Acquiring Lexical Semantic Knowledge

...And exploring ways to use it in applications

Vered Shwartz

Talk at Google Research IL, November 9, 2017
Outline

Introduction and Motivation

Acquiring Lexical Knowledge
  Recognizing Semantic Relations between Nouns
  Acquiring Predicate Paraphrases

Using Lexical Knowledge in Sentence-level Applications
  The Holy Grail: Recognizing Textual Entailment
What is “lexical knowledge”?

Knowledge about how words relate to each other. Valuable for making inferences:

"pets are allowed" \implies "dogs are allowed"

"restaurant in Tel Aviv" \implies "restaurant in Israel"

Vered Shwartz · Acquiring Lexical Semantic Knowledge · Talk at Google Research IL, November 9, 2017
What is “lexical knowledge”?

Knowledge about how words relate to each other.

"pets are allowed" ⇒ "dogs are allowed"?

"restaurant in Tel Aviv" ⇒ "restaurant in Israel"?

"restaurant in Tel Aviv"
What is “lexical knowledge”?

- Knowledge about how words \textit{relate} to each other.

- Valuable for making inferences:
  - “pets are allowed” $\Rightarrow$ “dogs are allowed”
  - “dogs are allowed” \textit{??} “pets are allowed”
What is “lexical knowledge”?

- Knowledge about how words relate to each other.

- Valuable for making inferences:
  - “pets are allowed” ⇒ “dogs are allowed”
  - “dogs are allowed” ?? “pets are allowed”
  - “restaurant in Tel Aviv” ⇒ “restaurant in Israel”
  - “restaurant in Israel” ?? “restaurant in Tel Aviv”
Word Embeddings

First, let’s get this off the table: “why not just use word embeddings?”
Word Embeddings

First, let’s get this off the table: **“why not just use word embeddings?”**

- Word embeddings are great in capturing semantic relatedness!
Introduction and Motivation

Word Embeddings

First, let’s get this off the table: “why not just use word embeddings?”

- Word embeddings are great in capturing semantic relatedness!
- ...but they mix all semantic relations together.
Word Embeddings

- To illustrate, take famous texts and replace nouns with their word2vec neighbours:¹

```
"I have a daydream that my four little kids will one week live in a country where they will not be judged by the hues of their epidermis but by the Classical.com of their protagonist."
```

¹More examples here: https://goo.gl/LJHzbi
Word Embeddings

To illustrate, take famous texts and replace nouns with their word2vec neighbours:\(^1\)

"I have a daydream that my four little kids will one week live in a country where they will not be judged by the hues of their epidermis but by the Classical.com of their protagonist wrong sense

\(^1\)More examples here: https://goo.gl/LJHzbi
Acquiring Lexical Knowledge
Recognizing Semantic Relations between Nouns
The Hypernymy Detection Task

- We first focused on hypernymy
  - The hyponym is a subclass of / instance of the hypernym
  - (cat, animal), (Google, company)
The Hypernymy Detection Task

- We first focused on hypernymy
  - The hyponym is a subclass of / instance of the hypernym
  - (cat, animal), (Google, company)

- Given two terms, $x$ and $y$, decide whether $y$ is a hypernym of $x$
  - in some senses of $x$ and $y$, e.g. (apple, fruit), (apple, company)
Corpus-based Hypernymy Detection

Hypernymy Detection

prior work

path-based
Corpus-based Hypernymy Detection

Hyponymy Detection

- path-based
- distributional

prior work
Corpus-based Hypernymy Detection

Hypernymy Detection

- path-based
- distributional

prior work

neural

path-based

our work
Corpus-based Hypernymy Detection

Hypernymy Detection

prior work
path-based
distributional

neural
path-based

Integrated Model
“HypeNET”

our work
Prior Methods

Hypernymy Detection

- Path-based
- Distributional

Integrated Model “HypeNET”

Prior work

Our work
Distributional Approach

Hypernymy Detection

Prior work

path-based

neural path-based

Integrated Model "HypeNET"

Our work

distributional
Supervised Distributional Methods

- Recognize the relation between $x$ and $y$ based on their *separate* occurrences in the corpus
Supervised Distributional Methods

- Recognize the relation between \( x \) and \( y \) based on their *separate* occurrences in the corpus
- Represent \((x, y)\) as a feature vector, based on the terms’ embeddings:
  - Concatenation \( \vec{x} \oplus \vec{y} \) [Baroni et al., 2012]
  - Difference \( \vec{y} - \vec{x} \) [Roller et al., 2014, Weeds et al., 2014]

Train a classifier to predict whether \( y \) is a hypernym of \( x \)

Achieved very good results on common hypernymy detection datasets

Is it a solved task?

Probably not. They don’t learn the relation between \( x \) and \( y \), but mostly that \( y \) is a prototypical hypernym [Levy et al., 2015].

\( \text{e.g. that } (x, \text{fruit}) \) or \( (x, \text{animal}) \) are always hypernyms
Supervised Distributional Methods

- Recognize the relation between $x$ and $y$ based on their separate occurrences in the corpus.
- Represent $(x, y)$ as a feature vector, based on the terms’ embeddings:
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- Train a classifier to predict whether $y$ is a hypernym of $x$.
Supervised Distributional Methods

- Recognize the relation between $x$ and $y$ based on their separate occurrences in the corpus
- Represent $(x, y)$ as a feature vector, based of the terms’ embeddings:
  - Concatenation $\bar{x} \oplus \bar{y}$ [Baroni et al., 2012]
  - Difference $\bar{y} - \bar{x}$ [Roller et al., 2014, Weeds et al., 2014]
- Train a classifier to predict whether $y$ is a hypernym of $x$
- Achieved very good results on common hypernymy detection datasets
Supervised Distributional Methods

- Recognize the relation between $x$ and $y$ based on their separate occurrences in the corpus
- Represent $(x, y)$ as a feature vector, based on the terms’ embeddings:
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- Train a classifier to predict whether $y$ is a hypernym of $x$

- Achieved very good results on common hypernymy detection datasets
- Is it a solved task?
Supervised Distributional Methods

- Recognize the relation between $x$ and $y$ based on their separate occurrences in the corpus
- Represent $(x, y)$ as a feature vector, based on the terms’ embeddings:
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- Train a classifier to predict whether $y$ is a hypernym of $x$

- Achieved very good results on common hypernymy detection datasets
- Is it a solved task?
- Probably not. They don’t learn the relation between $x$ and $y$, but mostly that $y$ is a prototypical hypernym [Levy et al., 2015].
  - e.g. that $(x, fruit)$ or $(x, animal)$ are always hypernyms
Path-based Approach

Hypernymy Detection

Prior work

Path-based

Distributional

Neural
Path-based

Integrated Model
“HypeNET”

Our work
Path-based Approach

- Recognize the relation between $x$ and $y$ based on their *joint* occurrences in the corpus

Hearst Patterns [Hearst, 1992] - patterns connecting $x$ and $y$ may indicate that $y$ is a hypernym of $x$. e.g. $X$ or other $Y$, $X$ is a $Y$, $Y$, including $X$.
Path-based Approach

- Recognize the relation between $x$ and $y$ based on their *joint* occurrences in the corpus
- Hearst Patterns [Hearst, 1992] - patterns connecting $x$ and $y$ may indicate that $y$ is a hypernym of $x$
  - e.g. $X$ or other $Y$, $X$ is a $Y$, $Y$, including $X$
**Path-based Approach**

- Recognize the relation between $x$ and $y$ based on their *joint* occurrences in the corpus
- Hearst Patterns [Hearst, 1992] - patterns connecting $x$ and $y$ may indicate that $y$ is a hypernym of $x$
  - e.g. *X or other Y, X is a Y, Y, including X*
- Patterns can be represented using dependency paths:

```
apple  is  a  fruit
  NOUN     VERB    DET    NOUN
```

Hearst Patterns [Hearst, 1992] - patterns connecting $x$ and $y$ may indicate that $y$ is a hypernym of $x$. For example, *X or other Y, X is a Y, Y, including X*. Patterns can be represented using dependency paths.
Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:

  Features: all dependency paths that connected x and y in a corpus:
  ▲ ▼
  X and other Y such Y as X
Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Features: all dependency paths that connected \( x \) and \( y \) in a corpus:

\[
\begin{array}{cccccccc}
0 & 0 & \ldots & 58 & 0 & \ldots & 97 & 0 & \ldots & 0 \\
\end{array}
\]

\[\uparrow \quad \uparrow\]

X and other Y \quad such Y as X
Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Features: all dependency paths that connected \( x \) and \( y \) in a corpus:
    
    \[
    \begin{array}{cccccccc}
    0 & 0 & \ldots & 58 & 0 & \ldots & 97 & 0 & \ldots & 0
    \end{array}
    \]

    \[\uparrow\ \ \uparrow\]

    \( X \) and other \( Y \)  \ such \( Y \) as \( X \)

- Trained a logistic regression classifier to predict hypernymy
Path-based Approach Issues

- The feature space is too sparse:
Path-based Approach Issues

- The feature space is too sparse:
  - Similar paths share no information:
    - $X$ inc. is $Y$
    - $X$ group is $Y$
    - $X$ organization is $Y$
Path-based Approach Issues

- The feature space is too sparse:
  - Similar paths share no information:
    - X inc. is a Y
    - X group is a Y
    - X organization is a Y

- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:

```
X corporation is a Y
```
Path-based Approach Issues

- The feature space is too sparse:
  - Similar paths share no information:
    - X inc. is a Y
    - X group is a Y
    - X organization is a Y

- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - its POS tag

```
COMPOUND
   ↘
    X

NOUN

NSUBJ
   ↘
    is

VERB

X

NOUN

is

VERB

Y

DET

NOUN

ATTR

X

NOUN
```
Path-based Approach Issues

- The feature space is too sparse:
  - Similar paths share no information:
    - $X$ inc. is a $Y$
    - $X$ group is a $Y$
    - $X$ organization is a $Y$

- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - a wild-card

\[
\begin{array}{c}
\text{COMPOUND} \\
\downarrow
\end{array} \quad \begin{array}{c}
\text{NSUBJ} \\
\downarrow
\end{array} \quad \begin{array}{c}
\text{ATTR} \\
\downarrow
\end{array} \quad \begin{array}{c}
\text{DET} \\
\downarrow
\end{array} \quad \begin{array}{c}
\text{NOUN} \\
\downarrow
\end{array}
\]

\[
\begin{array}{c}
X \\
\text{NOUN}
\end{array} \quad \begin{array}{c}
* \\
\text{VERB}
\end{array} \quad \begin{array}{c}
is \\
\text{NOUN}
\end{array} \quad \begin{array}{c}
a \\
\text{DET}
\end{array} \quad \begin{array}{c}
Y \\
\text{NOUN}
\end{array}
\]
Path-based Approach Issues

- The feature space is too sparse:
  - Similar paths share no information:
    - X inc. is a Y
    - X group is a Y
    - X organization is a Y
  - PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
    - a wild-card

- Some of these generalizations are too general:
  - X is defined as Y $\approx$ X is described as Y via X is VERB as Y
Path-based Approach Issues

- The feature space is too sparse:
  - Similar paths share no information:
    - $X$ inc. is a $Y$
    - $X$ group is a $Y$
    - $X$ organization is a $Y$

- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - a wild-card

Some of these generalizations are too general:
- $X$ is defined as $Y \approx X$ is described as $Y$ via $X$ is VERB as $Y$
- $X$ is defined as $Y \neq X$ is rejected as $Y$
**HypeNET**: Integrated Path-based and Distributional Method

[Shwartz et al., 2016]

---

Prior work

Path-based

Distributional

Neural path-based

Integrated Model

“HypeNET”
First Step: Improving Path Representation

Hypernymy Detection

path-based

neural path-based

Integrated Model “HypeNET”

distributional

prior work

our work
Path Representation (1/2)

1. Split each path to edges

\[
X \text{ is a } Y \Rightarrow \text{‘}X/NOUN/nsubj/> \text{ be/VERB/ROOT/- } Y/NOUN/attr/<\text{‘} \Rightarrow \text{‘}X/NOUN/nsubj/> \text{‘ be/VERB/ROOT/-‘ } Y/NOUN/attr/<\text{‘}
\]

- Each edge consists of 4 components:
  - dependent lemma
  - dependent POS
  - dependency label
  - direction
Path Representation (1/2)

1. Split each path to edges

\[ X \text{ is a } Y \Rightarrow \]
\[ 'X/NOUN/nsubj'/> be/VERB/ROOT/- Y/NOUN/attr/' \Rightarrow \]
\[ 'X/NOUN/nsubj'/> 'be/VERB/ROOT/-' 'Y/NOUN/attr/' \]

- Each edge consists of 4 components:
  - dependent lemma / dependent POS / dependency label / direction

- We learn embedding vectors for each component
  - Lemma embeddings are initialized with pre-trained word embeddings
1. Split each path to edges

\[ X \text{ is a } Y \Rightarrow \]
\[ \text{‘X/NOUN/nsubj/'} \text{ be/VERB/ROOT/'} \text{ Y/NOUN/attr/'} \]
\[ \text{‘X/NOUN/nsubj/'} \text{ ‘be/VERB/ROOT/’ ‘Y/NOUN/attr/'} \]

- Each edge consists of 4 components:
  - dependent lemma / dependent POS / dependency label / direction

- We learn embedding vectors for each component
  - Lemma embeddings are initialized with pre-trained word embeddings
  - The edge’s vector is the concatenation of its components’ vectors:

\[ \text{be/VERB/ROOT/-} \]

- Generalization: similar edges should have similar vectors!
Path Representation (2/2)

2. Feed the edges sequentially to an LSTM

- Use the last output vector as the path embedding
- The LSTM may focus on edges that are more informative for the classification task, while ignoring others
The LSTM encodes a single path
- Each term-pair has multiple paths
  - Represent a term-pair as its averaged path embedding
- Classify for hypernymy (path-based network):

```
X/NOUN/nsubj/ >
be/VERB/ROOT/-
Y/NOUN/attr/ <
```

```
X/NOUN/dobj/ >
define/VERB/ROOT/-
as/ADP/prep/<
Y/NOUN/pobj/ <
```

Embeddings:
- lemma
- POS
- dependency label
- direction

Term-pair Classifier

\( \hat{O}_p \)

average pooling

\( V_{xy} \)

\( (x, y) \) classification (softmax)
Second Step: Integrating Distributional Information

Hypernymy Detection

prior work

path-based

neural
path-based

Integrated Model
“HypeNET”

distributional

our work
Second Step: Integrating Distributional Information

- Integrated network: add distributional information
  - Simply concatenate \( x \) and \( y \)'s word embeddings to the averaged path

- Classify for hypernymy (integrated network):

```
\vec{o}_p ...
\vec{v}_w \vec{w} \vec{x} \vec{w} \vec{y} \vec{v}_x \vec{y} \vec{v}_xy 
```

(x, y) paths in Path LSTM

Vered Shwartz • Acquiring Lexical Semantic Knowledge • Talk at Google Research IL, November 9, 2017
## Results

- On a new dataset, built from knowledge resources

<table>
<thead>
<tr>
<th>method</th>
<th>precision</th>
<th>recall</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path-based</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snow</td>
<td>0.843</td>
<td>0.452</td>
<td>0.589</td>
</tr>
<tr>
<td>Snow + GEN</td>
<td>0.852</td>
<td>0.561</td>
<td>0.676</td>
</tr>
<tr>
<td>HypeNET Path-based</td>
<td>0.811</td>
<td>0.716</td>
<td>0.761</td>
</tr>
<tr>
<td>Distributional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best Supervised</td>
<td>0.901</td>
<td>0.637</td>
<td>0.746</td>
</tr>
<tr>
<td>Combined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HypeNET Integrated</td>
<td>0.913</td>
<td>0.890</td>
<td>0.901</td>
</tr>
</tbody>
</table>

- Path-based:
  - Compared to Snow + Snow with PATTY style generalizations
  - Our method outperforms path-based baselines with improved recall
Results

- On a new dataset, built from knowledge resources

<table>
<thead>
<tr>
<th>method</th>
<th>precision</th>
<th>recall</th>
<th>(F_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path-based</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snow</td>
<td>0.843</td>
<td>0.452</td>
<td>0.589</td>
</tr>
<tr>
<td>Snow + GEN</td>
<td>0.852</td>
<td>0.561</td>
<td>0.676</td>
</tr>
<tr>
<td>HypeNET Path-based</td>
<td>0.811</td>
<td>0.716</td>
<td>0.761</td>
</tr>
<tr>
<td>Distributional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best Supervised</td>
<td>0.901</td>
<td>0.637</td>
<td>0.746</td>
</tr>
<tr>
<td>Combined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HypeNET Integrated</td>
<td>0.913</td>
<td>0.890</td>
<td>0.901</td>
</tr>
</tbody>
</table>

- The integrated method substantially outperforms both path-based and distributional methods
Analysis - Path Representation (1/2)

- Identify hypernymy-indicating paths:
  - **Baselines**: according to logistic regression feature weights
Analysis - Path Representation (1/2)

- Identify hypernymy-indicating paths:
  - **Baselines**: according to logistic regression feature weights
  - **HypeNET**: measure path contribution to positive classification:

- Take the top scoring paths according to $\text{softmax}(W \cdot [\vec{0}, \vec{o}_p, \vec{0}])[1]$
Analysis - Path Representation (2/2)

- Snow’s method finds certain common paths:
  - X company is a Y
  - X ltd is a Y

PATTY-style generalizations find very general, possibly noisy paths:

HypeNET makes fine-grained generalizations:

- X association is a Y
- X co. is a Y
- X company is a Y
- X corporation is a Y
- X foundation is a Y
- X group is a Y
...
Analysis - Path Representation (2/2)

Snow’s method finds certain common paths:

- X company is a Y
- X ltd is a Y

PATTY-style generalizations find very general, possibly noisy paths:

- X NOUN is a Y
Analysis - Path Representation (2/2)

- Snow’s method finds certain common paths:
  
  \[ X \text{ company is a } Y \]
  \[ X \text{ ltd is a } Y \]

- PATTY-style generalizations find very general, possibly noisy paths:
  
  \[ X \text{ NOUN is a } Y \]

- HypeNET makes fine-grained generalizations:
  
  \[ X \text{ association is a } Y \]
  \[ X \text{ co. is a } Y \]
  \[ X \text{ company is a } Y \]
  \[ X \text{ corporation is a } Y \]
  \[ X \text{ foundation is a } Y \]
  \[ X \text{ group is a } Y \]
  ...
Other Semantic Relations
Recognizing Lexical Semantic Relations

- Given two terms, $x$ and $y$, decide what is the semantic relation that holds between them (if any)
  - in some senses of $x$ and $y$
  - e.g. both *fruit* and *company* are hypernyms of *apple*
LexNET - Multiple Semantic Relation Classification
[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations:
  hypernymy, meronymy, co-hyponymy, event, attribute, synonymy, antonymy, random

\[ \vec{V}_{xy} \]

\[ \vec{V}_{wx} \]

\[ \vec{V}_{wy} \]

\[ \vec{V}_{paths(x,y)} \]

\[ (x, y) \]

classification (softmax)
Results and Analysis

- LexNET outperforms individual path-based and distributional methods
Results and Analysis

- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is small when lexical memorization is enabled
Results and Analysis

- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is small when lexical memorization is enabled
- It is prominent in the following scenarios:
  - $x$ or $y$ are polysemous, e.g. $mero:(\text{piano}, \text{key})$. 
Results and Analysis

- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is small when lexical memorization is enabled
- It is prominent in the following scenarios:
  - $x$ or $y$ are polysemous, e.g. \texttt{mero:(piano, key)}.
  - the relation is not prototypical, e.g. \texttt{event:(cherry, pick)}.
Results and Analysis

- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is small when lexical memorization is enabled
- It is prominent in the following scenarios:
  - $x$ or $y$ are polysemous, e.g. $\text{mero:}(\text{piano}, \text{key})$.
  - the relation is not prototypical, e.g. $\text{event:}(\text{cherry}, \text{pick})$.
  - $x$ or $y$ are rare, e.g. $\text{hyper:}(\text{mastodon}, \text{proboscidean})$. 
Results and Analysis

- LexNET outperforms individual path-based and distributional methods.
- Path-based contribution over distributional info is small when lexical memorization is enabled.
- It is prominent in the following scenarios:
  - $x$ or $y$ are polysemous, e.g. $mero: (piano, key)$.
  - The relation is not prototypical, e.g. $event: (cherry, pick)$.
  - $x$ or $y$ are rare, e.g. $hyper: (mastodon, proboscidean)$.
- Thanks to the path representation, such relations are captured even with a single meaningful co-occurrence of $x$ and $y$. 
Limitations

- All methods and baselines are bad in recognizing synonyms and antonyms.
Limitations

- All methods and baselines are bad in recognizing synonyms and antonyms.
  - **Path-based:**
    - Synonyms do not tend to occur together
Limitations

- All methods and baselines are bad in recognizing synonyms and antonyms.
  - **Path-based:**
    - Synonyms do not tend to occur together
    - Antonyms occur in similar paths as co-hyponyms: *hot and cold, cats and dogs*
Limitations

- All methods and baselines are bad in recognizing synonyms and antonyms.
  - **Path-based:**
    - Synonyms do not tend to occur together
    - Antonyms occur in similar paths as co-hyponyms: *hot* and *cold*, *cats* and *dogs*
  - **Distributional:**
    - Synonyms and antonyms occur in similar contexts:
      “go down in the *elevator/lift*”, “it is *hot/cold* today”
Limitations

- All methods and baselines are bad in recognizing synonyms and antonyms.
  - Path-based:
    - Synonyms do not tend to occur together
    - Antonyms occur in similar paths as co-hyponyms: *hot and cold, cats and dogs*
  - Distributional:
    - Synonyms and antonyms occur in similar contexts: “go down in the elevator/lift”, “it is hot/cold today”
- [Nguyen et al., 2017] used the method successfully to distinguish only between synonyms and antonyms.
Limitations

- All methods and baselines are bad in recognizing synonyms and antonyms.
  - **Path-based:**
    - Synonyms do not tend to occur together
    - Antonyms occur in similar paths as co-hyponyms: *hot and cold, cats and dogs*
  - **Distributional:**
    - Synonyms and antonyms occur in similar contexts:
      “go down in the elevator/lift”, “it is hot/cold today”

- [Nguyen et al., 2017] used the method successfully to distinguish only between synonyms and antonyms.

- [Rajana et al., 2017] integrated morphological cues (negated prefixes) to distinguish antonymy from other relations.
Interpreting Noun-Compounds

- Given a noun-compound $w_1w_2$, classify the relation between the head $w_2$ and the modifier $w_1$
  - to one of a set of pre-defined relations
  - e.g. olive oil $\rightarrow$ source, baby oil $\rightarrow$ purpose
Interpreting Noun-Compounds

- Given a noun-compound $w_1 w_2$, classify the relation between the head $w_2$ and the modifier $w_1$
  - to one of a set of pre-defined relations
  - e.g. olive oil $\rightarrow$ source, baby oil $\rightarrow$ purpose

- Similar yet different from semantic relation classification:
  - We are interested in the relation between olive and oil in the context of the noun-compound, not in general
Interpreting Noun-Compounds

Previous Approaches

- **Paraphrasing**: Find joint corpus occurrences of $w_1$ and $w_2$, use paraphrases as features
  - e.g.: $[w_2]$ *obtained from* $[w_1]$ (*oil obtained from olives*)
Interpreting Noun-Compounds

Previous Approaches

- **Paraphrasing:** Find joint corpus occurrences of $w_1$ and $w_2$, use paraphrases as features
  - e.g.: $[w_2]$ obtained from $[w_1]$ (*oil obtained from olives*)
  - **Problem:** too sparse. e.g. $[w_2]$ extracted from $[w_1]$
Interpreting Noun-Compounds

Previous Approaches

- **Paraphrasing:** Find joint corpus occurrences of $w_1$ and $w_2$, use paraphrases as features
  - e.g.: $[w_2]$ obtained from $[w_1]$ (*oil obtained from olives*)
  - Problem: too sparse. e.g. $[w_2]$ extracted from $[w_1]$

- **Distributional:** Noun-compound representation as a function of $w_1$ and $w_2$ distributional representations
Interpreting Noun-Compounds

Previous Approaches

- **Paraphrasing**: Find joint corpus occurrences of $w_1$ and $w_2$, use paraphrases as features
  - e.g.: $[w_2]$ obtained from $[w_1]$ (*oil obtained from olives*)
  - Problem: too sparse. e.g. $[w_2]$ extracted from $[w_1]$

- **Distributional**: Noun-compound representation as a function of $w_1$ and $w_2$ distributional representations
  - Problem: memorizes common relations of $w_1$ and $w_2$ separately
Interpreting Noun-Compounds
[Shwartz and Waterson, in preparation]

- We applied LexNET to this task
Interpreting Noun-Compounds
[Shwartz and Waterson, in preparation]

- We applied LexNET to this task
- LexNET improves performance:
  - On a lexical split dataset (i.e. not allowing lexical memorization)
  - On a new, challenging dataset we created
Interpreting Noun-Compounds
[Shwartz and Waterson, in preparation]

- We applied LexNET to this task
- LexNET improves performance:
  - On a lexical split dataset (i.e. not allowing lexical memorization)
  - On a new, challenging dataset we created
- Performs worse than the baseline when lexical memorization is possible
Interpreting Noun-Compounds
[Shwartz and Waterson, in preparation]

- We applied LexNET to this task
- LexNET improves performance:
  - On a lexical split dataset (i.e. not allowing lexical memorization)
  - On a new, challenging dataset we created
- Performs worse than the baseline when lexical memorization is possible
- In general, the task is very difficult:
  - Lots of relations
  - Some relations have no indicative paths (e.g. non-compositional)
Acquiring Predicate Paraphrases
Acquiring Predicate Paraphrases from News Tweets
[Shwartz et al., 2017]

- Binary verbal predicate paraphrases

\[
\begin{align*}
[a]_0 \text{ introduce } [a]_1 \\
[a]_0 \text{ appoint } [a]_1 \\
[a]_0 \text{ die at } [a]_1 \\
[a]_0 \text{ hit } [a]_1 \\
[a]_0 \text{ be investigate } [a]_1 \\
[a]_0 \text{ eliminate } [a]_1 \\
[a]_0 \text{ announce } [a]_1 \\
[a]_0 \text{ quit after } [a]_1 \\
[a]_0 \text{ announce as } [a]_1 \\
[a]_0 \text{ threaten } [a]_1 \\
[a]_0 \text{ die at } [a]_1 \\
[a]_0 \text{ double down on } [a]_1 \\
[a]_0 \text{ kill } [a]_1 \\
[a]_0 \text{ approve } [a]_1 \\
\text{seize } [a]_0 \text{ at } [a]_1 \\
[a]_0 \text{ welcome } [a]_1 \\
[a]_0 \text{ to become } [a]_1 \\
[a]_0 \text{ pass away at } [a]_1 \\
[a]_0 \text{ sink to } [a]_1 \\
[a]_0 \text{ be probe } [a]_1 \\
[a]_0 \text{ slash } [a]_1 \\
[a]_0 \text{ unveil } [a]_1 \\
[a]_0 \text{ resign after } [a]_1 \\
[a]_0 \text{ to become } [a]_1 \\
[a]_0 \text{ warn } [a]_1 \\
[a]_0 \text{ live until } [a]_1 \\
[a]_0 \text{ stand by } [a]_1 \\
[a]_0 \text{ shoot } [a]_1 \\
[a]_0 \text{ pass } [a]_1 \\
to grab [a]_0 \text{ at } [a]_1
\end{align*}
\]
Acquiring Predicate Paraphrases from News Tweets
[Shwartz et al., 2017]$^2$

- Binary verbal predicate paraphrases
- Extracted from Twitter

<table>
<thead>
<tr>
<th>[a]_0 introduce [a]_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>[a]_0 appoint [a]_1</td>
</tr>
<tr>
<td>[a]_0 die at [a]_1</td>
</tr>
<tr>
<td>[a]_0 hit [a]_1</td>
</tr>
<tr>
<td>[a]_0 be investigate [a]_1</td>
</tr>
<tr>
<td>[a]_0 eliminate [a]_1</td>
</tr>
<tr>
<td>[a]_0 announce [a]_1</td>
</tr>
<tr>
<td>[a]_0 quit after [a]_1</td>
</tr>
<tr>
<td>[a]_0 announce as [a]_1</td>
</tr>
<tr>
<td>[a]_0 threaten [a]_1</td>
</tr>
<tr>
<td>[a]_0 die at [a]_1</td>
</tr>
<tr>
<td>[a]_0 double down on [a]_1</td>
</tr>
<tr>
<td>[a]_0 kill [a]_1</td>
</tr>
<tr>
<td>[a]_0 approve [a]_1</td>
</tr>
<tr>
<td>seize [a]_0 at [a]_1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>[a]_0 welcome [a]_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>[a]_0 to become [a]_1</td>
</tr>
<tr>
<td>[a]_0 pass away at [a]_1</td>
</tr>
<tr>
<td>[a]_0 sink to [a]_1</td>
</tr>
<tr>
<td>[a]_0 be probe [a]_1</td>
</tr>
<tr>
<td>[a]_0 slash [a]_1</td>
</tr>
<tr>
<td>[a]_0 unveil [a]_1</td>
</tr>
<tr>
<td>[a]_0 resign after [a]_1</td>
</tr>
<tr>
<td>[a]_0 to become [a]_1</td>
</tr>
<tr>
<td>[a]_0 warn [a]_1</td>
</tr>
<tr>
<td>[a]_0 live until [a]_1</td>
</tr>
<tr>
<td>[a]_0 stand by [a]_1</td>
</tr>
<tr>
<td>[a]_0 shoot [a]_1</td>
</tr>
<tr>
<td>[a]_0 pass [a]_1</td>
</tr>
<tr>
<td>to grab [a]_0 at [a]_1</td>
</tr>
</tbody>
</table>

$^2$Available at https://github.com/vered1986/Chirps
Acquiring Predicate Paraphrases from News Tweets
[Shwartz et al., 2017]

- Binary verbal predicate paraphrases
- Extracted from Twitter
- Ever-growing resource: currently around 1.5M paraphrases

Available at https://github.com/vered1986/Chirps
Assumptions

Main assumption: redundant news headlines of the same event are likely to describe it with different words [Shinyama et al., 2002, Barzilay and Lee, 2003].
Assumptions

- **Main assumption:** redundant news headlines of the same event are likely to describe it with different words [Shinyama et al., 2002, Barzilay and Lee, 2003].
- **This work:** propositions extracted from tweets discussing news events, published on the same day, that agree on their arguments, are predicate paraphrases.
Resource Collection

1. Collect News Tweets
2. Extract Propositions
3. Generate Paraphrase Instances
4. Generate Types
5. Resource Release
Query the Twitter Search API for news tweets in English

- Amazon is buying Whole Foods in $13.7B
- Amazon to acquire Whole Foods Market in deal valued at nearly $14 billion
  ...
Resource Collection

- Extract propositions from tweets using PropS [Stanovsky et al., 2016]
- Get binary verbal predicate templates, and apply argument reduction [Stanovsky and Dagan, 2016]

[Amazon] buy [Whole Foods]
[Amazon] acquire [Whole Foods Market]

...
We consider two predicates as paraphrases if:

1. They appear on the same day.
2. Each of their arguments aligns with a unique argument in the other predicate.

Two levels of argument matching: **strict** (exact match / short edit distance) and **loose** (partial token matching / WordNet synonyms)
Resource Collection

Heuristic score for a predicate paraphrase type:

\[ p_1 = [a]_0 \text{buy} [a]_1, \quad p_2 = [a]_0 \text{acquire} [a]_1 \]

\[ s(p_1, p_2) = \text{count}(p_1, p_2) \cdot \left(1 + \frac{\text{days}(p_1, p_2)}{N}\right) \]

- \( \text{count}(p_1, p_2) \) assigns high scores for frequent paraphrases
- \( N \) - number of days since the resource collection begun
- \( \frac{\text{days}(p_1, p_2)}{N} \) eliminates noise from two arguments participating in different events on the same day

1) Last year when Chuck Berry turned 90; 2) Chuck Berry dies at 90
Resource Collection

- We release our resource daily, with two files:
  - **Instances**: predicates, arguments and tweet IDs.
  - **Types**: predicate paraphrase pair types ranked in a descending order according to the heuristic accuracy score.
Using Lexical Knowledge in Sentence-level Applications
Sentence-level Inference

- RTE: given a premise $p$ and a hypothesis $h$, can a reader reading $p$ infer that $h$ is likely true? [Dagan et al., 2013].
  - Very small datasets, unsuitable for today’s neural models
Sentence-level Inference

- RTE: given a premise $p$ and a hypothesis $h$, can a reader reading $p$ infer that $h$ is likely true? [Dagan et al., 2013].
  - Very small datasets, unsuitable for today’s neural models
- NLI: natural language inference - 3-way classification for entailment, neutral, and contradiction:
  - SNLI [Bowman et al., 2015]
  - MultiNLI [Williams et al., 2017]
Knowledge Required for Sentence-level Inference

Lexical Overlap → Lexical Semantic Relations → Context-sensitive Lexical Inference → World Knowledge → Reasoning
Knowledge Required for Sentence-level Inference

- **Premise:**
  *Three young women embrace while displaying baked goods in kitchen.*

- **Hypothesis:**
  *Three young women embrace while they show off their baked goods to potential buyers.*
Knowledge Required for Sentence-level Inference

- **Premise:**
  *Three young women embrace while displaying baked goods in kitchen.*

- **Hypothesis:**
  *Three young women embrace while they show off their baked goods to potential buyers.*
Knowledge Required for Sentence-level Inference

- **Premise:**
  Elderly bald man with a beard **playing** the guitar in a **band**.

- **Hypothesis:**
  There are **people making music** together.
Knowledge Required for Sentence-level Inference

- **Premise:**
  A performer standing on a platform in *Times Square*.

- **Hypothesis:**
  *The performer is in New York.*
Knowledge Required for Sentence-level Inference

- **Premise:**
  In a train station, an attractive woman in a blue skirt and jacket, surrounded by her luggage, passes time with a crossword.

- **Hypothesis:**
  A woman is doing a crossword puzzle while waiting for a train.
**Existing Solutions**

- Recent neural models are good with lexical overlap and reasonable with semantic relations.
Existing Solutions

- Recent neural models are good with lexical overlap and reasonable with semantic relations.
- Many papers claim to solve “reasoning”, but their success stems from the dataset being too easy.
  - e.g. high correlation between lexical overlap and entailment
Existing Solutions

- Recent neural models are good with lexical overlap and reasonable with semantic relations.
- Many papers claim to solve “reasoning”, but their success stems from the dataset being too easy.
  - e.g. high correlation between lexical overlap and entailment

YOU KEEP SAYING "REASONING"

I DON'T THINK IT MEANS WHAT YOU THINK IT MEANS
Our Vision

- **Goal**: improve sentence-level inference with lexical knowledge
Our Vision

- **Goal**: improve sentence-level inference with lexical knowledge
- **Means**: inject knowledge into neural models to combine the best of both worlds
Our Vision

**P:** An elderly man is drinking orange juice at a cafe.

**H:** An old man is sipping a beverage.

1. Extract propositions:

   **Premise**
   - [man] drink [orange juice]
   - [man] be at [cafe]
   - [man] be [elderly]

   **Hypothesis**
   - [man] sip [beverage]
   - [man] be [old]
Our Vision

**P**: An elderly man is drinking orange juice at a cafe.

**H**: An old man is sipping a beverage.

2. Align arguments based on lexical semantic relations:

Premise

- [man]$_1$ drink [orange juice]$_2$
- [man]$_1$ be at [cafe]$_4$
- [man]$_1$ be [elderly]$_3$

Hypothesis

- [man]$_1$ sip [beverage]$_2$
- [man]$_1$ be [old]$_3$
Our Vision

**P**: An elderly man is drinking orange juice at a cafe.

**H**: An old man is sipping a beverage.

3. Align propositions based on argument and predicate entailment:

Premise:
- [man]₁ drink [orange juice]₂
- [man]₁ be at [cafe]₄
- [man]₁ be [elderly]₃

Hypothesis:
- [man]₁ sip [beverage]₂
- [man]₁ be [old]₃
Our Vision

**P:** An elderly man is drinking orange juice at a cafe.

**H:** An old man is sipping a beverage.

4. Make a sentence-level decision based on proposition alignment:

Premise:
- [man]$_1$ drink [orange juice]$_2$
- [man]$_1$ be at [cafe]$_4$
- [man]$_1$ be [elderly]$_3$

Hypothesis:
- [man]$_1$ sip [beverage]$_2$
- [man]$_1$ be [old]$_3$

Entailment
Limitations and Drawbacks

- Difficult to show improvement on existing datasets
  - Current SOTA: SNLI - 90% accuracy, MultiNLI - 80% accuracy
  - Most models work on surface level, no external knowledge
Limitations and Drawbacks

- Difficult to show improvement on existing datasets
  - Current SOTA: SNLI - 90% accuracy, MultiNLI - 80% accuracy
  - Most models work on surface level, no external knowledge

- Tools and knowledge introduce new errors:
  - Parsing
  - Proposition extraction
  - Automatically-extracted lexical knowledge
Thank You!
References I


References II


