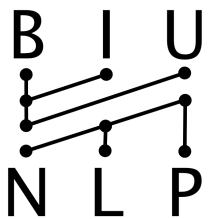
How are you two related?

Corpus-based Learning of Lexical Semantic Relations

Vered Shwartz

Workshop on WordNets and Word Embeddings
9th Global Wordnet Conference

11 January 2018



What's in the talk?

- WordNet

What's in the talk?

- WordNet
  - Applications in NLP
  - What's missing?

What's in the talk?

- WordNet
  - Applications in NLP
  - What's missing?
- Word Embeddings
  - Do they obviate the usage of lexicons like WordNet?

What's in the talk?

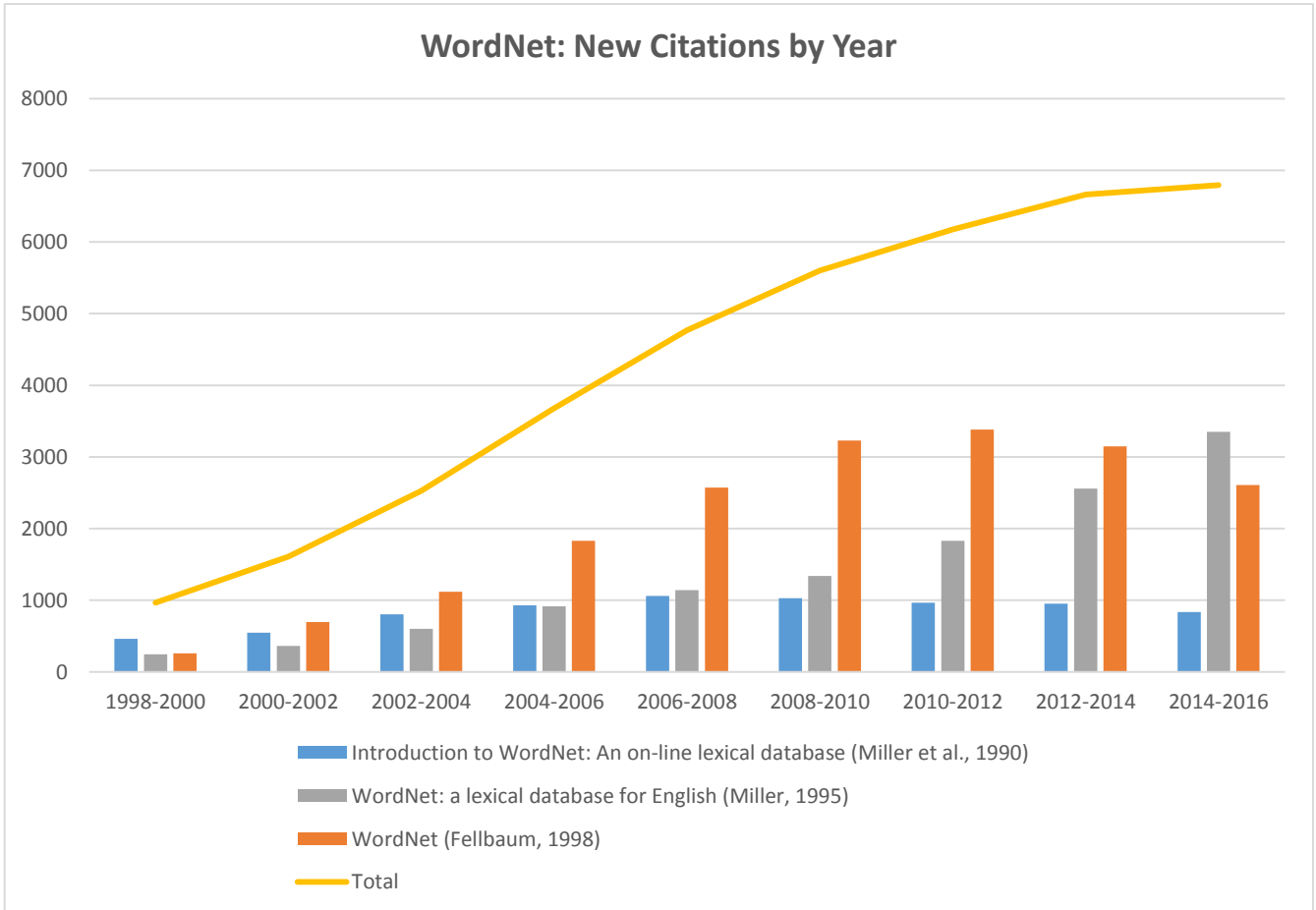
- WordNet
  - Applications in NLP
  - What's missing?
- Word Embeddings
  - Do they obviate the usage of lexicons like WordNet?
- **Using Word Embeddings to Enhance WordNet**
  - Corpus-based learning of lexical-semantic relations

What's in the talk?

- WordNet
  - Applications in NLP
  - What's missing?
- Word Embeddings
  - Do they obviate the usage of lexicons like WordNet?
- **Using Word Embeddings to Enhance WordNet**
  - Corpus-based learning of lexical-semantic relations
- Encoding WordNet into Word Embeddings
  - How can we keep leveraging WordNet in DL-based applications?

Usage of WordNet

Usage of WordNet



Usage of WordNet

- Extensively used in applications that make inferences

Usage of WordNet

- Extensively used in applications that make inferences
- Various relations help dealing with lexical variability, e.g.:
  - *"pets are allowed"* $\Rightarrow$ *"dogs are allowed"* (hypernym)

Usage of WordNet

- Extensively used in applications that make inferences
- Various relations help dealing with lexical variability, e.g.:
  - *"pets are allowed"* $\Rightarrow$ *"dogs are allowed"* (hypernym)
  - *"restaurant in Japan"* $\Rightarrow$ *"restaurant in Asia"* (holonym)

Usage of WordNet

- Extensively used in applications that make inferences
- Various relations help dealing with lexical variability, e.g.:
  - *"pets are allowed"* $\Rightarrow$ *"dogs are allowed"* (hypernym)
  - *"restaurant in Japan"* $\Rightarrow$ *"restaurant in Asia"* (holonym)
  - *"restaurant in Japan"* $\neq$ *"restaurant in China"* (co-hyponym)

Usage of WordNet

- Extensively used in applications that make inferences
- Various relations help dealing with lexical variability, e.g.:
  - *"pets are allowed"* $\Rightarrow$ *"dogs are allowed"* (hypernym)
  - *"restaurant in Japan"* $\Rightarrow$ *"restaurant in Asia"* (holonym)
  - *"restaurant in Japan"* $\neq$ *"restaurant in China"* (co-hyponym)
  - *"good restaurant"* $\neq$ *"bad restaurant"* (antonym)

Example Application 1 - Question Answering

Question

"When did Donald Trump visit in Alabama?"

Example Application 1 - Question Answering

Question

"When did Donald Trump visit in Alabama?"

Candidate Passages

1. Trump visited Huntsville on September 23.
2. Trump visited Mississippi on June 21.

Example Application 1 - Question Answering

Question

"When did Donald Trump visit in Alabama?"

Candidate Passages

1. Trump visited Huntsville on September 23.
2. Trump visited Mississippi on June 21.

Knowledge

holonym:(Huntsville, Alabama), co-hyponym:(Mississippi, Alabama).

Example Application 2 - Recognizing Textual Entailment

Premise

A boy is hitting a baseball

Hypotheses

Example Application 2 - Recognizing Textual Entailment

Premise

A boy is hitting a baseball

Hypotheses

1. A **child** is hitting a baseball $\Rightarrow$ ENTAILMENT: *hypernym:(boy, child)*

Example Application 2 - Recognizing Textual Entailment

Premise

A boy is hitting a baseball

Hypotheses

1. A **child** is hitting a baseball ⇒ ENTAILMENT: *hypernym:(boy, child)*
2. A boy is **missing** a baseball ⇒ CONTRADICTION: *antonym:(hitting, missing)*

Example Application 2 - Recognizing Textual Entailment

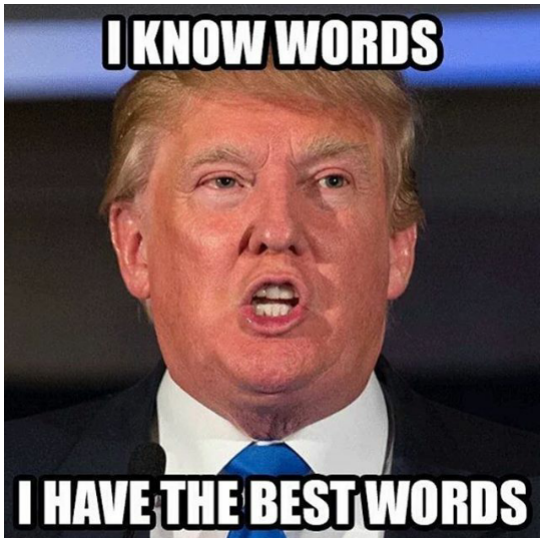
Premise

A boy is hitting a baseball

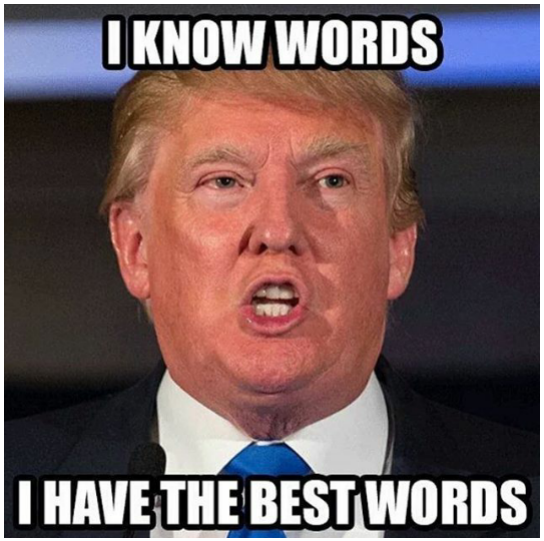
Hypotheses

1. A **child** is hitting a baseball ⇒ ENTAILMENT: *hypernym:(boy, child)*
2. A boy is **missing** a baseball ⇒ CONTRADICTION: *antonym:(hitting, missing)*
3. A **girl** is hitting a baseball ⇒ NEUTRAL: *co-hyponym:(boy, girl)*

What's Missing from WordNet?



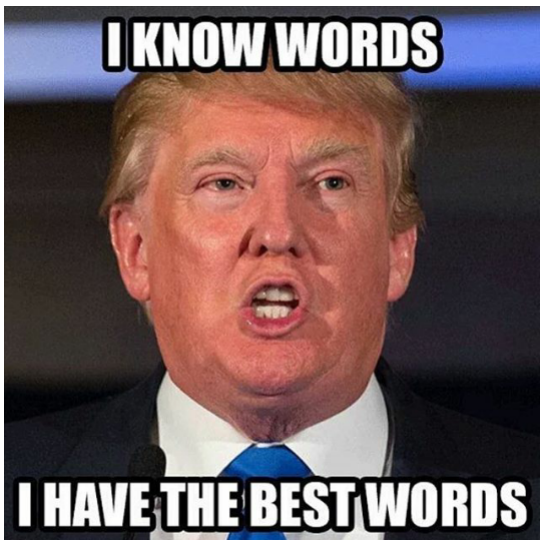
What's Missing from WordNet?



- Many named entities

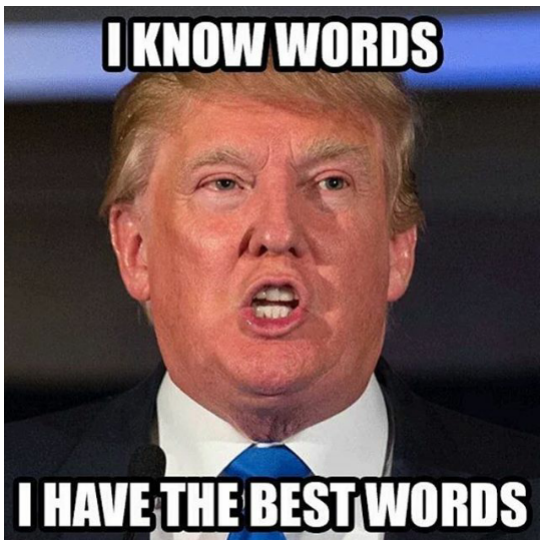
  Donald Trump

What's Missing from WordNet?



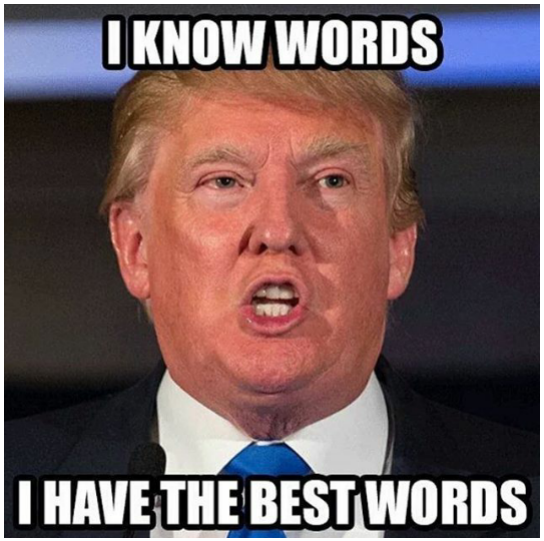
- Many named entities
  Donald Trump
- Recent terminology
  social network, *selfie*

What's Missing from WordNet?



- Many named entities
  Donald Trump
- Recent terminology
  social network, *selfie*
- Relations between existing synsets
  hyper:*(Louisa May Alcott, woman)*

What's Missing from WordNet?



- Many named entities
  Donald Trump
- Recent terminology
  social network, *selfie*
- Relations between existing synsets
  hyper:*(Louisa May Alcott, woman)*

This information can be completed using corpus statistics.

Usage of Word Embeddings

Using Word Embeddings

First, let's get this off the table: ***"why not just use word embeddings?"***

Using Word Embeddings

First, let's get this off the table: ***"why not just use word embeddings?"***

- Word embeddings are great in capturing semantic relatedness!

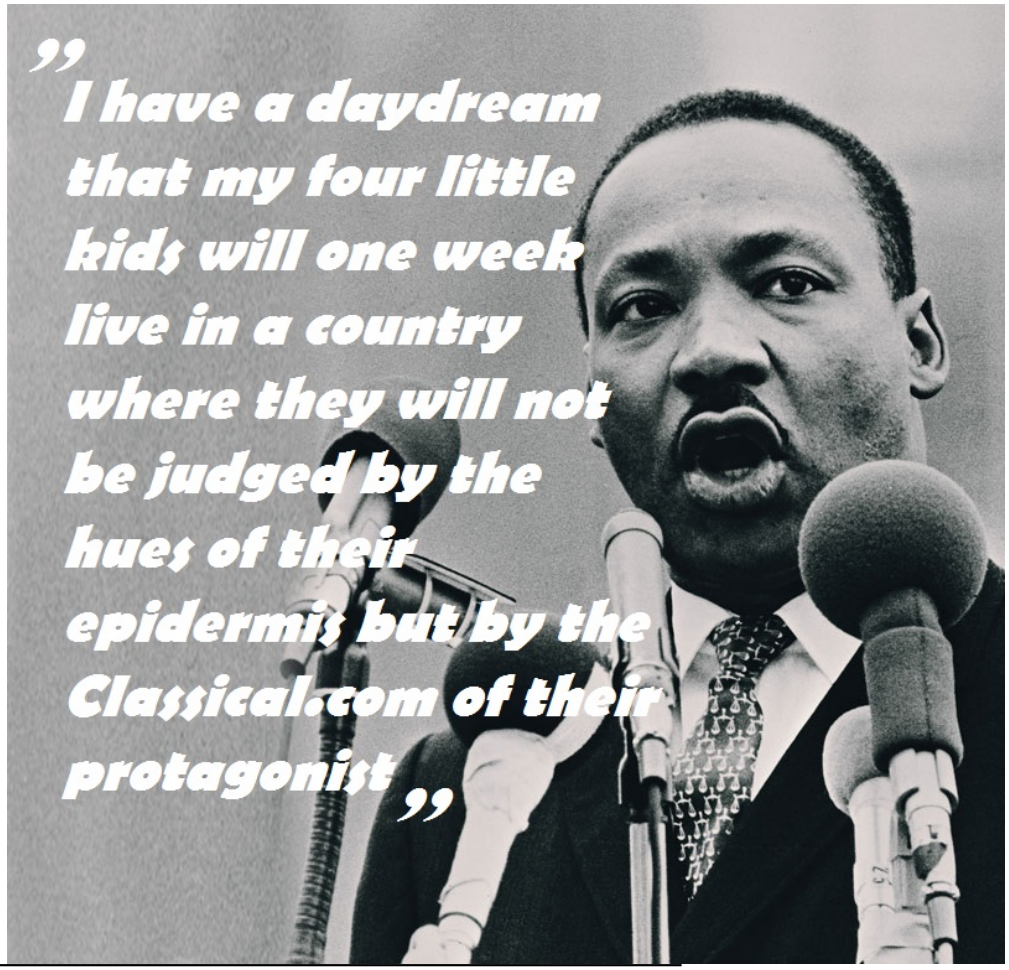
Using Word Embeddings

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.

Using Word Embeddings

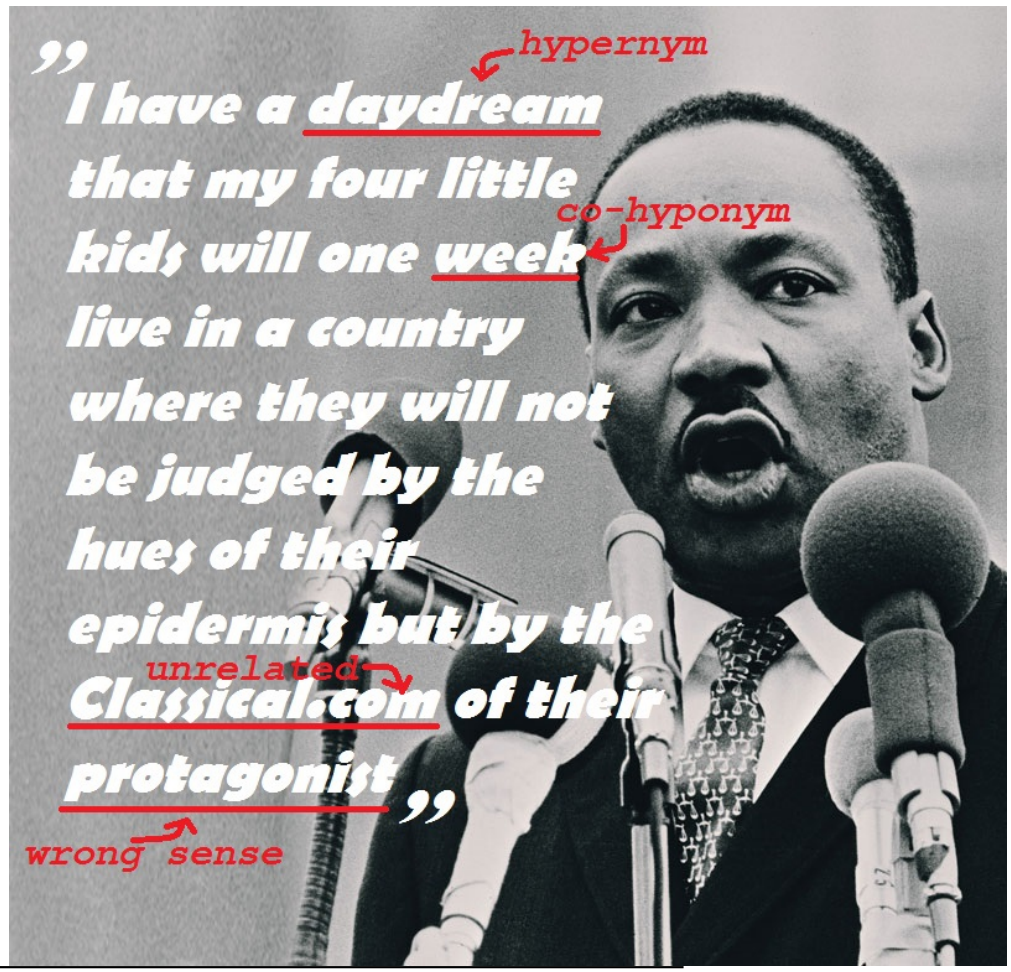
- To illustrate, take famous texts and replace nouns with their word2vec neighbours:[1]



[1] More examples here: `https://goo.gl/LJHzbi`

Using Word Embeddings

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



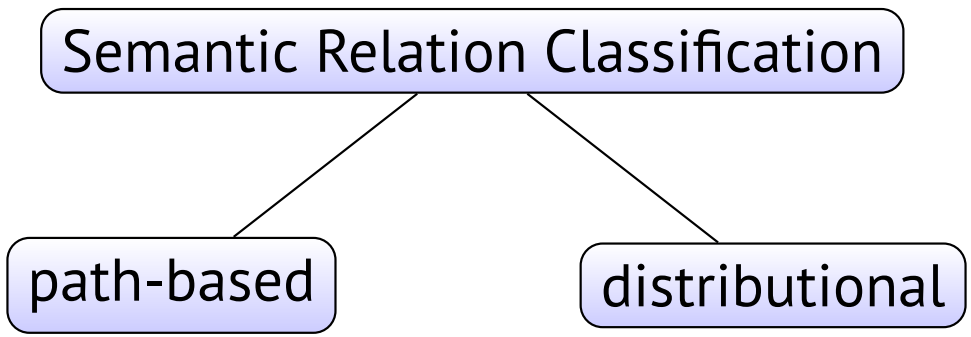
[1] More examples here: https://goo.gl/LJHzbi

Word Embeddings to Enhance WordNet:
Corpus-based Learning of Semantic Relations

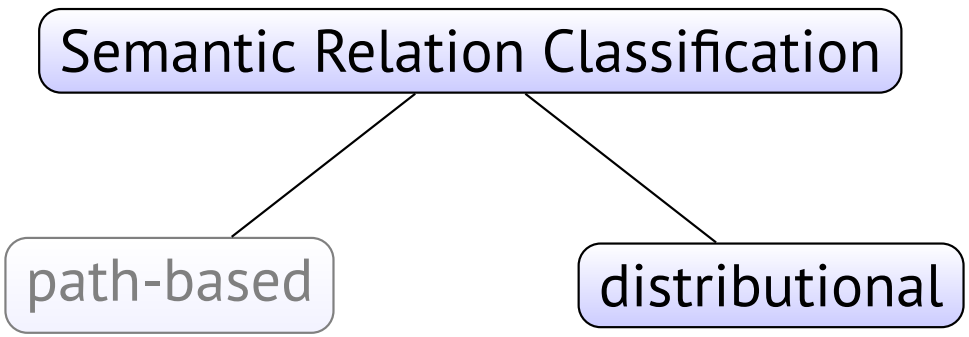
Lexical Semantic Relation Classification

- Given two words, *x* and *y*, decide what is the semantic relation that holds between them (if any)
  - usually works at the word-level, not the synset-level
  - e.g. both *fruit* and *company* are hypernyms of *apple*

Corpus-based Semantic Relation Classification



Distributional Approach



Distributional Approach

- Recognize the relation between words based on their *separate* occurrences in the corpus

Distributional Approach

- Recognize the relation between words based on their *separate* occurrences in the corpus

- Distributional Hypothesis [Harris, 1954]:
  Words that occur in similar contexts tend to have similar meanings

Distributional Approach

- Recognize the relation between words based on their *separate* occurrences in the corpus

- Distributional Hypothesis [Harris, 1954]:
  Words that occur in similar contexts tend to have similar meanings
  - e.g. *elevator* and *lift* will both appear next to *up*, *floor* and *stairs*

Distributional Approach

- Recognize the relation between words based on their *separate* occurrences in the corpus
- Distributional Hypothesis [Harris, 1954]:
  Words that occur in similar contexts tend to have similar meanings
  - e.g. *elevator* and *lift* will both appear next to *up*, *floor* and *stairs*
- Word embeddings [Mikolov et al., 2013, Pennington et al., 2014] are low-dimensional vector representations of words
  - Learned from distributional information
  - Similar words have similar vectors

Supervised Distributional Methods

- Represent (x, y) as a feature vector, based of the word embeddings:
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]

Supervised Distributional Methods

- Represent (x, y) as a feature vector, based of the word 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 the semantic relation between x and y

Supervised Distributional Methods

- Represent (x, y) as a feature vector, based of the word 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 the semantic relation between x and y

- Achieved very good results on various common datasets

Supervised Distributional Methods

- Represent (x, y) as a feature vector, based of the word 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 the semantic relation between x and y

- Achieved very good results on various common datasets
- Is it a solved task?

Supervised Distributional Methods

- Semantic Relation Classification is not solved.

Supervised Distributional Methods

- Semantic Relation Classification is not solved.
- Supervised distributional methods learn **separate** properties of either *x* or *y*

Supervised Distributional Methods

- Semantic Relation Classification is not solved.
- Supervised distributional methods learn **separate** properties of either *x* or *y*
- [Levy et al., 2015]: *"lexical memorization"*: overfitting to the most common relation of a specific word
  - Training: *(cat, animal)*, *(dog, animal)*, *(cow, animal)*, ... all labeled as hypernymy
  - Model: *(x, animal)* is a hypernym pair, regardless of *x*

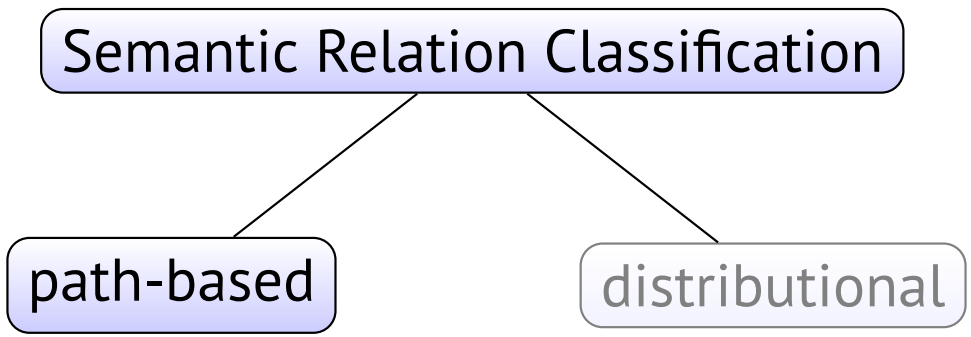
Supervised Distributional Methods

- Semantic Relation Classification is not solved.
- Supervised distributional methods learn **separate** properties of either *x* or *y*
- [Levy et al., 2015]: *"lexical memorization"*: overfitting to the most common relation of a specific word
  - Training: *(cat, animal)*, *(dog, animal)*, *(cow, animal)*, ... all labeled as hypernymy
  - Model: *(x, animal)* is a hypernym pair, regardless of *x*
- [Roller and Erk, 2016]: methods do more than memorize, learn about *x* and *y*'s roles

Supervised Distributional Methods

- Semantic Relation Classification is not solved.
- Supervised distributional methods learn **separate** properties of either *x* or *y*
- [Levy et al., 2015]: *"lexical memorization"*: overfitting to the most common relation of a specific word
  - Training: *(cat, animal)*, *(dog, animal)*, *(cow, animal)*, ... all labeled as hypernymy
  - Model: *(x, animal)* is a hypernym pair, regardless of *x*
- [Roller and Erk, 2016]: methods do more than memorize, learn about *x* and *y*'s roles
- [Shwartz et al., 2016]: methods provide only the **prior** of *x* or *y* to fit each relation

Path-based Approach



Path-based Approach

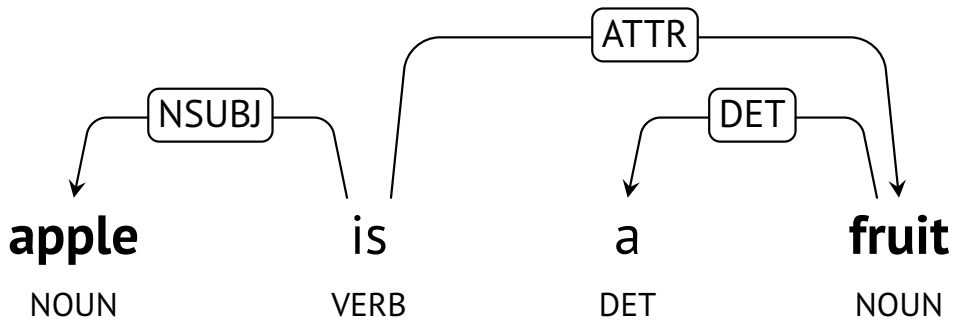
- Recognize the relation between *x* and *y* based on their *joint* occurrences in the corpus

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:

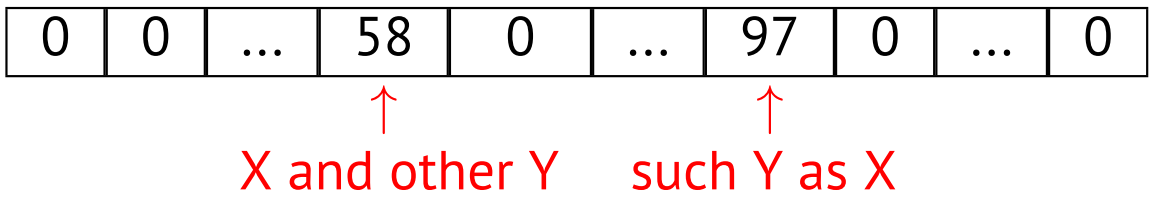


Supervised Path-based Approach

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

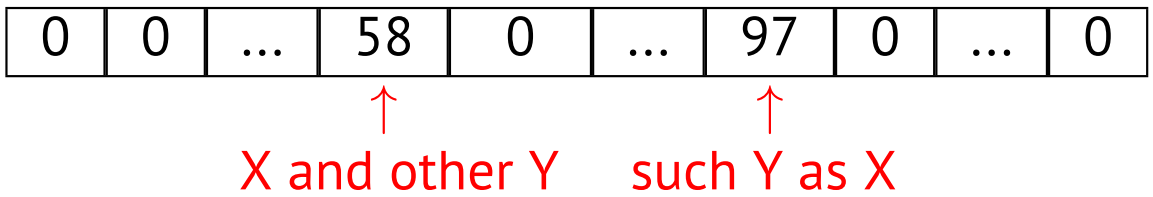
Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Features: all dependency paths connecting *x* and *y* in a corpus:



Supervised Path-based Approach

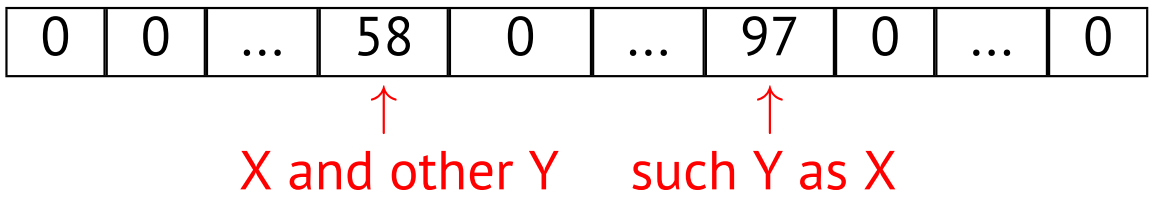
- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Features: all dependency paths connecting *x* and *y* in a corpus:



  - Trained a logistic regression classifier to predict hypernymy

Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Features: all dependency paths connecting *x* and *y* in a corpus:



  - Trained a logistic regression classifier to predict hypernymy
  - Learns Hearst patterns and other paths, improved performance

Path-based Approach Issues

- Similar paths share no information:

Path-based Approach Issues

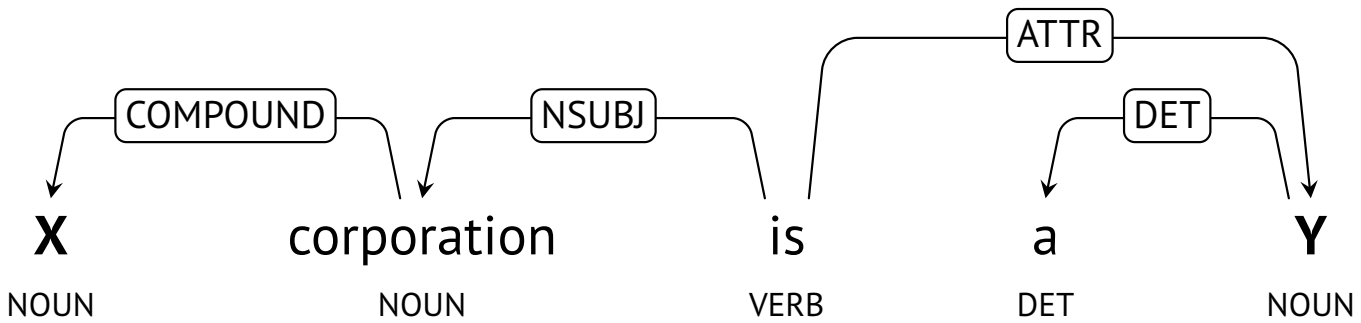
- Similar paths share no information:
  - X inc. is a Y
  - X group is a Y
  - X organization is a Y

Path-based Approach Issues

- Similar paths share no information:

  X inc. is a Y
  X group is a Y
  X organization is a Y

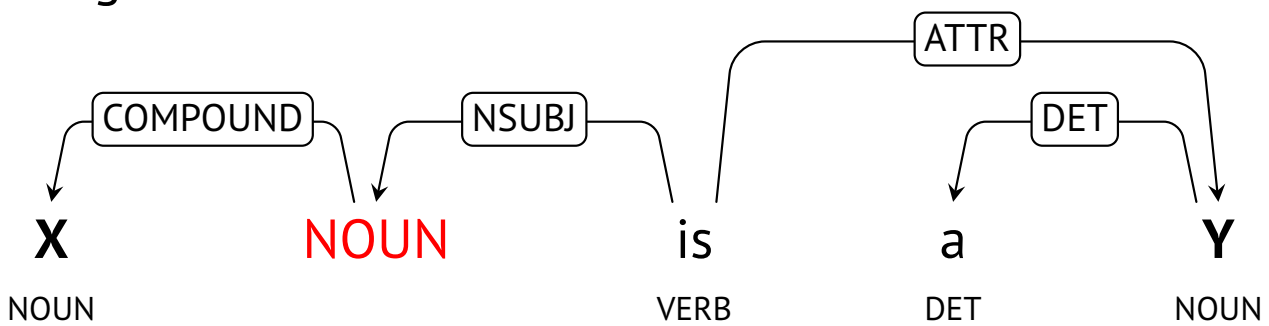
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:



Path-based Approach Issues

- Similar paths share no information:
  - X inc. is a Y
  - X group is a Y
  - X organization is a Y

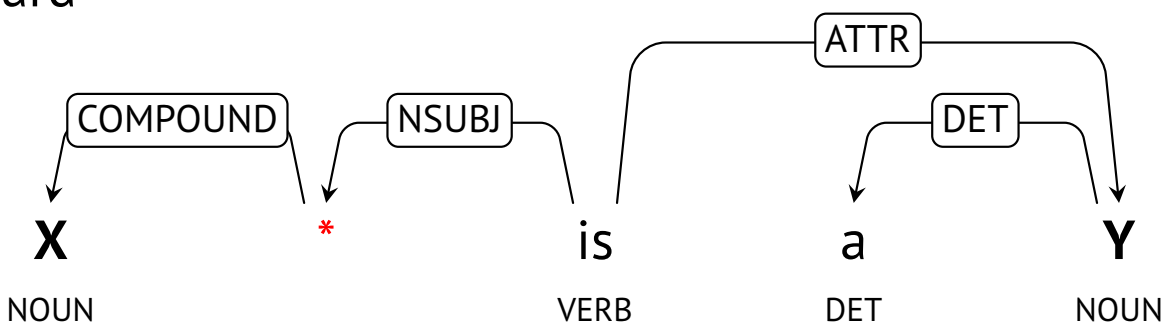
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - its POS tag



Path-based Approach Issues

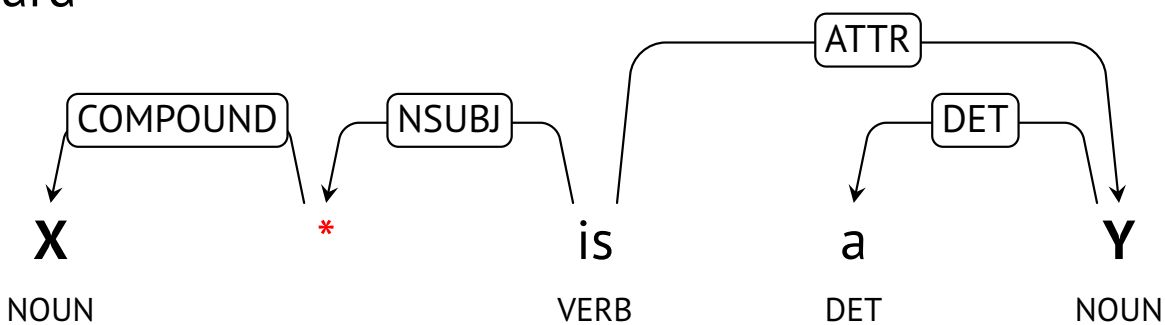
- Similar paths share no information:
  - X inc. is a Y
  - X group is a Y
  - X organization is a Y

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



Path-based Approach Issues

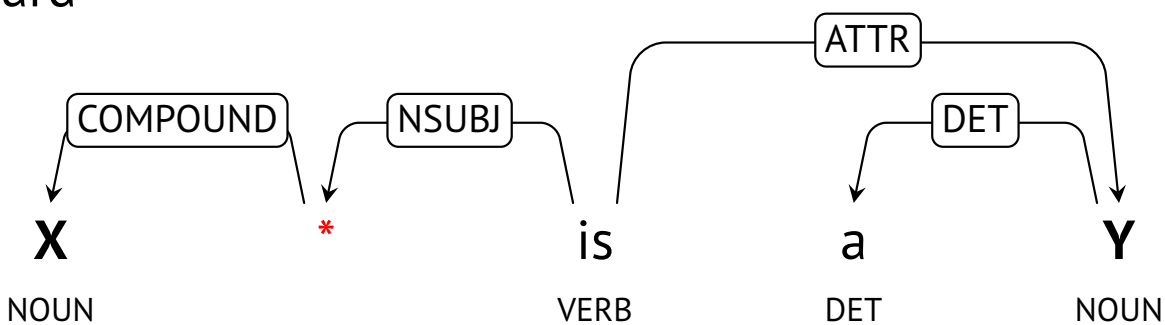
- Similar paths share no information:
  - X inc. is a Y
  - X group is a Y
  - X organization is a Y
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - a wild-card



- Some of these generalizations are too general:
  - X is defined as Y ≈ X is described as Y via X is VERB as Y

Path-based Approach Issues

- 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 ≈ X is described as Y via X is VERB as Y
  - X is defined as Y ≠ X is rejected as Y

Integrated Path-based and Distributional Method [Shwartz et al., 2016]

Integrating Path-based and Distributional Information

- Path-based and distributional sources are considered complementary

Integrating Path-based and Distributional Information

- Path-based and distributional sources are considered complementary
- Recently, distributional methods outpeformed path-based ones
  - No performance gain from adding path-based information

Integrating Path-based and Distributional Information

- Path-based and distributional sources are considered complementary
- Recently, distributional methods outpeformed path-based ones
  - No performance gain from adding path-based information
- Is path-based information redundant given distributional information?

The Hypernymy Detection Task

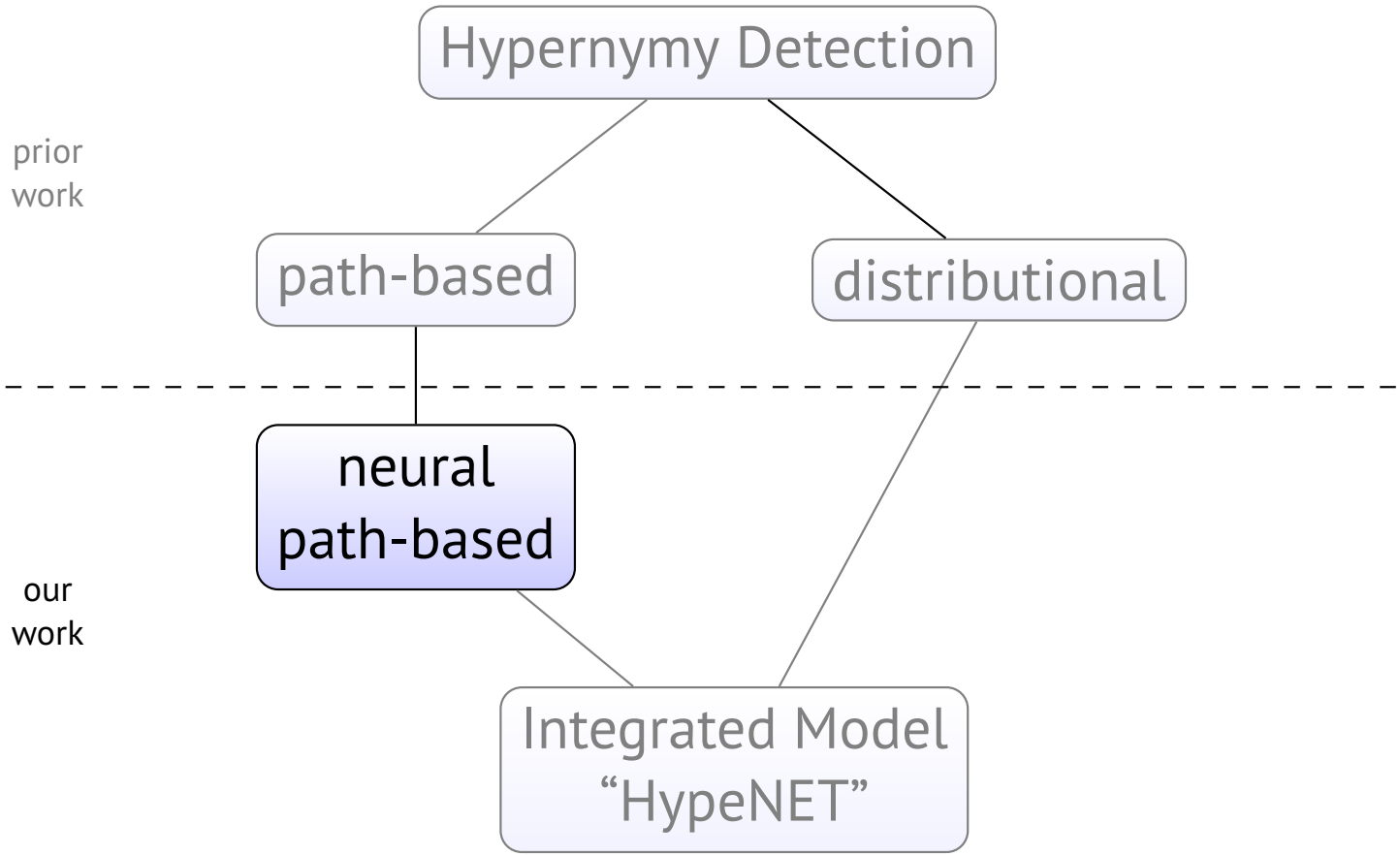
- We first focused on hypernymy
  - The hyponym is a subclass of / instance of the hypernym
  - *(cat, animal)*, *(Google, company)*

The Hypernymy Detection Task

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

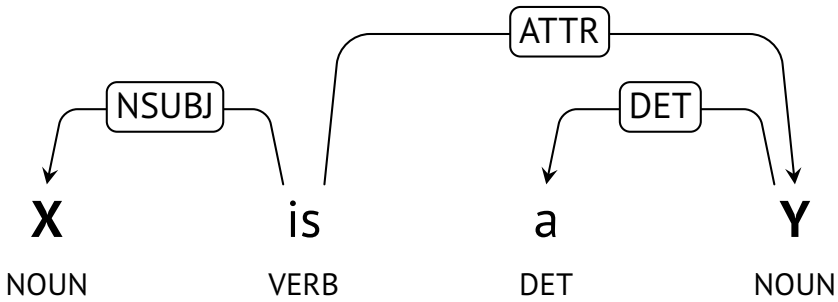
- Given two words, *x* and *y*, decide whether *y* is a hypernym of *x*
  - works at the word-level, not the synset-level

First Step: Improving Path Representation



Path Representation (1/2)

1. Split each path to edges



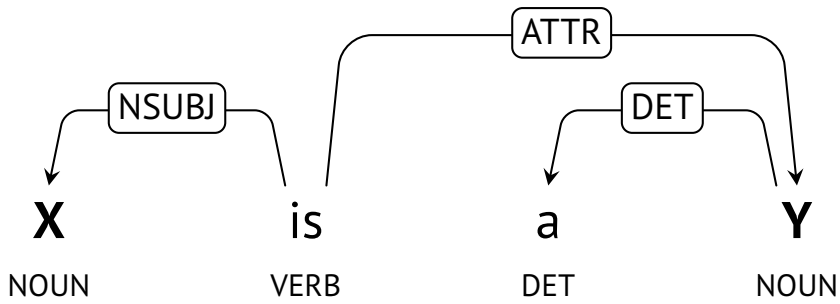
`X/NOUN/nsubj/>` `be/VERB/ROOT/-` `Y/NOUN/attr/<`

- Each edge consists of 4 components:

`dependent lemma` / `dependent POS` / `dependency label` / `direction`

Path Representation (1/2)

1. Split each path to edges



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

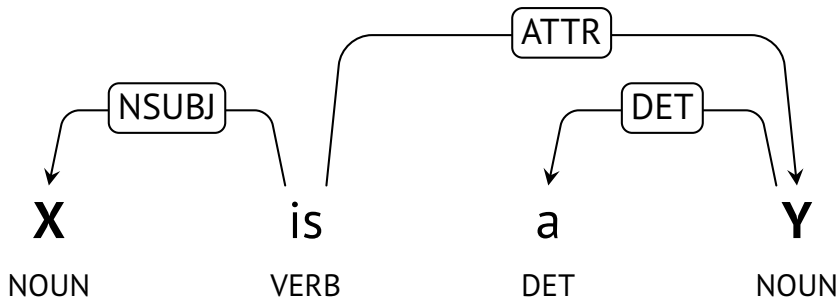
- Each edge consists of 4 components:

`dependent lemma` / `dependent POS` / `dependency label` / `direction`

- We learn embedding vectors for each component
  - Lemma: initialized with pre-trained word embeddings

Path Representation (1/2)

1. Split each path to edges



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

- Each edge consists of 4 components:

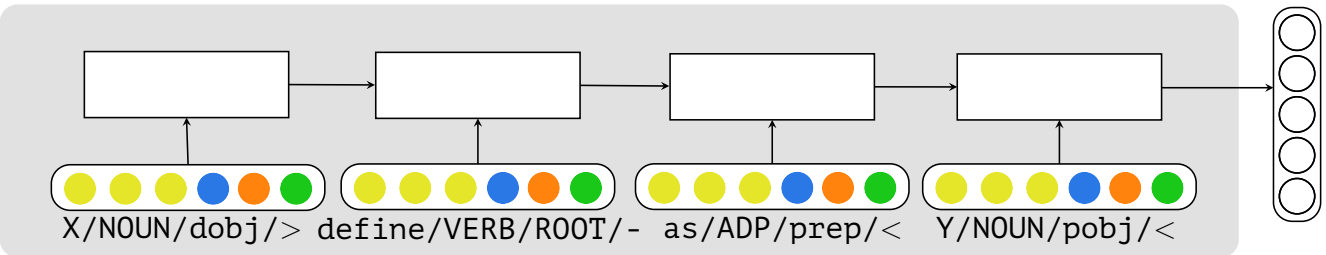
`dependent lemma` / `dependent POS` / `dependency label` / `direction`

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

`be/VERB/ROOT/-`

Path Representation (2/2)

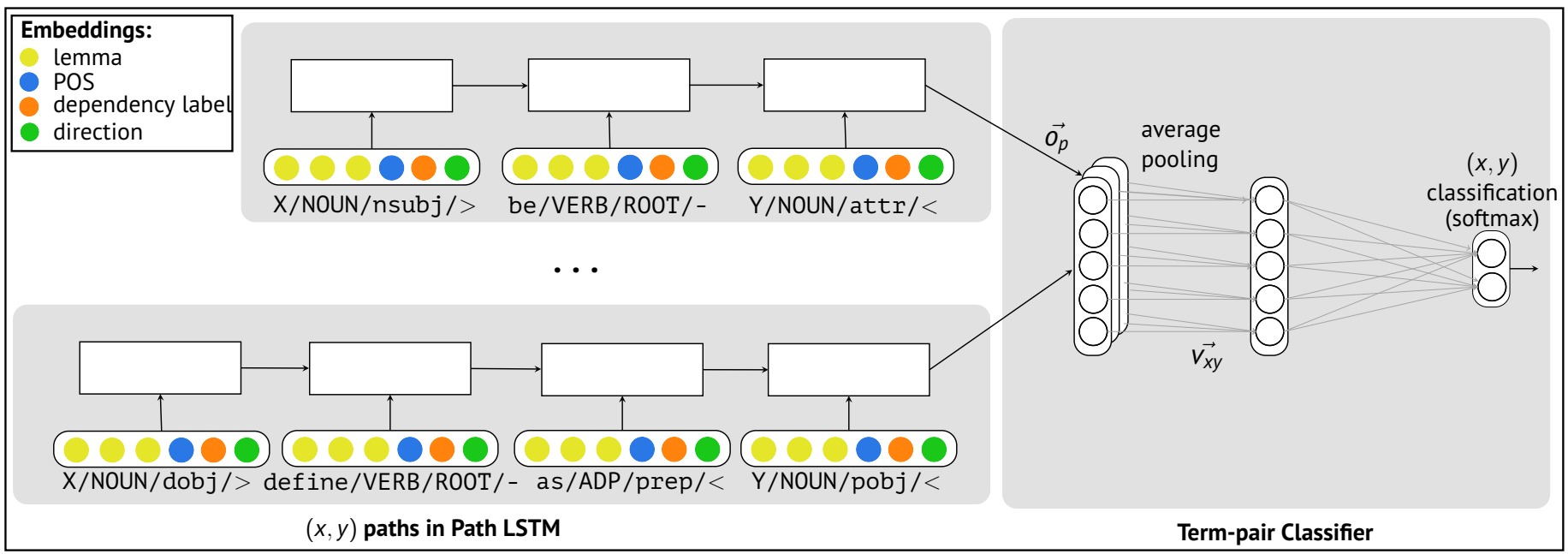
2. Feed the edges sequentially to an LSTM



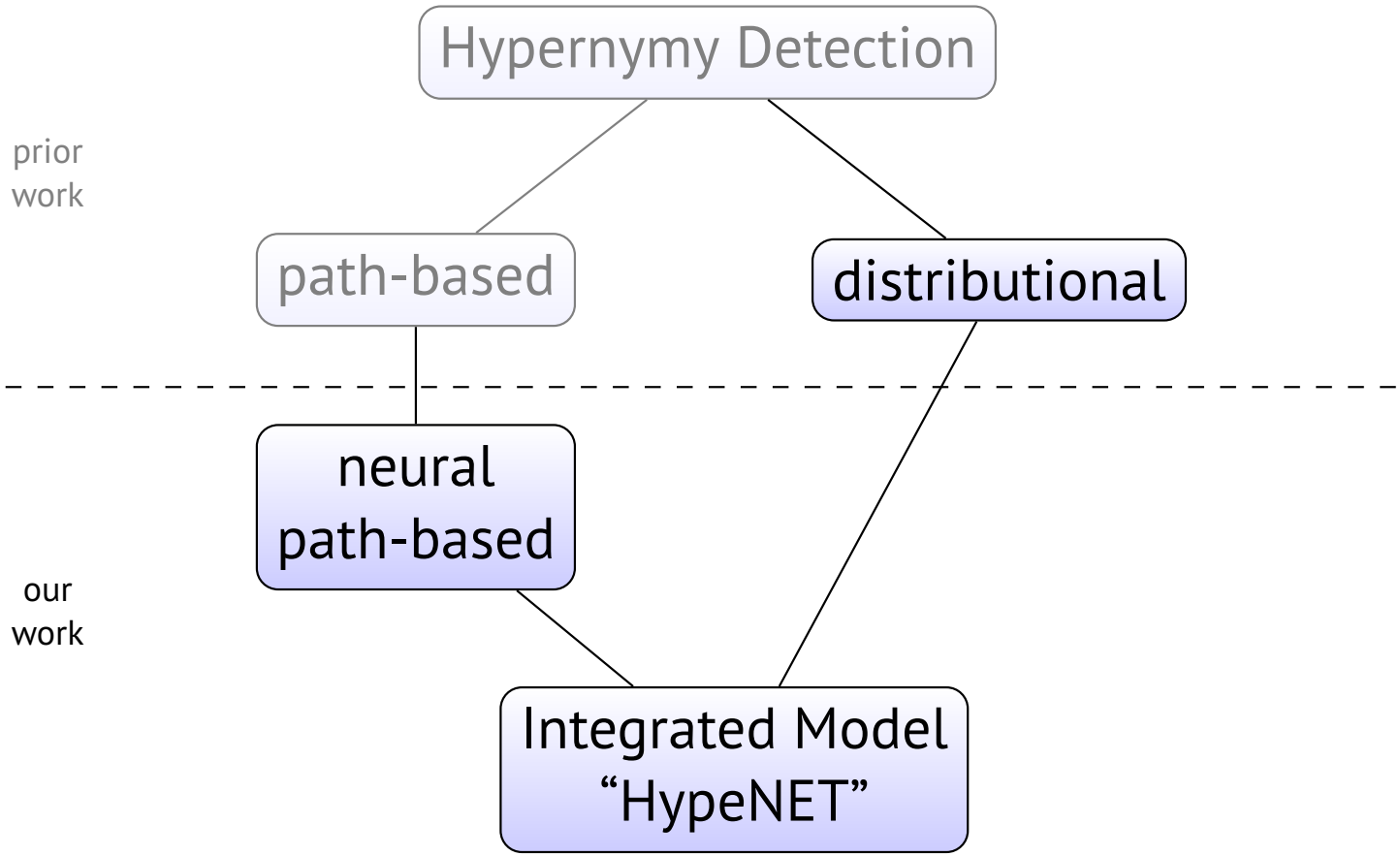
- Use the last output vector as the path embedding
- The LSTM may focus on edges that are more informative for the classification task, while ignoring others

Path-based Word-pair Classification

- The LSTM encodes a single path
- Each word-pair has multiple paths
  - Represent a word-pair as its averaged path embedding
- Classify for hypernymy (path-based network):



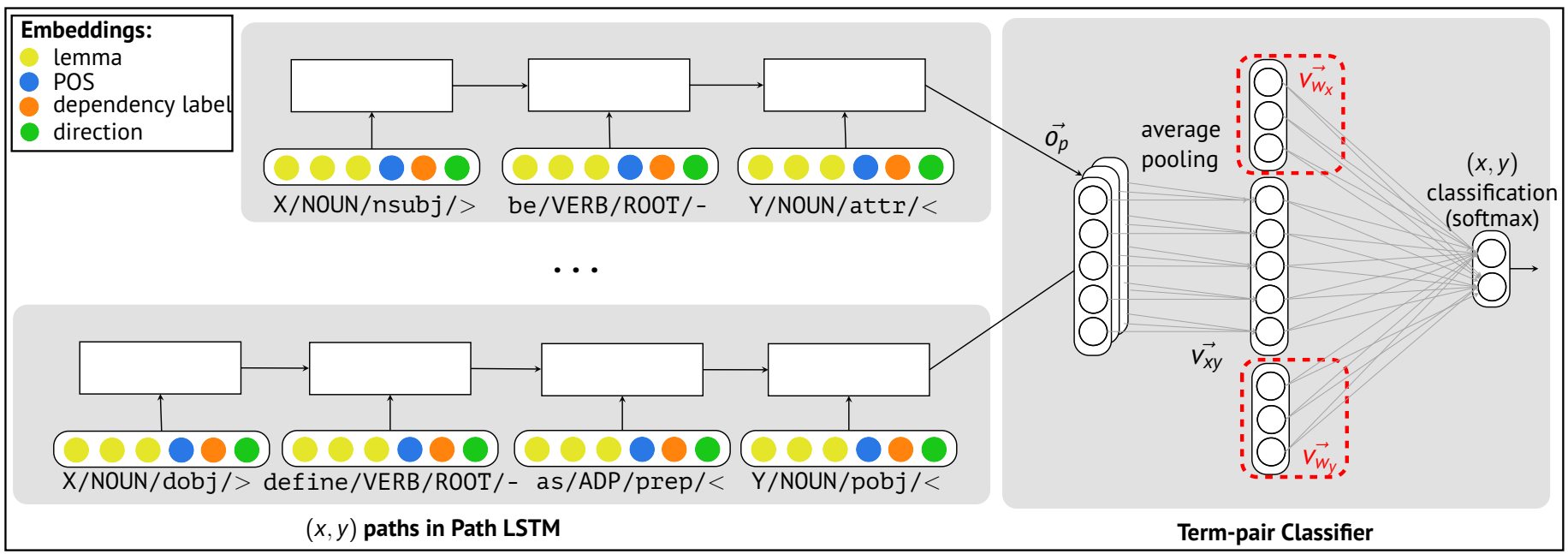
Second Step: Integrating Distributional Information



Second Step: Integrating Distributional Information

- Integrated network: add distributional information
  - Concatenate *x* and *y*'s word embeddings to the averaged path

- Classify for hypernymy (integrated network):



Results

- On a new dataset
  - built from WordNet, Wikidata, DBPedia, and Yago

method		precision	recall	F_1
Path-based	Snow	0.843	0.452	0.589
	Snow + GEN	0.852	0.561	0.676
	HypeNET Path-based	0.811	0.716	0.761
Distributional	Best Supervised	0.901	0.637	0.746
Integrated	HypeNET Integrated	**0.913**	**0.890**	**0.901**

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

Results

- On a new dataset
  - built from WordNet, Wikidata, DBPedia, and Yago

method		precision	recall	F_1
Path-based	Snow	0.843	0.452	0.589
	Snow + GEN	0.852	0.561	0.676
	HypeNET Path-based	0.811	0.716	0.761
Distributional	Best Supervised	0.901	0.637	0.746
Integrated	HypeNET Integrated	**0.913**	**0.890**	**0.901**

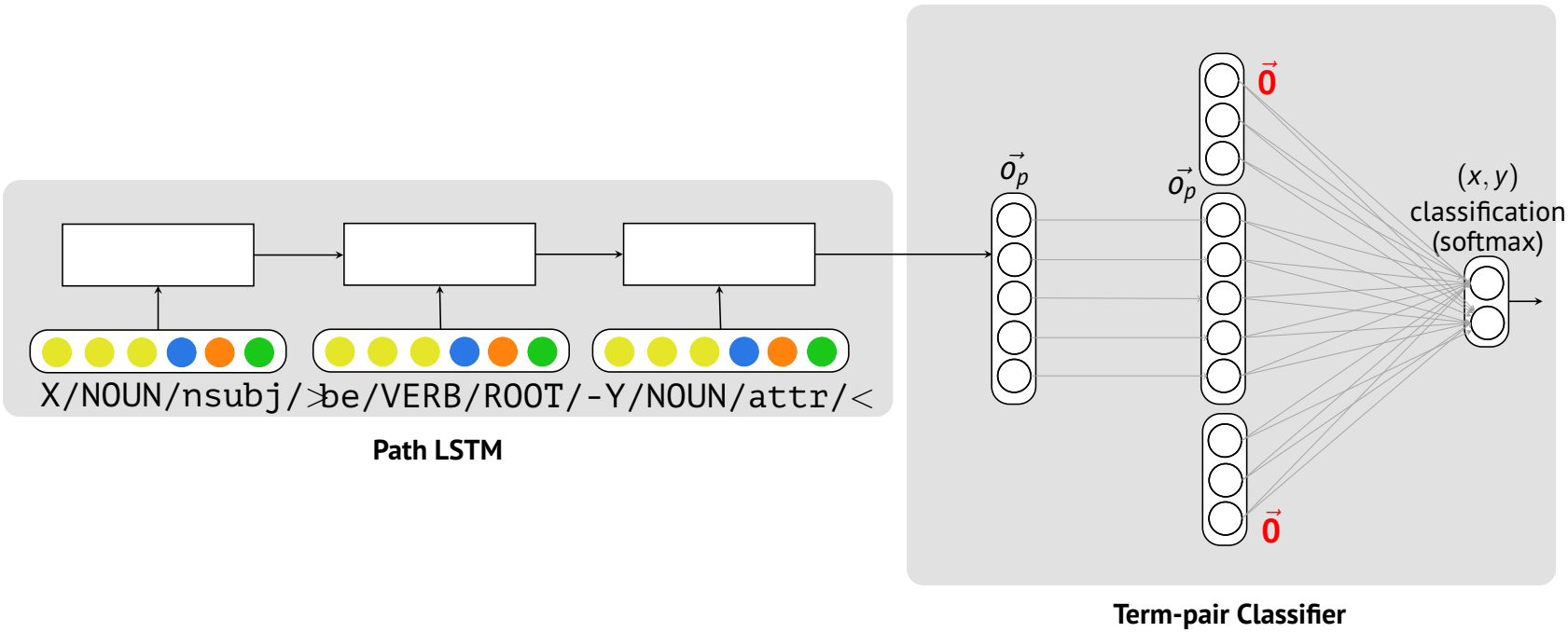
- The integrated method substantially outperforms both path-based and distributional methods

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 $softmax(W \cdot [\vec{0}, \vec{o_p}, \vec{0}])[1]$

Analysis - Path Representation (2/2)

- Snow's method finds certain common paths:

  X company is a Y
  X ltd is a Y

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 general, possibly noisy paths:

  X NOUN 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 general, possibly noisy paths:

  X NOUN is a Y

- HypeNET makes fine-grained generalizations:

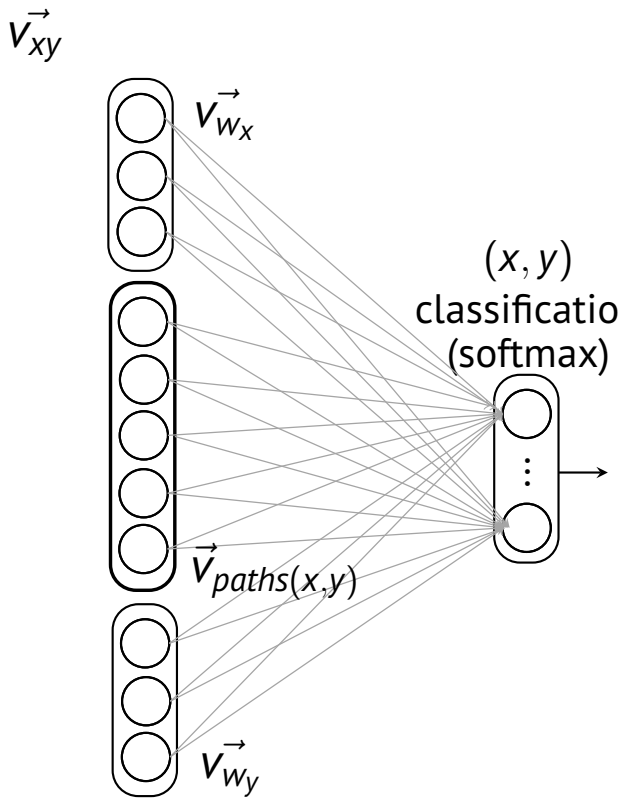
  X association is a Y
  X co. is a Y
  X company is a Y
  X corporation is a Y
  X foundation is a Y
  X group is a Y
  ...

Other Semantic Relations

LexNET - Multiple Semantic Relation Classification [Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations:

  hypernymy, meroynymy, co-hyponymy, event, attribute, synonymy, antonymy, random



Path-based Component Contribution

- LexNET outperforms individual path-based and distributional methods on all datasets

Path-based Component Contribution

- LexNET outperforms individual path-based and distributional methods on all datasets
- Path-based contribution over distributional info is small when lexical memorization is enabled

Path-based Component Contribution

- LexNET outperforms individual path-based and distributional methods on all datasets
- 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)*.

Path-based Component Contribution

- LexNET outperforms individual path-based and distributional methods on all datasets
- 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)*.

Path-based Component Contribution

- LexNET outperforms individual path-based and distributional methods on all datasets
- 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)*.

Path-based Component Contribution

- LexNET outperforms individual path-based and distributional methods on all datasets
- 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*

Which Relations can the Model Learn?

- All methods are bad in recognizing synonyms and antonyms.

Which Relations can the Model Learn?

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

Which Relations can the Model Learn?

- All methods 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*

Which Relations can the Model Learn?

- All methods 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"

Which Relations can the Model Learn?

- All methods 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.

Learning Antonyms

- [Rajana et al., 2017] integrated morphological cues to distinguish antonymy from other relations:
  - Added a “negated” feature

Learning Antonyms

- [Rajana et al., 2017] integrated morphological cues to distinguish antonymy from other relations:
  - Added a “negated” feature
  - List of negated prefixes: *de, un, in, anti, il, non, dis*

Learning Antonyms

- [Rajana et al., 2017] integrated morphological cues to distinguish antonymy from other relations:
  - Added a "negated" feature
  - List of negated prefixes: *de, un, in, anti, il, non, dis*
  - A word with a negated prefix is replaced with the non-negated form, and the "negated" feature is turned on

Learning Antonyms

- [Rajana et al., 2017] integrated morphological cues to distinguish antonymy from other relations:
  - Added a "negated" feature
  - List of negated prefixes: *de, un, in, anti, il, non, dis*
  - A word with a negated prefix is replaced with the non-negated form, and the "negated" feature is turned on
  - e.g. *unhappy* $\rightarrow$ neg + *happy*

Learning Antonyms

- [Rajana et al., 2017] integrated morphological cues to distinguish antonymy from other relations:
  - Added a "negated" feature
  - List of negated prefixes: *de, un, in, anti, il, non, dis*
  - A word with a negated prefix is replaced with the non-negated form, and the "negated" feature is turned on
  - e.g. *unhappy* $\rightarrow$ neg + *happy*

- Improved performance on both binary and multiclass tasks

Learning Synonyms

- Leverage the fact that synonyms do not occur together!

Learning Synonyms

- Leverage the fact that synonyms do not occur together!
- In the CogALex shared task [Shwartz and Dagan, 2016b], we added a heuristic to LexNET:
- If the classification score is similar for synonym and another relation, and *x* and *y* occur together less than 3 times in the corpus, classify as synonym

WordNet in Word Embeddings:
Encoding WordNet into Word Embeddings

Using WordNet in the World of Deep Learning

- So, word embeddings alone don't obviate the usage of WordNet

Using WordNet in the World of Deep Learning

- So, word embeddings alone don't obviate the usage of WordNet
- But what about DL-based applications?

Using WordNet in the World of Deep Learning

- So, word embeddings alone don't obviate the usage of WordNet
- But what about DL-based applications?
- Let's look at RTE as an example

Recognizing Textual Entailment (RTE)

- Given two sentences, a *premise* and a *hypothesis*, does the *premise* entail, contradict or is neutral to the *hypothesis*?

Recognizing Textual Entailment (RTE)

- Given two sentences, a *premise* and a *hypothesis*, does the *premise* entail, contradict or is neutral to the *hypothesis*?

Premise

A boy is hitting a baseball

Hypotheses

1. A **child** is hitting a baseball ⇒ ENTAILMENT
2. A boy is **missing** a baseball ⇒ CONTRADICTION
3. A **girl** is hitting a baseball ⇒ NEUTRAL

Recognizing Textual Entailment (RTE)

- Lexical semantic relations play an important role in the decision

Premise

A boy is hitting a baseball

Hypotheses

1. A **child** is hitting a baseball ⇒ ENTAILMENT: *hypernym:(boy, child)*
2. A boy is **missing** a baseball ⇒ CONTRADICTION: *antonym:(hitting, missing)*
3. A **girl** is hitting a baseball ⇒ NEUTRAL: *co-hyponym:(boy, girl)*

Lexical Entailment in RTE

- Traditional RTE systems: lexical entailment step
  - *child-boy*, *missing-hitting*, etc.
  - Often aided by WordNet

[2] https://explosion.ai/blog/deep-learning-formula-nlp

Lexical Entailment in RTE

- Traditional RTE systems: lexical entailment step
  - *child-boy*, *missing-hitting*, etc.
  - Often aided by WordNet

- Today's end-to-end neural models:
  - Four steps: embed, encode, attend, predict[2]

[2] https://explosion.ai/blog/deep-learning-formula-nlp

Lexical Entailment in RTE

- Traditional RTE systems: lexical entailment step
  - *child-boy*, *missing-hitting*, etc.
  - Often aided by WordNet

- Today's end-to-end neural models:
  - Four steps: embed, encode, attend, predict[2]
  - Embed: external lexical knowledge through pre-trained word embeddings

[2] https://explosion.ai/blog/deep-learning-formula-nlp

Lexical Entailment in RTE

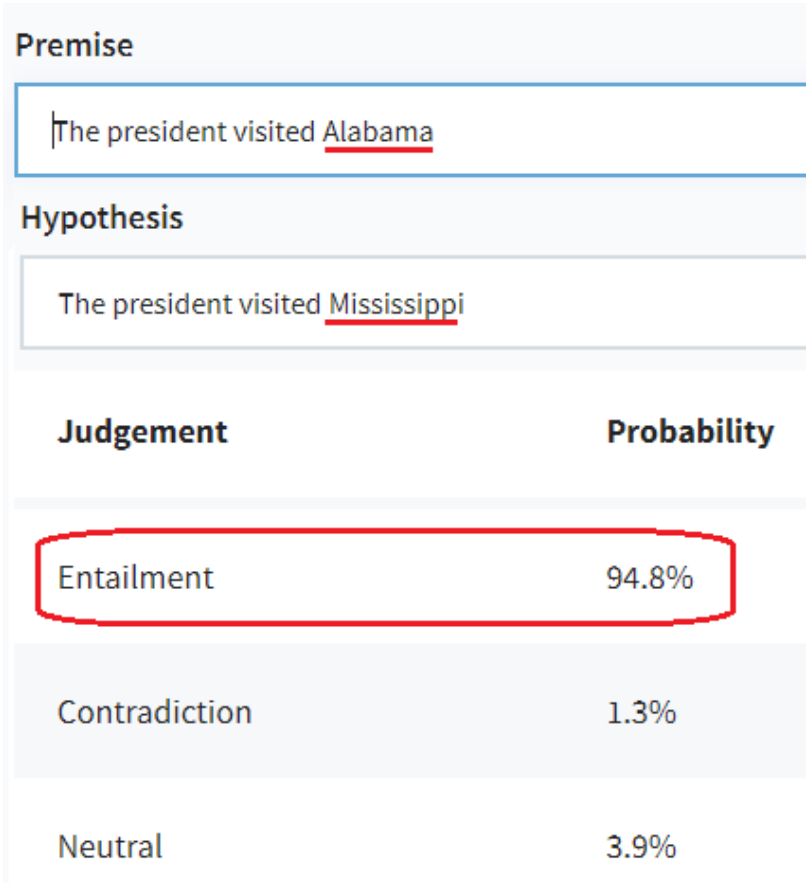
- Traditional RTE systems: lexical entailment step
  - *child-boy*, *missing-hitting*, etc.
  - Often aided by WordNet

- Today's end-to-end neural models:
  - Four steps: embed, encode, attend, predict[2]
  - Embed: external lexical knowledge through pre-trained word embeddings
  - Attend/predict: may implicitly learn semantic relations between words

[2] https://explosion.ai/blog/deep-learning-formula-nlp

Lexical Entailment in Neural RTE

- ...Easily confused by similarity:



from `http://demo.allennlp.org/textual-entailment`

Injecting Lexical Knowledge to Neural Inference Models

- Explicit information from WordNet can improve performance and/or training time

Injecting Lexical Knowledge to Neural Inference Models

- Explicit information from WordNet can improve performance and/or training time
- How can this knowledge be injected to neural models?

WordNet Embeddings

- We already have WordNet, why do we need to put into vectors?

WordNet Embeddings

- We already have WordNet, why do we need to put into vectors?
  - Suitable as a plug & play component for DL models

WordNet Embeddings

- We already have WordNet, why do we need to put into vectors?
  - Suitable as a plug & play component for DL models
  - Generic solution for lexical knowledge across systems

WordNet Embeddings

- We already have WordNet, why do we need to put into vectors?
  - Suitable as a plug & play component for DL models
  - Generic solution for lexical knowledge across systems
  - Just replace pre-trained word embeddings with WordNet embeddings

WordNet Embeddings

- Many methods to encode WordNet into vectors:

WordNet Embeddings

- Many methods to encode WordNet into vectors:
  - Retrofitting [Faruqui et al., 2015]
  - Order Embeddings [Vendrov et al., 2015]
  - Poincaré Embeddings [Nickel and Kiela, 2017]
  - LEAR [Vulić and Mrkšić, 2017]
  - and more...

WordNet Embeddings

- Many methods to encode WordNet into vectors:
  - Retrofitting [Faruqui et al., 2015]
  - Order Embeddings [Vendrov et al., 2015]
  - Poincaré Embeddings [Nickel and Kiela, 2017]
  - LEAR [Vulić and Mrkšić, 2017]
  - and more...
- Can these embeddings just be plugged into neural models?

Using WordNet Embeddings in Neural Inference Models

- Not so easily.

Using WordNet Embeddings in Neural Inference Models

- Not so easily.
- No extrinsic evaluation
  - Evaluated on their ability to complete missing edges from WordNet

Using WordNet Embeddings in Neural Inference Models

- Not so easily.
- No extrinsic evaluation
  - Evaluated on their ability to complete missing edges from WordNet
- Typically define a new hypernymy/entailment measure
  - E.g. based on vector norms

Using WordNet Embeddings in Neural Inference Models

- Not so easily.
- No extrinsic evaluation
  - Evaluated on their ability to complete missing edges from WordNet
- Typically define a new hypernymy/entailment measure
  - E.g. based on vector norms

- Either don't preserve distance/ don't preserve directionality

Injecting Lexical Knowledge to Neural Inference Models

- Research question: what is the best way to encode WordNet knowledge?
  - Easily pluggable

Injecting Lexical Knowledge to Neural Inference Models

- Research question: what is the best way to encode WordNet knowledge?
  - Easily pluggable
  - Generic solution for various models

Injecting Lexical Knowledge to Neural Inference Models

- Research question: what is the best way to encode WordNet knowledge?
  - Easily pluggable
  - Generic solution for various models
  - Show actual improvement in end-tasks

Recap

- Discussed the complementary nature of WordNet & word embeddings:

Recap

- Discussed the complementary nature of WordNet & word embeddings:
  - WordNet: specific semantic relations, limited coverage

Recap

- Discussed the complementary nature of WordNet & word embeddings:
  - WordNet: specific semantic relations, limited coverage
  - Word embeddings: general relatedness, broad coverage

Recap

- Discussed the complementary nature of WordNet & word embeddings:
  - WordNet: specific semantic relations, limited coverage
  - Word embeddings: general relatedness, broad coverage

- Showed a method for corpus-based learning of semantic relations:
  - Word embeddings $\rightarrow$ WordNet

Recap

- Discussed the complementary nature of WordNet & word embeddings:
  - WordNet: specific semantic relations, limited coverage
  - Word embeddings: general relatedness, broad coverage
- Showed a method for corpus-based learning of semantic relations:
  - Word embeddings $\rightarrow$ WordNet
- Next step:
  - WordNet $\rightarrow$ Word embeddings ($\rightarrow$ applications)

Recap

- Discussed the complementary nature of WordNet & word embeddings:
  - WordNet: specific semantic relations, limited coverage
  - Word embeddings: general relatedness, broad coverage
- Showed a method for corpus-based learning of semantic relations:
  - Word embeddings $\rightarrow$ WordNet
- Next step:
  - WordNet $\rightarrow$ Word embeddings ($\rightarrow$ applications)

Thank you!

References I

[Baroni et al., 2012] Baroni, M., Bernardi, R., Do, N.-Q., and Shan, C.-c. (2012). Entailment above the word level in distributional semantics. In *EACL*, pages 23–32.

[Faruqui et al., 2015] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. *NAACL*.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *ACL*, pages 539–545.

[Levy et al., 2015] Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations. *NAACL*.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

[Nakashole et al., 2012] Nakashole, N., Weikum, G., and Suchanek, F. (2012). Patty: a taxonomy of relational patterns with semantic types. In *EMNLP and CoNLL*, pages 1135–1145.

[Nguyen et al., 2017] Nguyen, K. A., Schulte im Walde, S., and Vu, N. T. (2017). Distinguishing antonyms and synonyms in a pattern-based neural network. In *EACL*.

[Nickel and Kiela, 2017] Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *Automated Knowledge Base Completion workshop*.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

[Rajana et al., 2017] Rajana, S., Callison-Burch, C., Apidianaki, M., and Shwartz, V. (2017). Learning antonyms with paraphrases and a morphology-aware neural network. In **SEM*, pages 12–21.

[Roller and Erk, 2016] Roller, S. and Erk, K. (2016). Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of EMNLP 2016*.

References II

[Roller et al., 2014] Roller, S., Erk, K., and Boleda, G. (2014). Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING*, pages 1025–1036.

[Shwartz and Dagan, 2016a] Shwartz, V. and Dagan, I. (2016a). path-based vs. distributional information in recognizing lexical semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V), in COLING*, Osaka, Japan.

[Shwartz and Dagan, 2016b] Shwartz, V. and Dagan, I. (2016b). cogalex-v shared task: Lexnet - integrated path-based and distributional method for the identification of semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V), in COLING*, Osaka, Japan.

[Shwartz et al., 2016] Shwartz, V., Goldberg, Y., and Dagan, I. (2016). Improving hypernymy detection with an integrated path-based and distributional method. In *ACL*, pages 2389–2398.

[Snow et al., 2004] Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning syntactic patterns for automatic hypernym discovery. In *NIPS*.

[Vendrov et al., 2015] Vendrov, I., Kiros, R., Fidler, S., and Urtasun, R. (2015). Order-embeddings of images and language. *ICLR*.

[Vulić and Mrkšić, 2017] Vulić, I. and Mrkšić, N. (2017). Specialising word vectors for lexical entailment. *arXiv preprint arXiv:1710.06371*.

[Weeds et al., 2014] Weeds, J., Clarke, D., Reffin, J., Weir, D., and Keller, B. (2014). Learning to distinguish hypernyms and co-hyponyms. In *COLING*, pages 2249–2259.