vec{v}_w^y\)
LexNET Architecture (1)

- \((x, y)\) is represented as a feature vector, a concatenation of:
  - **Path-based features** - averaged path embedding: \(\vec{v}_{\text{paths}(x,y)}\)
  - **Distributional features** - \(x\) and \(y\)’s word embeddings: \(\vec{v}_{w_x}, \vec{v}_{w_y}\)
- An MLP classifies \((x, y)\) to the semantic relation that holds between them:

![Diagram](image-url)
LexNET Architecture (2)

**Dependency Path Representation** [Shwartz et al., 2016]:

1. An edge is a concatenation of 4 component vectors:

   - dependent lemma / dependent POS / dependency label / direction

   ![Edge Diagram]

   - be/VERB/ROOT/-

   - Embeddings:
     - lemma
     - POS
     - dependency label
     - direction

   - Average pooling

   - $\overrightarrow{o}$ paths $(x, y)$
**LexNET Architecture (2)**

**Dependency Path Representation** [Shwartz et al., 2016]:

1. An edge is a concatenation of 4 component vectors:
   
   ![Diagram of edge components]

   - dependent lemma / dependent POS / dependency label / direction

2. Edges are fed sequentially to an LSTM to get the path embedding:
Experimental Settings

- Most hyper-parameters are tuned on a validation set:
  - We split the provided train set to 90% train and 10% validation
Experimental Settings

- Most hyper-parameters are tuned on a validation set:
  - We split the provided train set to 90% train and 10% validation
  - Our split is lexical (for the $x$ slot), to avoid lexical memorization [Levy et al., 2015]

- Some hyper-parameters are fixed:
  - We use Wikipedia for a corpus (3B tokens)
  - Network’s word embeddings initialized with GloVe [Pennington et al., 2014] (6B tokens)

More on corpus size later...
Experimental Settings

- Most hyper-parameters are tuned on a validation set:
  - We split the provided train set to 90% train and 10% validation
  - Our split is lexical (for the $x$ slot), to avoid lexical memorization [Levy et al., 2015]

- Some hyper-parameters are fixed:
  - We use Wikipedia for a corpus (3B tokens)
  - Network’s word embeddings initialized with GloVe [Pennington et al., 2014] (6B tokens)
Experimental Settings

- Most hyper-parameters are tuned on a validation set:
  - We split the provided train set to 90% train and 10% validation
  - Our split is lexical (for the x slot), to avoid lexical memorization [Levy et al., 2015]

- Some hyper-parameters are fixed:
  - We use Wikipedia for a corpus (3B tokens)
  - Network’s word embeddings initialized with GloVe [Pennington et al., 2014] (6B tokens)

- More on corpus size later…
Subtask 1
Word Relatedness
Common Approaches

- Typically: compute vector similarity on $x$ and $y$’s distributional representations

- Tune a threshold to separate related and unrelated word pairs

- Most common: cosine similarity

- Achieves $F_1 = 0.747$ on the test set

- When can this go wrong?

  - the relation holds in a rare sense of $x$ or $y$: e.g. (fire, shoot)

  - the relation is weak / non-prototypical: e.g. (compact, car)
Common Approaches

- Typically: compute vector similarity on $x$ and $y$’s distributional representations
  - Tune a threshold to separate related and unrelated word pairs
- Most common: cosine similarity
  - Achieves $F_1 = 0.747$ on the test set
- When can this go wrong?
  - the relation holds in a rare sense of $x$ or $y$: e.g. (fire, shoot)
  - the relation is weak / non-prototypical: e.g. (compact, car)
Common Approaches

- Typically: compute vector similarity on $x$ and $y$’s distributional representations
  - Tune a threshold to separate related and unrelated word pairs
  - Most common: cosine similarity

- When can this go wrong?
  - The relation holds in a rare sense of $x$ or $y$: e.g. (fire, shoot)
  - The relation is weak / non-prototypical: e.g. (compact, car)

\[ F_1 = 0.747 \text{ on the test set} \]
Common Approaches

- Typically: compute vector similarity on \( x \) and \( y \)'s distributional representations
  - Tune a threshold to separate related and unrelated word pairs
  - Most common: cosine similarity
  - Achieves \( F_1 = 0.747 \) on the test set

When can this go wrong?
- the relation holds in a rare sense of \( x \) or \( y \): e.g. (fire, shoot)
- the relation is weak / non-prototypical: e.g. (compact, car)
Common Approaches

- Typically: compute vector similarity on $x$ and $y$’s distributional representations
  - Tune a threshold to separate related and unrelated word pairs
  - Most common: cosine similarity
  - Achieves $F_1 = 0.747$ on the test set

- When can this go wrong?
  - the relation holds in a rare sense of $x$ or $y$: e.g. *fire, shoot*
Common Approaches

- Typically: compute vector similarity on $x$ and $y$’s distributional representations
  - Tune a threshold to separate related and unrelated word pairs
  - Most common: cosine similarity
  - Achieves $F_1 = 0.747$ on the test set

- When can this go wrong?
  - the relation holds in a rare sense of $x$ or $y$: e.g. (fire, shoot)
  - the relation is weak / non-prototypical: e.g. (compact, car)
Subtask 1 Model

- We combine cosine similarity with LexNET:
  - Train LexNET to distinguish between related / unrelated pairs
Subtask 1 Model

- We combine cosine similarity with LexNET:
  - Train LexNET to distinguish between related / unrelated pairs
  - Compute a linear combination of cosine and LexNET:
    \[
    \text{Rel}(x, y) = w_C \cdot \cos(\vec{v}_{wx}, \vec{v}_{wy}) + w_L \cdot \vec{c}[\text{RELATED}]
    \]
Subtask 1 Model

- We combine cosine similarity with LexNET:
  - Train LexNET to distinguish between related / unrelated pairs
  - Compute a linear combination of cosine and LexNET:
    \[ \text{Rel}(x, y) = w_C \cdot \cos(\vec{v}_{wx}, \vec{v}_{wy}) + w_L \cdot \bar{c}[\text{RELATED}] \]
  - Weights, threshold and word embeddings (for Cosine) are tuned on the validation set
## Subtask 1 Results

<table>
<thead>
<tr>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority Baseline</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Random Baseline</td>
<td>0.283</td>
<td>0.503</td>
<td>0.362</td>
</tr>
<tr>
<td>ROOT18</td>
<td>-</td>
<td>-</td>
<td>0.731</td>
</tr>
<tr>
<td>Cosine Similarity</td>
<td>0.841</td>
<td>0.672</td>
<td>0.747</td>
</tr>
<tr>
<td>LexNET</td>
<td>0.754</td>
<td>0.777</td>
<td>0.765</td>
</tr>
<tr>
<td>Mach5</td>
<td>-</td>
<td>-</td>
<td>0.778</td>
</tr>
<tr>
<td>GHHH</td>
<td>-</td>
<td>-</td>
<td><strong>0.790</strong></td>
</tr>
</tbody>
</table>

**Table:** Performance scores on the test set of our method, the baselines, and the top 4 systems.

- Top performing systems achieve similar results.
Subtask 1 Results

<table>
<thead>
<tr>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority Baseline</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Random Baseline</td>
<td>0.283</td>
<td>0.503</td>
<td>0.362</td>
</tr>
<tr>
<td>ROOT18</td>
<td>-</td>
<td>-</td>
<td>0.731</td>
</tr>
<tr>
<td>Cosine Similarity</td>
<td>0.841</td>
<td>0.672</td>
<td>0.747</td>
</tr>
<tr>
<td>LexNET</td>
<td>0.754</td>
<td>0.777</td>
<td>0.765</td>
</tr>
<tr>
<td>Mach5</td>
<td>-</td>
<td>-</td>
<td>0.778</td>
</tr>
<tr>
<td>GHHH</td>
<td>-</td>
<td>-</td>
<td>0.790</td>
</tr>
</tbody>
</table>

Table: Performance scores on the test set of our method, the baselines, and the top 4 systems.

- Top performing systems achieve similar results
- Cosine baseline is strong
  - word2vec [Mikolov et al., 2013] on GoogleNews, 100B tokens
### Subtask 1 Results

<table>
<thead>
<tr>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority Baseline</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Random Baseline</td>
<td>0.283</td>
<td>0.503</td>
<td>0.362</td>
</tr>
<tr>
<td>ROOT18</td>
<td>-</td>
<td>-</td>
<td>0.731</td>
</tr>
<tr>
<td>Cosine Similarity</td>
<td>0.841</td>
<td>0.672</td>
<td>0.747</td>
</tr>
<tr>
<td>LexNET</td>
<td>0.754</td>
<td>0.777</td>
<td>0.765</td>
</tr>
<tr>
<td>Mach5</td>
<td>-</td>
<td>-</td>
<td>0.778</td>
</tr>
<tr>
<td>GHHH</td>
<td>-</td>
<td>-</td>
<td>0.790</td>
</tr>
</tbody>
</table>

**Table:** Performance scores on the test set of our method, the baselines, and the top 4 systems.

- Top performing systems achieve similar results
- Cosine baseline is strong
  - word2vec [Mikolov et al., 2013] on GoogleNews, 100B tokens
- LexNET contributes for rare senses and non-prototypical relatedness
Subtask 2
Semantic Relation Classification
Vanilla settings - train LexNET to distinguish between hypernyms, meronyms, antonyms, synonyms, and random
Vanilla settings - train LexNET to distinguish between hypernyms, meronyms, antonyms, synonyms, and random

- **Problem:**
  - The dataset is highly **imbalanced** ⇒ model overfits **random**!
Vanilla settings - train LexNET to distinguish between hypernyms, meronyms, antonyms, synonyms, and random

- **Problem:**
  - The dataset is highly **imbalanced** ⇒ model overfits **random**!

- **Solution:**
  - Use subtask 1 model to classify pairs to random / related
Vanilla settings - train LexNET to distinguish between hypernyms, meronyms, antonyms, synonyms, and random

- **Problem:**
  - The dataset is highly imbalanced ⇒ model overfits random!

- **Solution:**
  - Use subtask 1 model to classify pairs to random / related
  - Train LexNET to classify related pairs to different semantic relations
LexNET is now trained to distinguish between hypernyms, meronyms, antonyms, and synonyms
LexNET is now trained to distinguish between hypernyms, meronyms, antonyms, and synonyms

- **Problem:**
  - Synonyms are hard to recognize!
LexNET is now trained to distinguish between hypernyms, meronyms, antonyms, and synonyms

- **Problem:**
  - Synonyms are hard to recognize!
  - **Path-based:** Synonyms do not tend to occur together

- **Solution:**
  - If \((x, y)\)'s classification score for synonym and \(R\) are similar, classify as synonym only if \(x\) and \(y\) occur together less than 3 times in the corpus
LexNET is now trained to distinguish between hypernyms, meronyms, antonyms, and synonyms

▶ Problem:
▶ Synonyms are hard to recognize!
  Path-based: synonyms do not tend to occur together
  Distributional: synonyms are often mistaken for antonyms that also occur in similar contexts

▶ Solution:
▶ Add a heuristic: If \((x, y)\)'s classification score for synonym and \(R\) are similar, classify as synonym only if \(x\) and \(y\) occur together less than 3 times in the corpus
LexNET is now trained to distinguish between hypernyms, meronyms, antonyms, and synonyms

- **Problem:**
  - Synonyms are hard to recognize!
    - **Path-based:** synonyms do not tend to occur together
    - **Distributional:** synonyms are often mistaken for antonyms that also occur in similar contexts

- **Solution:**
  - Add a heuristic:
    - If \((x, y)\)’s classification score for synonym and \(R\) are similar,
LexNET is now trained to distinguish between hypernyms, meronyms, antonyms, and synonyms

- **Problem:**
  - Synonyms are hard to recognize!
    - Path-based: synonyms do not tend to occur together
    - Distributional: synonyms are often mistaken for antonyms that also occur in similar contexts

- **Solution:**
  - Add a heuristic:
    If \((x, y)’s\) classification score for synonym and \(R\) are similar, classify as synonym only if \(x\) and \(y\) occur together less than 3 times in the corpus
Subtask 2 Results

<table>
<thead>
<tr>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority Baseline</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Random Baseline</td>
<td>0.073</td>
<td>0.201</td>
<td>0.106</td>
</tr>
<tr>
<td>ROOT18</td>
<td>-</td>
<td>-</td>
<td>0.262</td>
</tr>
<tr>
<td>Mach5</td>
<td>-</td>
<td>-</td>
<td>0.295</td>
</tr>
<tr>
<td>Concatenation</td>
<td>0.469</td>
<td>0.371</td>
<td>0.411</td>
</tr>
<tr>
<td>GHHH</td>
<td>-</td>
<td>-</td>
<td>0.423</td>
</tr>
<tr>
<td>LexNET</td>
<td>0.480</td>
<td>0.418</td>
<td><strong>0.445</strong></td>
</tr>
</tbody>
</table>

Table: Performance scores on the test set of our method, the baselines, and the top 4 systems.

- Only GHHH achieves similar results
- The overall performance is very low!
Analysis

- Low results contrast the success of previous methods on common datasets
Analysis

- Low results contrast the success of previous methods on common datasets
- This can be attributed to the stricter and more informative evaluation:
  - random considered noise, excluded from $F_1$ average
Analysis

- Low results contrast the success of previous methods on common datasets
- This can be attributed to the stricter and more informative evaluation:
  - **random** considered noise, excluded from $F_1$ average
  - dataset is lexically split, disabling lexical memorization
    [Levy et al., 2015]
Analysis

- Low results contrast the success of previous methods on common datasets
- This can be attributed to the stricter and more informative evaluation:
  - **random** considered noise, excluded from $F_1$ average
  - dataset is lexically split, disabling lexical memorization
    [Levy et al., 2015]
- Motivates further research on this task!
Recap

- We presented our submission to the CogALex shared task

LexNET was the best-performing system on subtask 2 and the only system using path-based information... Performance on subtask 2 was low for all participating systems, demonstrating the difficulty of the task and motivating further research.

Thank you!
Recap

- We presented our submission to the CogALex shared task.
- The submission is based on LexNET [Shwartz and Dagan, 2016]
  - an integrated path-based and distributional method for semantic relation classification.
Recap

- We presented our submission to the CogALex shared task
- The submission is based on LexNET [Shwartz and Dagan, 2016]
  - an integrated path-based and distributional method for semantic relation classification
- LexNET was the best-performing system on subtask 2
  - and the only system using path-based information...
Recap

- We presented our submission to the CogALex shared task.
- The submission is based on LexNET [Shwartz and Dagan, 2016]
  - an integrated path-based and distributional method for semantic relation classification
- LexNET was the best-performing system on subtask 2
  - and the only system using path-based information...
- Performance on subtask 2 was low for all participating systems.
Recap

- We presented our submission to the CogALex shared task.
- The submission is based on LexNET [Shwartz and Dagan, 2016],
  - an integrated path-based and distributional method for semantic relation classification
- LexNET was the best-performing system on subtask 2,
  - and the only system using path-based information...
- Performance on subtask 2 was low for all participating systems
- Demonstrates the difficulty of the task, and motivates further research

Thank you!
Recap

- We presented our submission to the CogALex shared task.
- The submission is based on LexNET [Shwartz and Dagan, 2016]
  - an integrated path-based and distributional method for semantic relation classification.
- LexNET was the best-performing system on subtask 2
  - and the only system using path-based information...
- Performance on subtask 2 was low for all participating systems.
- Demonstrates the difficulty of the task, and motivates further research.

Thank you!
References

Do supervised distributional methods really learn lexical inference relations.
In *NAACL*.

Distributed representations of words and phrases and their compositionality.
In *NIPS*, pages 3111–3119.

Glove: Global vectors for word representation.
In *EMNLP*, pages 1532–1543.

Path-based vs. distributional information in recognizing lexical semantic relations.
*Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*.

Improving hyponymy detection with an integrated path-based and distributional method.
Appendix - Corpus Size

- **LexNET:**
  - Main corpus: Wikipedia (3B tokens)
  - Pre-trained GloVe embeddings [Pennington et al., 2014], trained on Wikipedia + Gigaword 5 (6B tokens)

- **Cosine:** pre-trained word2vec embeddings [Mikolov et al., 2013], trained on Google News (100B tokens)