Hypernymy detection with HypeNET: Leveraging Path-Based and Distributional Signals

Vered Shwartz, Yoav Goldberg and Ido Dagan

Bar-Ilan University

ICRI-CI 2016 Retreat, May 25, 2016

Hypernymy

- A semantic relation between two terms (x, y)
  - the *hyponym* (x) is a more specific type or an instance of the *hypernym* (y)
  - e.g. *(pineapple, fruit), (green, color), (Obama, president)*

The Hypernymy Detection Task

- Given two terms, x and y, decide whether y is a hypernym of x
  - in some senses of x and y

Example Motivation - Question Answering

Question

"What animals inhabit the Arctic regions?"

Candidate Passages

1. Polar bears inhabit the Arctic regions.
2. Indigenous people inhabit the Arctic regions.

Knowledge

(bear, animal) is a hyponym-hypernym pair, but *(people, animal)* is not.

Outline

1. Prior Methods
   - Distributional Approach
   - Path-based Approach
2. Integrated Path-based and Distributional Method
   - Adapting state-of-the-art deep learning techniques
3. Evaluation

Corpus-based Methods for Hypernymy Detection

- Consider the statistics of term occurrences in a large corpus

Corpus-based Methods for Hypernymy Detection

- Consider the statistics of term occurrences in a large corpus
- Roughly divided to two sub-approaches:
  - Distributional approach
  - Path-based approach

Distributional Approach

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

Distributional Approach

- 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

- 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*

- **Unsupervised hypernymy detection**: use the distributional vectors of x and y to decide whether y is a hypernym of x
  - Measure the distance between the vectors (e.g. cosine similarity)

Distributional Approach

- 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*

- **Unsupervised hypernymy detection**: use the distributional vectors of x and y to decide whether y is a hypernym of x
  - Measure the distance between the vectors (e.g. cosine similarity)

    Word similarity $\neq$ Hypernymy!
    e.g. co-hyponyms are similar (*(football, basketball), (cat, dog)*)

Distributional Approach

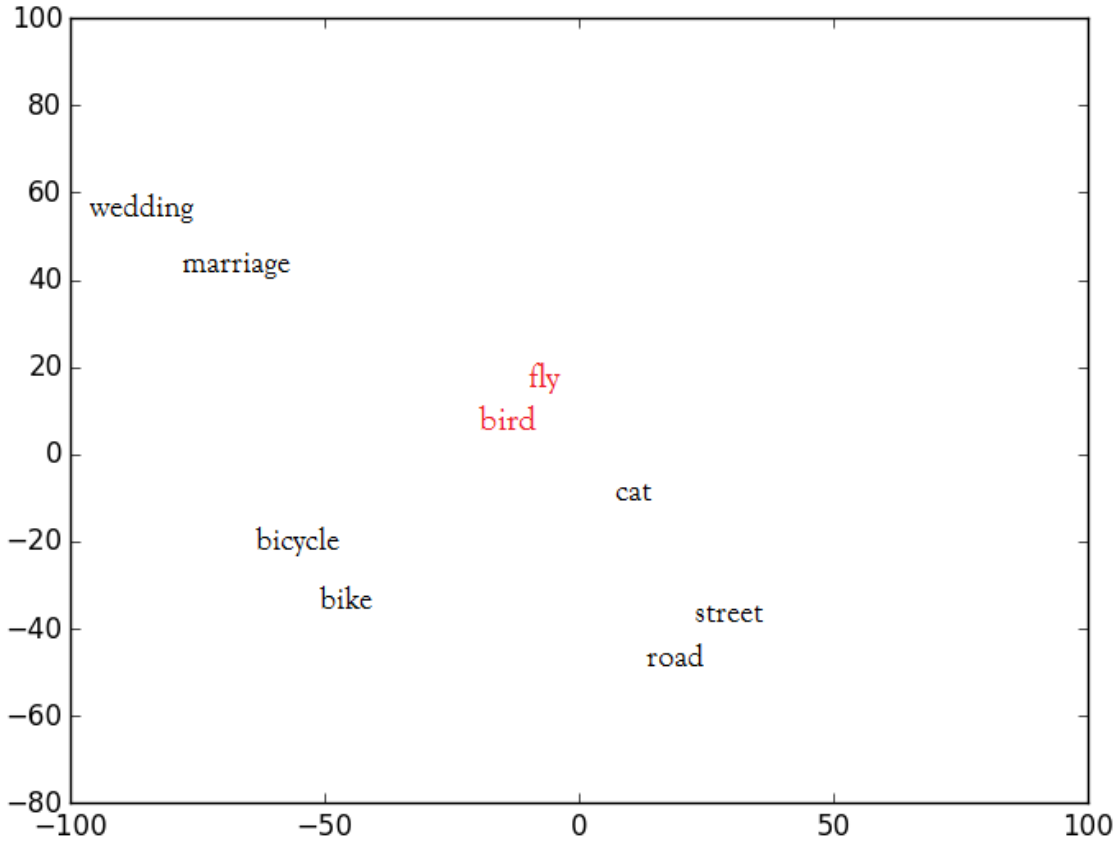
- Distributional Hypothesis [Harris, 1954]:
  Words that occur in similar contexts tend to have similar meanings
  - e.g. *elevator* and *lift* will both appear next to *up*, *floor* and *stairs*

- **Unsupervised hypernymy detection**: use the distributional vectors of x and y to decide whether y is a hypernym of x
  - Measure the distance between the vectors (e.g. cosine similarity)

    Word similarity $\neq$ Hypernymy!
    e.g. co-hyponyms are similar (*(football, basketball), (cat, dog)*)
  - Directional similarity measures [Weeds and Weir, 2003, Kotlerman et al., 2010, Santus et al., 2014, Rimell, 2014]

Distributional Approach - Word Embeddings

- Word embeddings [Mikolov et al., 2013, Pennington et al., 2014] are dense, low-dimensional vector representations of words
  - Created based on distributional data



- Similar words are close to each other in the vector space

Supervised Distributional Approach

- **Supervised hypernymy detection**: (x, y) is represented as a feature vector, based of the terms' embeddings vectors
  - 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 Approach

- **Supervised hypernymy detection**: (x, y) is represented as a feature vector, based of the terms' embeddings vectors
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- A classifier is trained over these vectors to predict whether y is a hypernym of x.

Supervised Distributional Approach

- **Supervised hypernymy detection**: (x, y) is represented as a feature vector, based of the terms' embeddings vectors
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- A classifier is trained over these vectors to predict whether y is a hypernym of x.

- These methods achieved very good results on common hypernymy detection datasets

Supervised Distributional Approach

- **Supervised hypernymy detection**: (x, y) is represented as a feature vector, based of the terms' embeddings vectors
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- A classifier is trained over these vectors to predict whether y is a hypernym of x.

- These methods achieved very good results on common hypernymy detection datasets
- Is it a solved task?

Supervised Distributional Approach

- **Supervised hypernymy detection**: (x, y) is represented as a feature vector, based of the terms' embeddings vectors
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- A classifier is trained over these vectors to predict whether y is a hypernym of x.

- These methods achieved very good results on common hypernymy detection datasets
- Is it a solved task?
- Probably not. These methods don't learn about the *relation* between x and y, but mostly that y is a *prototypical hypernym* [Levy et al., 2015].

Path-based Approach

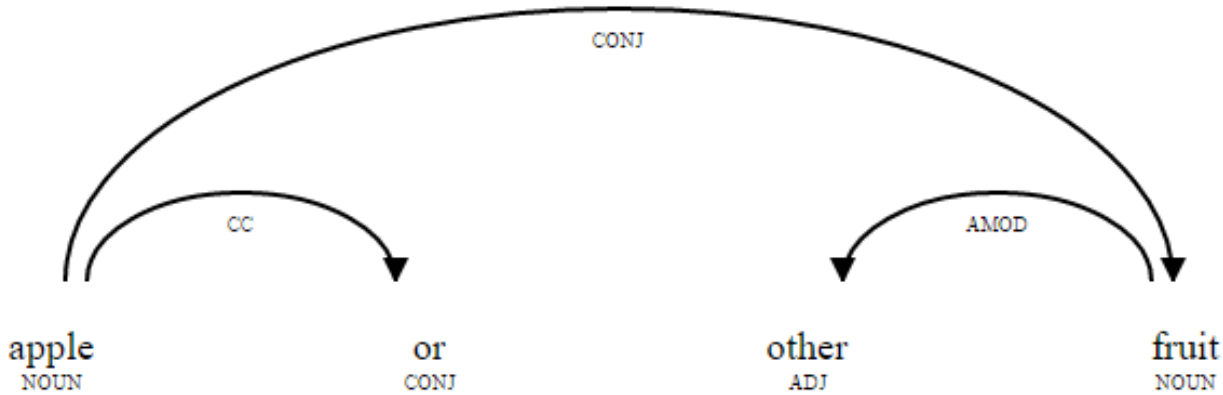
- The idea: the semantic relation between words can be recognized based on their *joint* occurrences in a corpus

Path-based Approach

- The idea: the semantic relation between words can be recognized based on their *joint* occurrences in a corpus
- Hearst Patterns [Hearst, 1992] - some patterns that connect x and y may indicate that y is a hypernym of x
  - e.g. *X or other Y*

Path-based Approach

- The idea: the semantic relation between words can be recognized based on their *joint* occurrences in a corpus
- Hearst Patterns [Hearst, 1992] - some patterns that connect x and y may indicate that y is a hypernym of x
  - e.g. *X or other Y*
- Patterns can be represented using dependency paths:

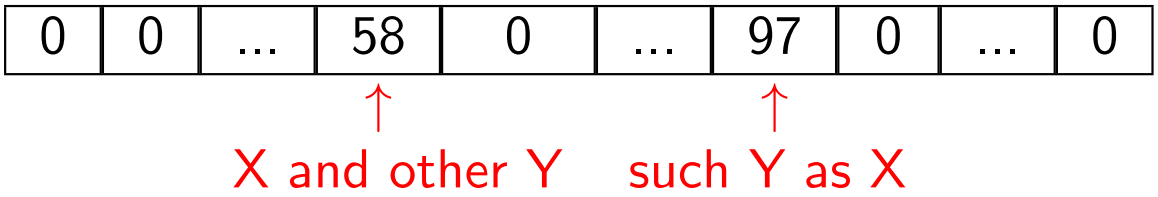


Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Predict whether y is a hypernym of x

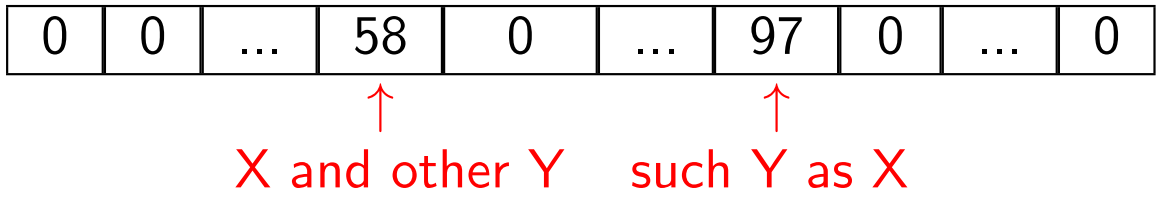
Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Predict whether y is a hypernym of x
  - Features: all dependency paths that connected x and y in a corpus:



Supervised Path-based Approach

- Supervised method to recognize hypernymy [Snow et al., 2004]:
  - Predict whether y is a hypernym of x
  - Features: all dependency paths that connected x and y in a corpus:



  - Supervision: set of known hyponym/hypernym pairs from WordNet

Path-based Approach Issues

- The feature space is too sparse:

Path-based Approach Issues

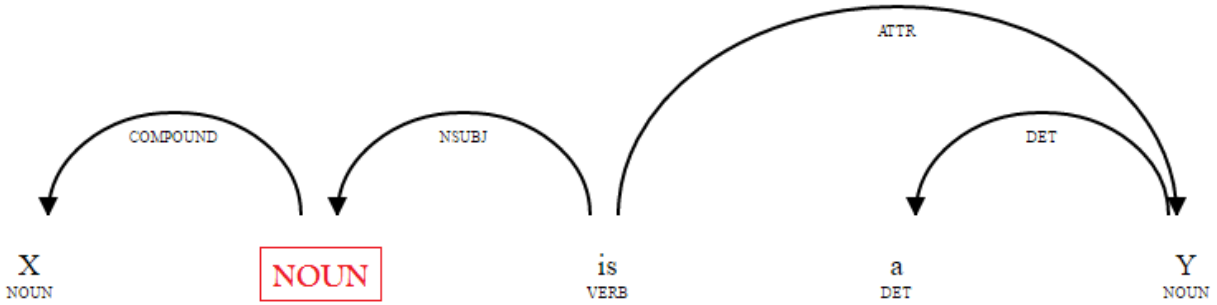
- The feature space is too sparse:
  - Some words along the path are not informative for hypernymy
  - e.g. X corporation/group/organization is a Y, e.g. *(Intel, company)*, *(Alibaba, company)*

Path-based Approach Issues

- The feature space is too sparse:
  - Some words along the path are not informative for hypernymy
  - e.g. X corporation/group/organization is a Y, e.g. *(Intel, company)*, *(Alibaba, company)*
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:

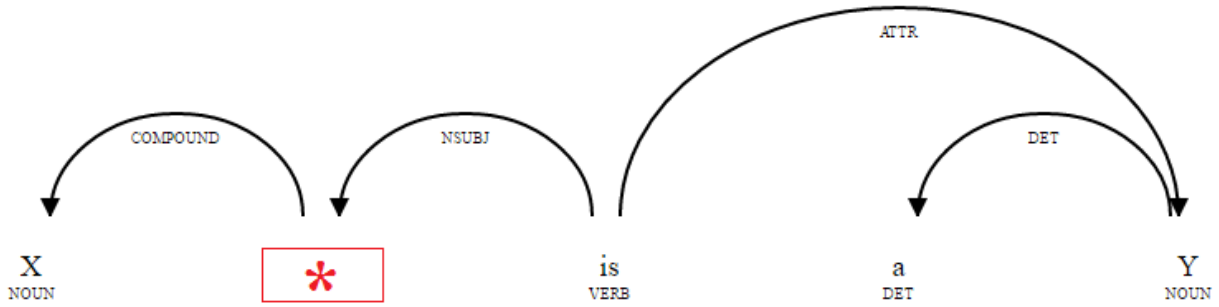
Path-based Approach Issues

- The feature space is too sparse:
  - Some words along the path are not informative for hypernymy
  - e.g. X corporation/group/organization is a Y, e.g. *(Intel, company)*, *(Alibaba, company)*
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - its POS tag



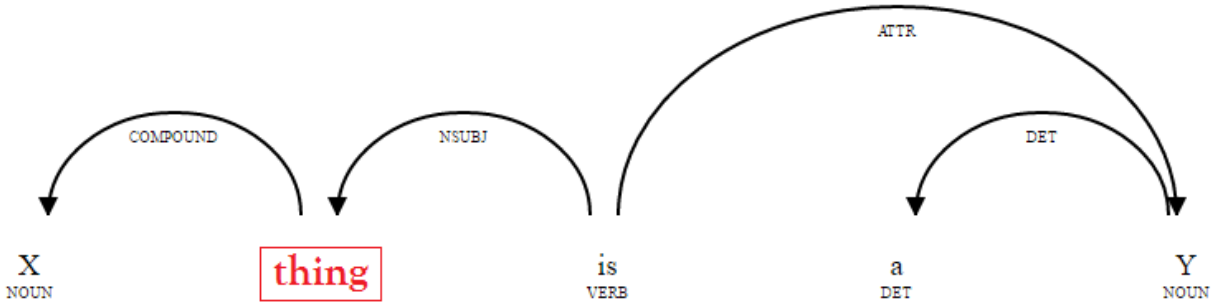
Path-based Approach Issues

- The feature space is too sparse:
  - Some words along the path are not informative for hypernymy
  - e.g. X corporation/group/organization is a Y, e.g. *(Intel, company)*, *(Alibaba, company)*
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - a wild-card



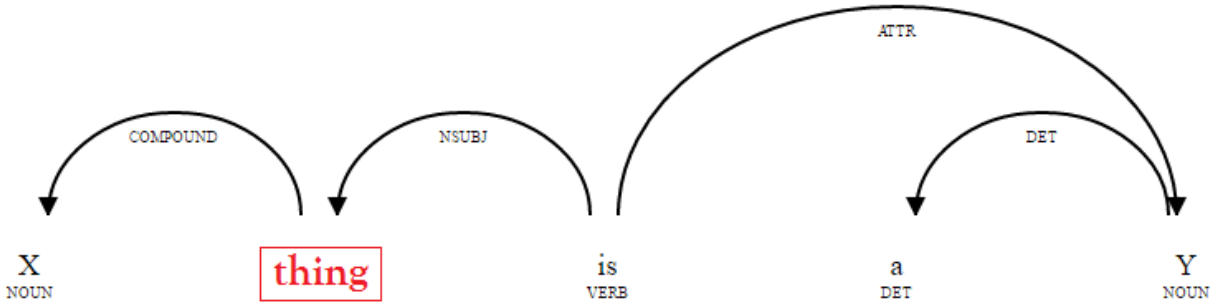
Path-based Approach Issues

- The feature space is too sparse:
  - Some words along the path are not informative for hypernymy
  - e.g. X corporation/group/organization is a Y, e.g. *(Intel, company)*, *(Alibaba, company)*
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - its ontological type



Path-based Approach Issues

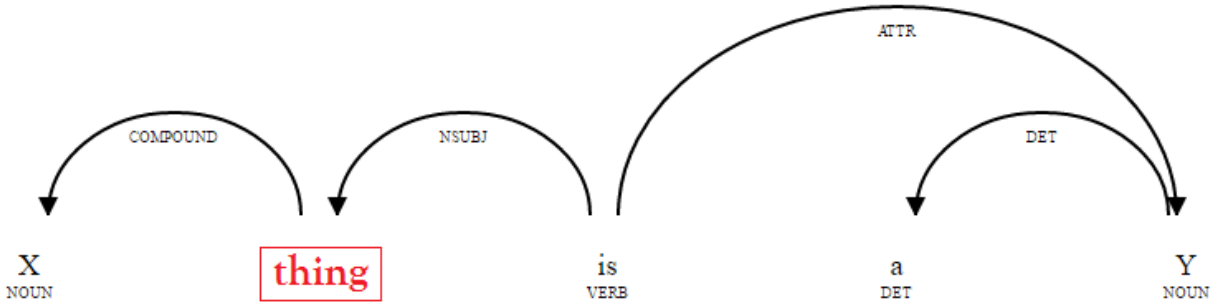
- The feature space is too sparse:
  - Some words along the path are not informative for hypernymy
  - e.g. X corporation/group/organization is a Y, e.g. *(Intel, company)*, *(Alibaba, company)*
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - its ontological type



- 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:
  - Some words along the path are not informative for hypernymy
  - e.g. X corporation/group/organization is a Y, e.g. *(Intel, company)*, *(Alibaba, company)*
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - its ontological type



- 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

First step: improving path representation (1/2)

1. Split each path to edges
   - "X is defined as Y" $\Rightarrow$
     `'X/NOUN/dobj/>', 'define/VERB/ROOT/-', 'as/ADP/prep/<', 'Y/NOUN/pobj/<'`
   - Each edge consists of 4 components:
     - source node lemma
     - source POS
     - edge dependency label
     - edge direction

First step: improving path representation (1/2)

1. Split each path to edges
   - "X is defined as Y" $\Rightarrow$
     `'X/NOUN/dobj/>', 'define/VERB/ROOT/-', 'as/ADP/prep/<', 'Y/NOUN/pobj/<'`
   - Each edge consists of 4 components:
     - source node lemma
     - source POS
     - edge dependency label
     - edge direction
   - We learn embedding vectors for each component
     - Lemma embeddings are initialized with pre-trained word embeddings

First step: improving path representation (1/2)

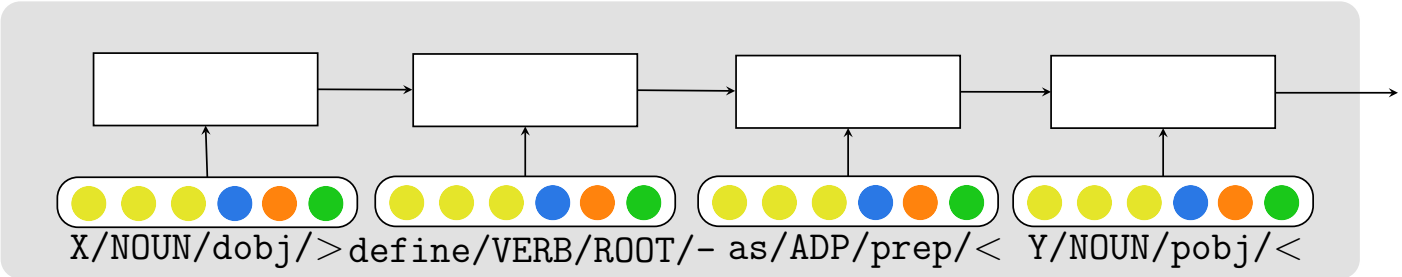
1. Split each path to edges
   - "X is defined as Y" $\Rightarrow$
     `'X/NOUN/dobj/>', 'define/VERB/ROOT/-', 'as/ADP/prep/<', 'Y/NOUN/pobj/<'`
   - Each edge consists of 4 components:
     - source node lemma
     - source POS
     - edge dependency label
     - edge direction
   - We learn embedding vectors for each component
     - Lemma embeddings are initialized with pre-trained word embeddings
   - The edge's vector is the concatenation of its components' vectors:

`as/ADP/prep/<`

   - Generalization: similar edges should have similar vectors!

First step: improving path representation (2/2)

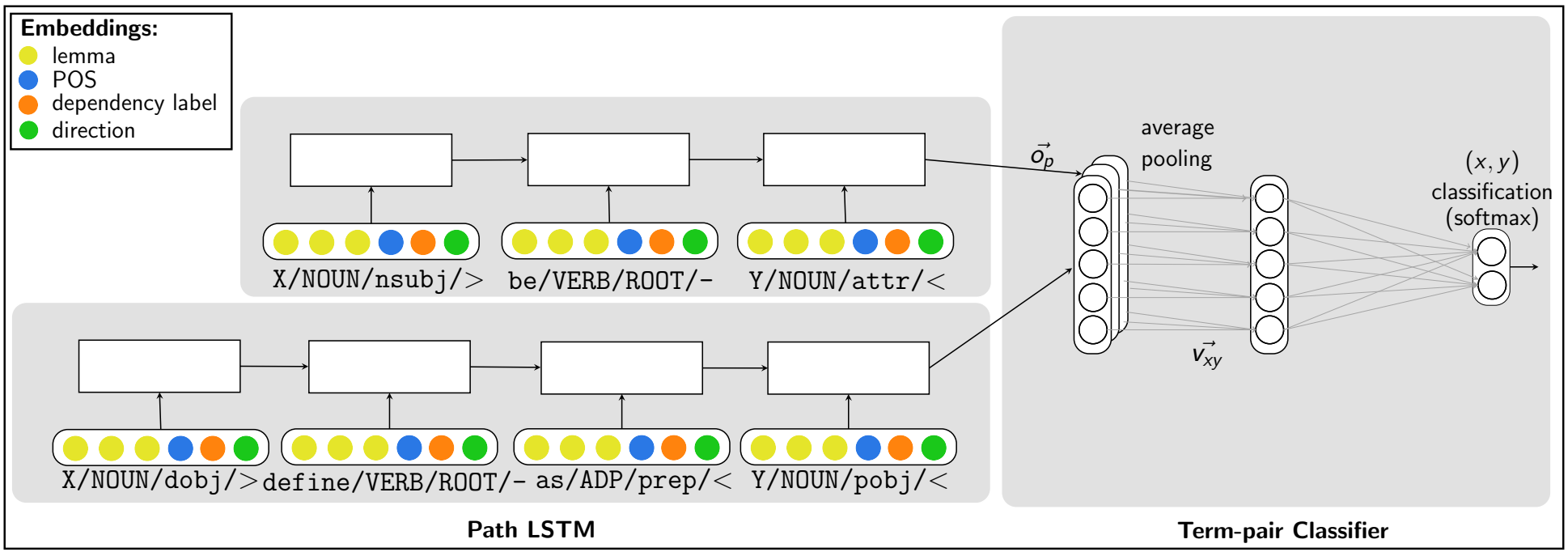
2. Feed the edges sequentially to an LSTM



- 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

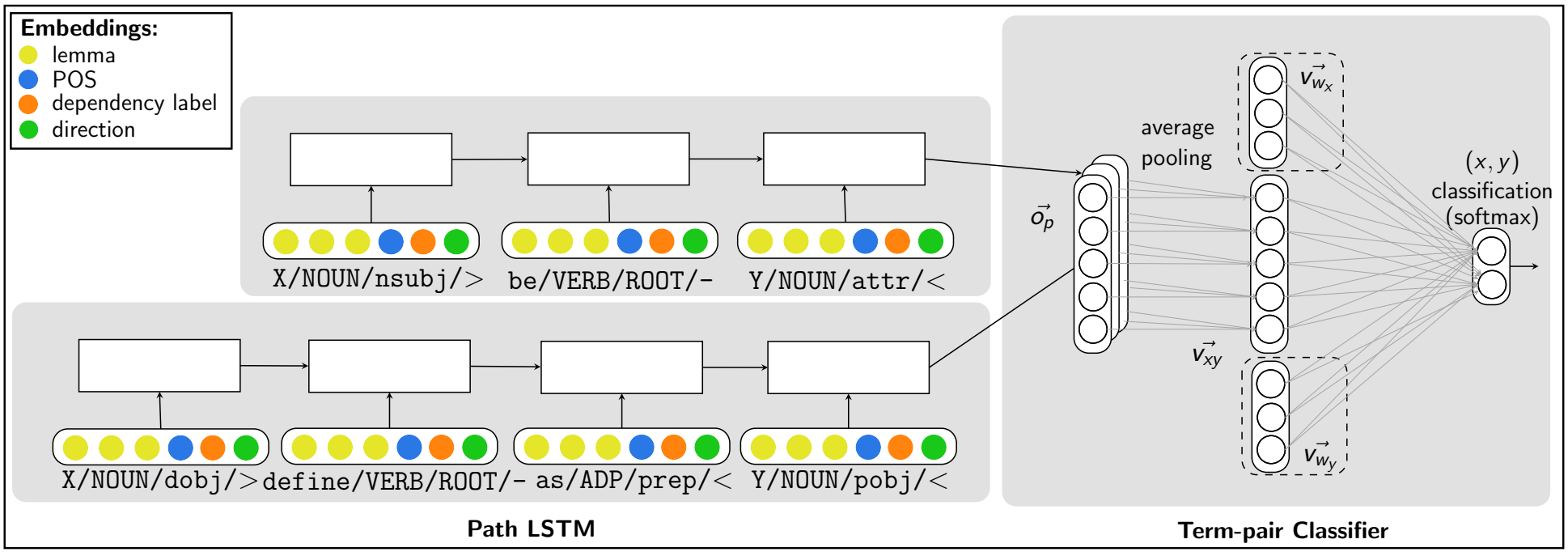
Second step: term-pair classification (1/2)

- 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):



Second step: term-pair classification (2/2)

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



Evaluation

Dataset

- Distant supervision from knowledge resources
  - Positive instances: term-pairs related in knowledge resource via hypernymy relations (e.g. *instance_of*)
  - Negative instances: term-pairs related via other relations
  - Filtering: pairs must co-occur at least twice - similar to [Snow et al., 2004]

Dataset

- Distant supervision from knowledge resources
  - Positive instances: term-pairs related in knowledge resource via hypernymy relations (e.g. *instance_of*)
  - Negative instances: term-pairs related via other relations
  - Filtering: pairs must co-occur at least twice - similar to [Snow et al., 2004]
- Two versions of train / test / validation split:

Dataset

- Distant supervision from knowledge resources
  - Positive instances: term-pairs related in knowledge resource via hypernymy relations (e.g. *instance_of*)
  - Negative instances: term-pairs related via other relations
  - Filtering: pairs must co-occur at least twice - similar to [Snow et al., 2004]
- Two versions of train / test / validation split:
  - Random (70% - 25% - 5%)

Dataset

- Distant supervision from knowledge resources
  - Positive instances: term-pairs related in knowledge resource via hypernymy relations (e.g. *instance_of*)
  - Negative instances: term-pairs related via other relations
  - Filtering: pairs must co-occur at least twice - similar to [Snow et al., 2004]
- Two versions of train / test / validation split:
  - Random (70% - 25% - 5%)
  - Lexical:
    - Distinct vocabulary in each set
    - Avoiding lexical memorization [Levy et al., 2015]

Results

method		random split			lexical split		
		precision	**recall**	**F_1**	**precision**	**recall**	**F_1**
Path-based	Snow	0.843	0.452	0.589	0.760	0.438	0.556
	Snow + GEN	0.852	0.561	0.676	0.759	0.530	0.624
	HypeNET Path-based	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	Unsupervised	0.491	0.737	0.589	0.375	0.610	0.464
	Supervised	0.901	0.637	0.746	0.754	0.551	0.637
Combined	HypeNET Integrated	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

Results

method		random split			lexical split		
		precision	**recall**	**F_1**	**precision**	**recall**	**F_1**
Path-based	Snow	0.843	0.452	0.589	0.760	0.438	0.556
	Snow + GEN	0.852	0.561	0.676	0.759	0.530	0.624
	HypeNET Path-based	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	Unsupervised	0.491	0.737	0.589	0.375	0.610	0.464
	Supervised	0.901	0.637	0.746	0.754	0.551	0.637
Combined	HypeNET Integrated	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

- Path-based:
  - Our method outperforms the baselines, including distributional

Results

method		random split			lexical split		
		precision	**recall**	**F_1**	**precision**	**recall**	**F_1**
Path-based	Snow	0.843	0.452	0.589	0.760	0.438	0.556
	Snow + GEN	0.852	0.561	0.676	0.759	0.530	0.624
	HypeNET Path-based	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	Unsupervised	0.491	0.737	0.589	0.375	0.610	0.464
	Supervised	0.901	0.637	0.746	0.754	0.551	0.637
Combined	HypeNET Integrated	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

- Path-based:
  - Our method outperforms the baselines, including distributional
  - The generalizations yield improved recall

Results

method		random split			lexical split		
		precision	**recall**	**F_1**	**precision**	**recall**	**F_1**
Path-based	Snow	0.843	0.452	0.589	0.760	0.438	0.556
	Snow + GEN	0.852	0.561	0.676	0.759	0.530	0.624
	HypeNET Path-based	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	Unsupervised	0.491	0.737	0.589	0.375	0.610	0.464
	Supervised	0.901	0.637	0.746	0.754	0.551	0.637
Combined	HypeNET Integrated	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

- Path-based:
  - Our method outperforms the baselines, including distributional
  - The generalizations yield improved recall
- The integrated method substantially outperforms both path-based and distributional methods

Analysis - Path Representation (1/2)

- Identify hypernymy-indicating paths:
  - Feed each path to the network as a term-pair instance: $\vec{v_{xy}} = [\vec{0}, \vec{o_p}, \vec{0}]$

Analysis - Path Representation (1/2)

- Identify hypernymy-indicating paths:
  - Feed each path to the network as a term-pair instance: $\vec{v_{xy}} = [\vec{0}, \vec{o_p}, \vec{0}]$
  - Score paths by their positive class score: $softmax(W \cdot \vec{v_{xy}})[1]$

Analysis - Path Representation (1/2)

- Identify hypernymy-indicating paths:
  - Feed each path to the network as a term-pair instance: $\vec{v_{xy}} = [\vec{0}, \vec{o_p}, \vec{0}]$
  - Score paths by their positive class score: $softmax(W \cdot \vec{v_{xy}})[1]$
  - Take the top k or above a threshold

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 very 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 very 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
  ...

What's Next?

- Detecting multiple semantic relations (*synonymy*, *co-hyponymy*, *meronymy*) - ongoing work, initial results:
  - Is the model extendable to classifying multiple relations?

What's Next?

- Detecting multiple semantic relations (*synonymy*, *co-hyponymy*, *meronymy*) - ongoing work, initial results:
  - Is the model extendable to classifying multiple relations?

    Yes!

What's Next?

- Detecting multiple semantic relations (*synonymy*, *co-hyponymy*, *meronymy*) - ongoing work, initial results:
  - Is the model extendable to classifying multiple relations?

    Yes!
  - Is it easier to distinguish similar semantic relations (e.g. *hypernymy* from *synonymy*) in a multiclass network?

What's Next?

- Detecting multiple semantic relations (*synonymy*, *co-hyponymy*, *meronymy*) - ongoing work, initial results:
  - Is the model extendable to classifying multiple relations?

    Yes!
  - Is it easier to distinguish similar semantic relations (e.g. *hypernymy* from *synonymy*) in a multiclass network?

    Generally, yes, to a small extent

What's Next?

- Detecting multiple semantic relations (*synonymy*, *co-hyponymy*, *meronymy*) - ongoing work, initial results:
  - Is the model extendable to classifying multiple relations?

    Yes!
  - Is it easier to distinguish similar semantic relations (e.g. *hypernymy* from *synonymy*) in a multiclass network?

    Generally, yes, to a small extent
  - What is the significance of each component (path-based and distributional) for each relation?

What's Next?

- Detecting multiple semantic relations (*synonymy*, *co-hyponymy*, *meronymy*) - ongoing work, initial results:
  - Is the model extendable to classifying multiple relations?

    Yes!
  - Is it easier to distinguish similar semantic relations (e.g. *hypernymy* from *synonymy*) in a multiclass network?

    Generally, yes, to a small extent
  - What is the significance of each component (path-based and distributional) for each relation?

    Each component is good at some relations and bad at others

What's Next?

- Detecting multiple semantic relations (*synonymy*, *co-hyponymy*, *meronymy*) - ongoing work, initial results:
  - Is the model extendable to classifying multiple relations?

    Yes!
  - Is it easier to distinguish similar semantic relations (e.g. *hypernymy* from *synonymy*) in a multiclass network?

    Generally, yes, to a small extent
  - What is the significance of each component (path-based and distributional) for each relation?

    Each component is good at some relations and bad at others

    e.g. synonyms don't tend to occur together - difficult for path-based

Questions?

References I


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.


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


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


Kotlerman, L., Dagan, I., Szpektor, I., and Zhitomirsky-Geffet, M. (2010).
Directional distributional similarity for lexical inference.
NLE, 16(04):359–389.

References II


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


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

References III


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


Rimell, L. (2014).
Distributional lexical entailment by topic coherence.
In *EACL*, pages 511–519.


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


Santus, E., Lenci, A., Lu, Q., and Im Walde, S. S. (2014).
Chasing hypernyms in vector spaces with entropy.
In *EACL*, pages 38–42.

References IV


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


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


Weeds, J. and Weir, D. (2003).
A general framework for distributional similarity.
In *EMLP*, pages 81–88.