Recognizing Lexical Inference

Vered Shwartz
Bar-Ilan University NLP Lab
26.04.2016

Motivation

- Question answering:

  Question: "When was *Friends* first aired?"
  Text: "*Friends* was first broadcast in 1994"
  Answer: 1994

Motivation (cont.)

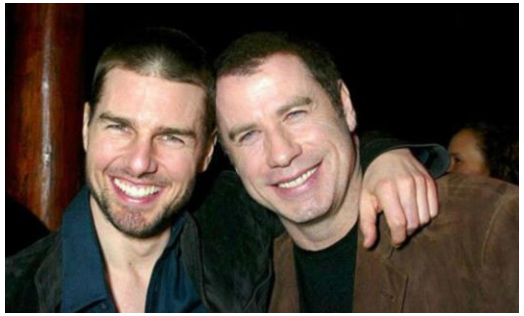
- Query Expansion:

Query: “Actors engaged in scientology“

Results:

Tom Cruise must ditch the vile cult of Scientology NOW before
www.dailymail.co.uk/.../PIERS-MORGAN-**Tom-Cruise**-ditch-vile-cult-**Scie**...
Apr 2, 2015 - PIERS MORGAN: I wanted to be **Tom Cruise**. As a fresh-faced, 21-year-old, I watched Top Gun a dozen times at my local movie theater in ...

John Travolta Says Scientology Is A Target Because It ...
www.huffingtonpost.com/.../**john-travolta-scientology**-target_n_710268...
Apr 20, 2015 - During an interview with "Good Morning America" Monday, **John Travolta** was asked why there is so much intrigue and interest surrounding the ...



Lexical Inference

- A directional semantic relation from one term (x) to another (y)

Lexical Inference

- A directional semantic relation from one term (x) to another (y)

- Encapsulates various relations, for example:
  - Synonymy: $(elevator, lift)$
  - Is a / hypernymy: $(apple, fruit)$, $(Barack\ Obama, president)$
  - Hyponymy: $(fruit, apple)$
  - Meronymy: $(London, England)$, $(chest, body)$
  - Holonymy: $(England, London)$, $(body, chest)$
  - Causality: $(flu, fever)$

Lexical Inference

- A directional semantic relation from one term (x) to another (y)

- Encapsulates various relations, for example:
  - Synonymy: $(elevator, lift)$
  - Is a / hypernymy: $(apple, fruit)$, $(Barack\ Obama, president)$
  - Hyponymy: $(fruit, apple)$
  - Meronymy: $(London, England)$, $(chest, body)$
  - Holonymy: $(England, London)$, $(body, chest)$
  - Causality: $(flu, fever)$

- Each relation is used to infer y from x ($x \rightarrow y$) in certain contexts:
  - I ate an $apple$ $\rightarrow$ I ate a $fruit$
  - I hate $fruit$ $\rightarrow$ I hate $apples$
  - I visited $London$ $\rightarrow$ I visited $England$
  - I left $London$ $\nrightarrow$ I left $England$ (What if I left to Manchester?)

Outline

- Learning to Exploit Structured Resources for Lexical Inference
- Improving Hypernymy Detection with an Integrated Path-based and Distributional Methods
- Future Work

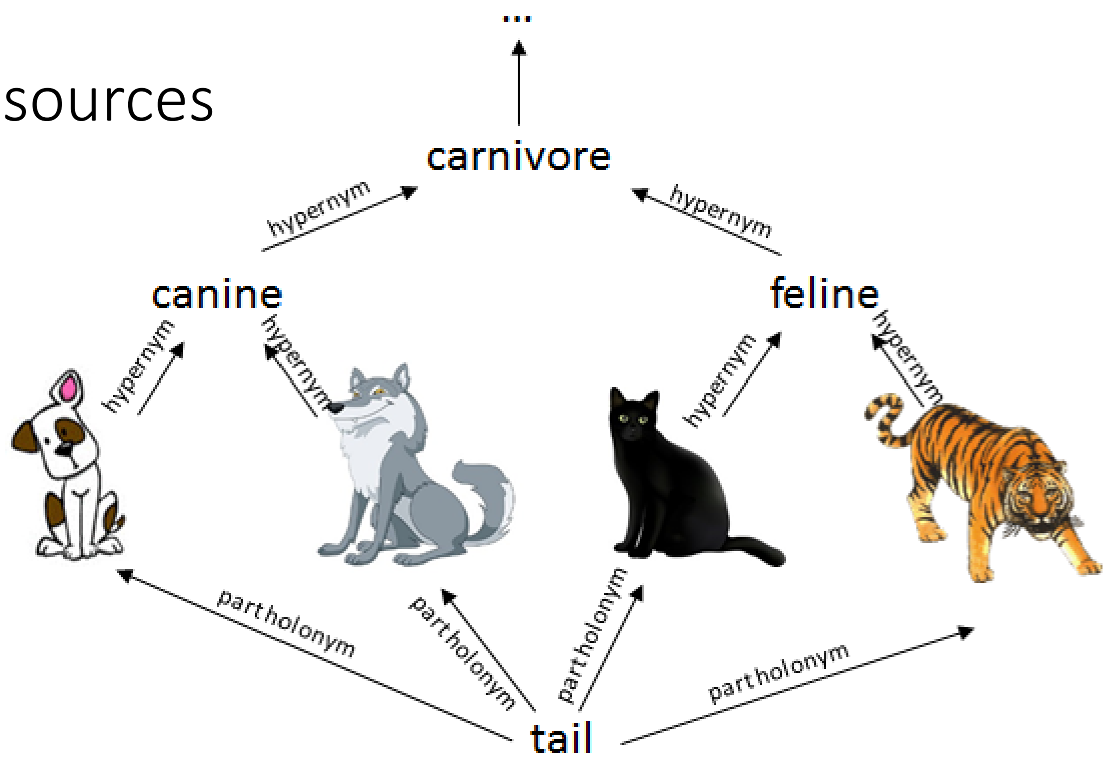
Learning to Exploit Structured Resources for Lexical Inference

Vered Shwartz, Omer Levy, Ido Dagan and Jacob Goldberger

CoNLL 2015

Resource-based methods for lexical inference

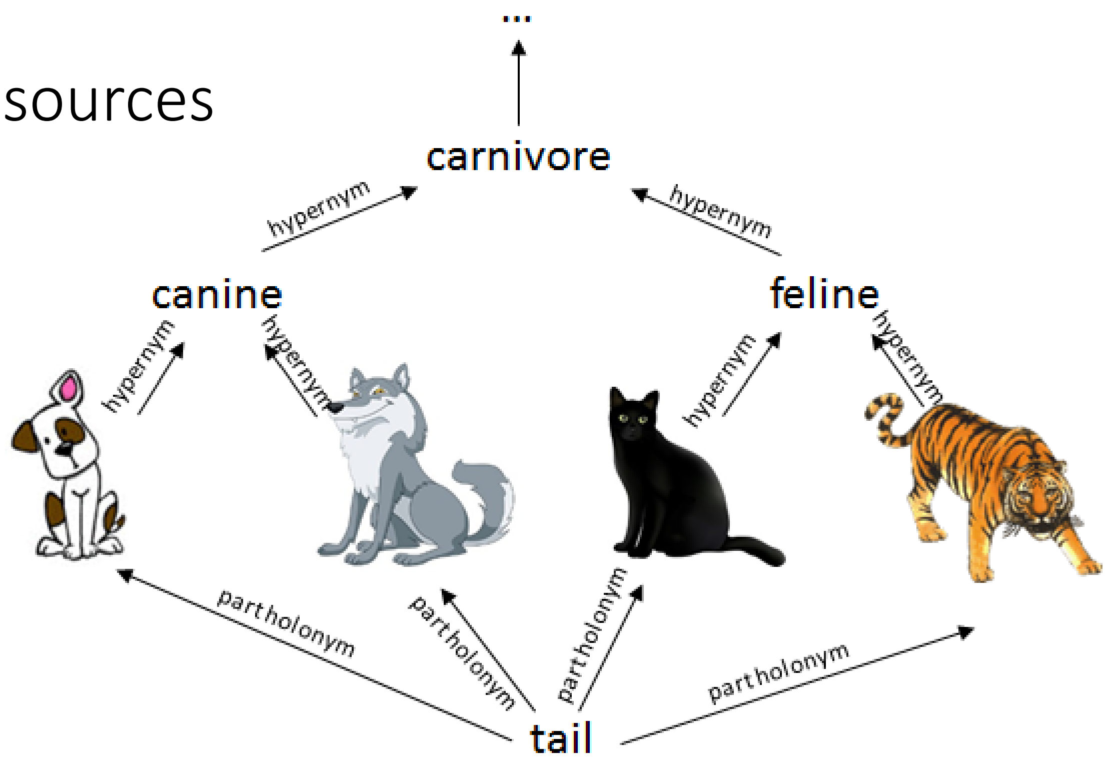
- Based on knowledge from hand-crafted resources
  - Dictionaries
  - Taxonomies (e.g. WordNet)



Resource-based methods for lexical inference

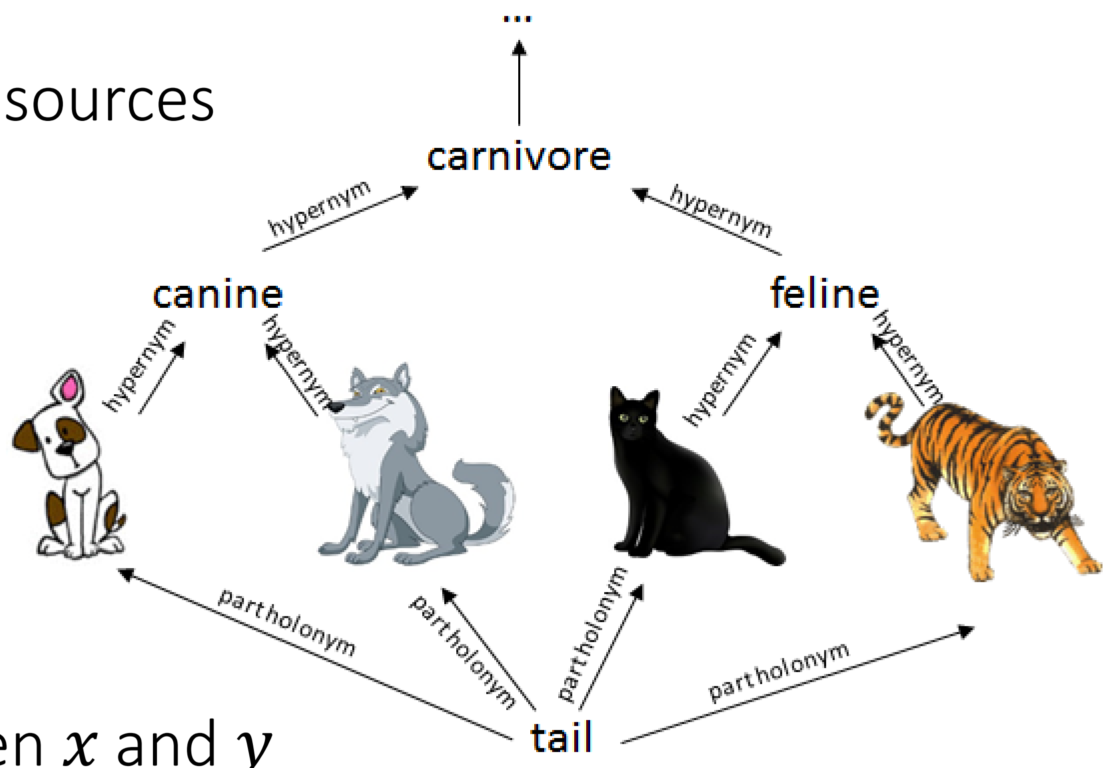
- Based on knowledge from hand-crafted resources
  - Dictionaries
  - Taxonomies (e.g. WordNet)

- Resources specify the lexical-semantic relation between terms



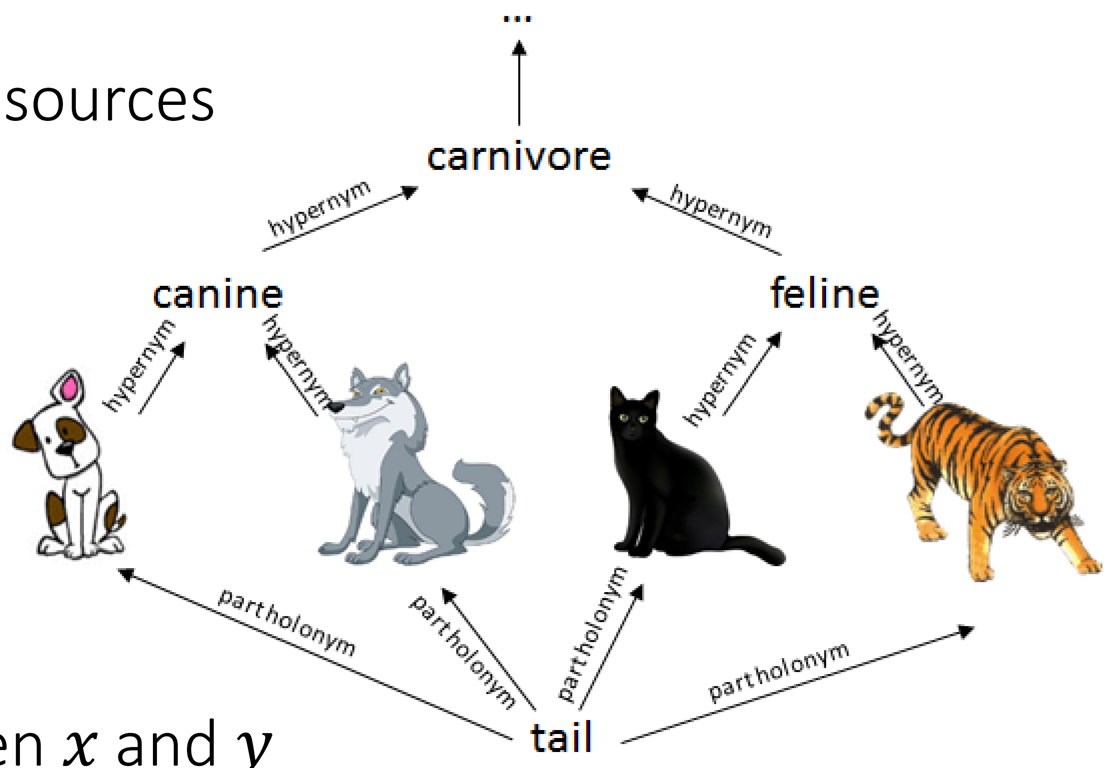
Resource-based methods for lexical inference

- Based on knowledge from hand-crafted resources
  - Dictionaries
  - Taxonomies (e.g. WordNet)
- Resources specify the lexical-semantic relation between terms
- The decision is based on the paths between x and y



Resource-based methods for lexical inference

- Based on knowledge from hand-crafted resources
  - Dictionaries
  - Taxonomies (e.g. WordNet)
- Resources specify the lexical-semantic relation between terms
- The decision is based on the paths between x and y
- Need to predefine which relations are relevant for the task



Resource-based methods for lexical inference

- High precision

Resource-based methods for lexical inference

- High precision
- Limited recall:
  - WordNet is small

Resource-based methods for lexical inference

- High precision
- Limited recall:
  - WordNet is small
  - Not up-to-date

    Recent terminology is missing: Social Network

Resource-based methods for lexical inference

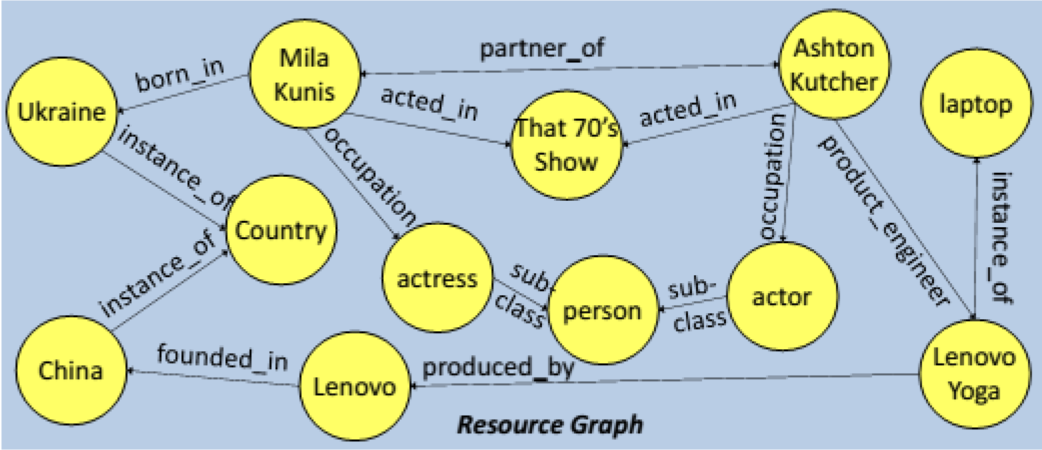
- High precision
- Limited recall:
  - WordNet is small
  - Not up-to-date

    Recent terminology is missing: Social Network
  - Contains mostly common nouns

    For example, it can’t tell us that *Lady Gaga* is a *singer*

Community-built Resources

- Huge
- Frequently updated
- Contain proper-names



WIKIDATA

6,000,000 entities in English

1,200 different properties

DBpedia

4,500,000 entities

1,367 different properties

yago select knowledge

10,000,000 entities in English

70 different properties

Utilizing Community-built Resources

- Idea: extend WordNet-based method using these resources

Utilizing Community-built Resources

- Idea: extend WordNet-based method using these resources

- Problem: utilizing these resources manually is infeasible
    - thousands of relations to select from!

Utilizing Community-built Resources

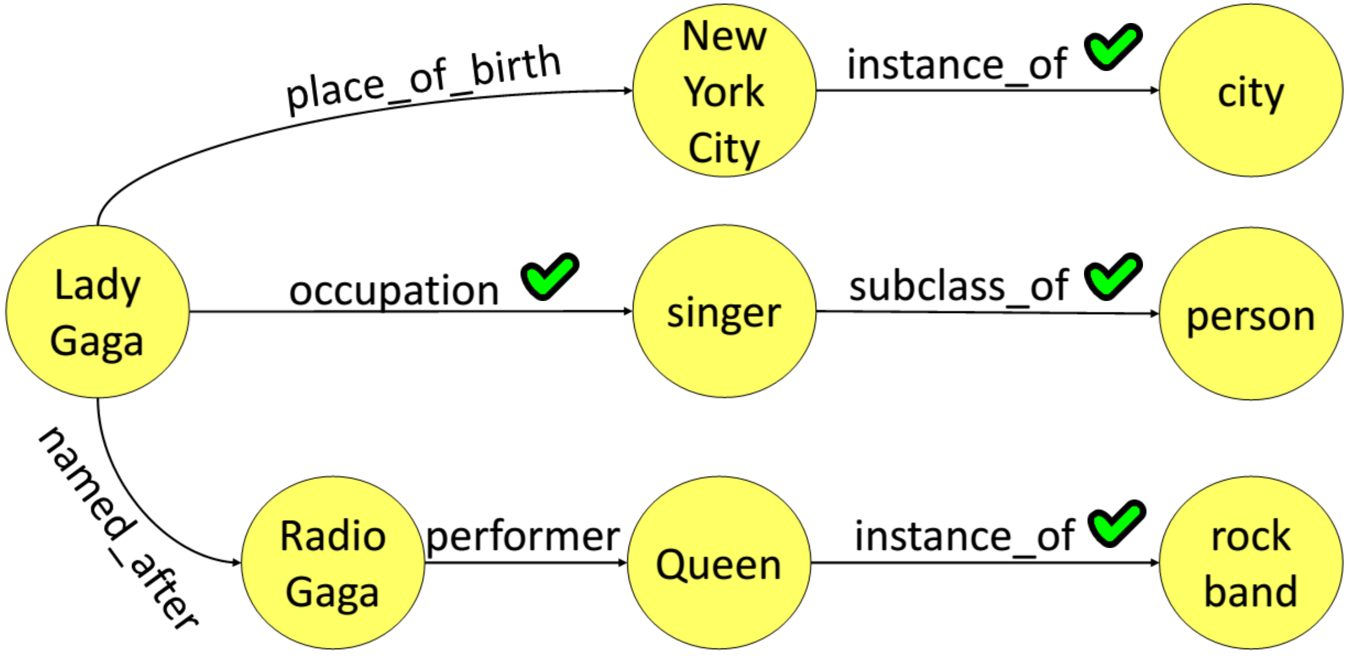
- Idea: extend WordNet-based method using these resources

- Problem: utilizing these resources manually is infeasible
    - thousands of relations to select from!

- Solution: learn to exploit these resources
  - Using genetic search

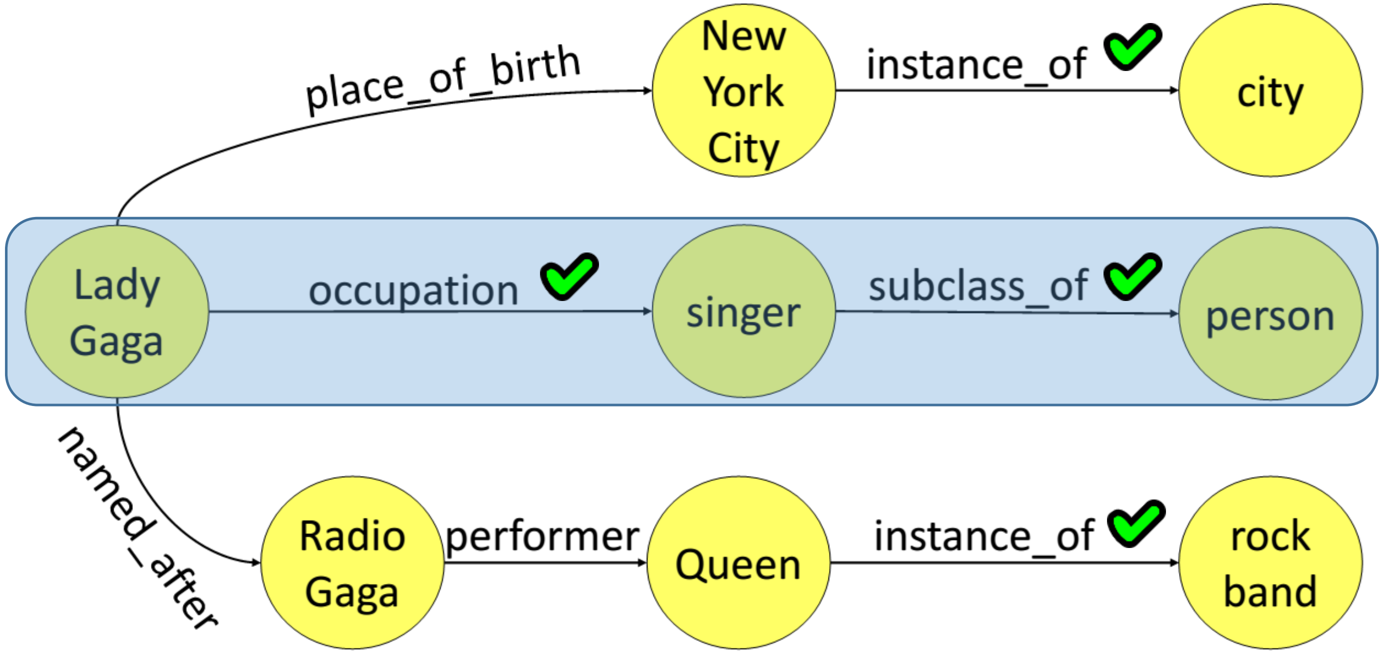
Our Method

- Training: learn which properties are indicative of given lexical inference relation (e.g. "is a")



Our Method

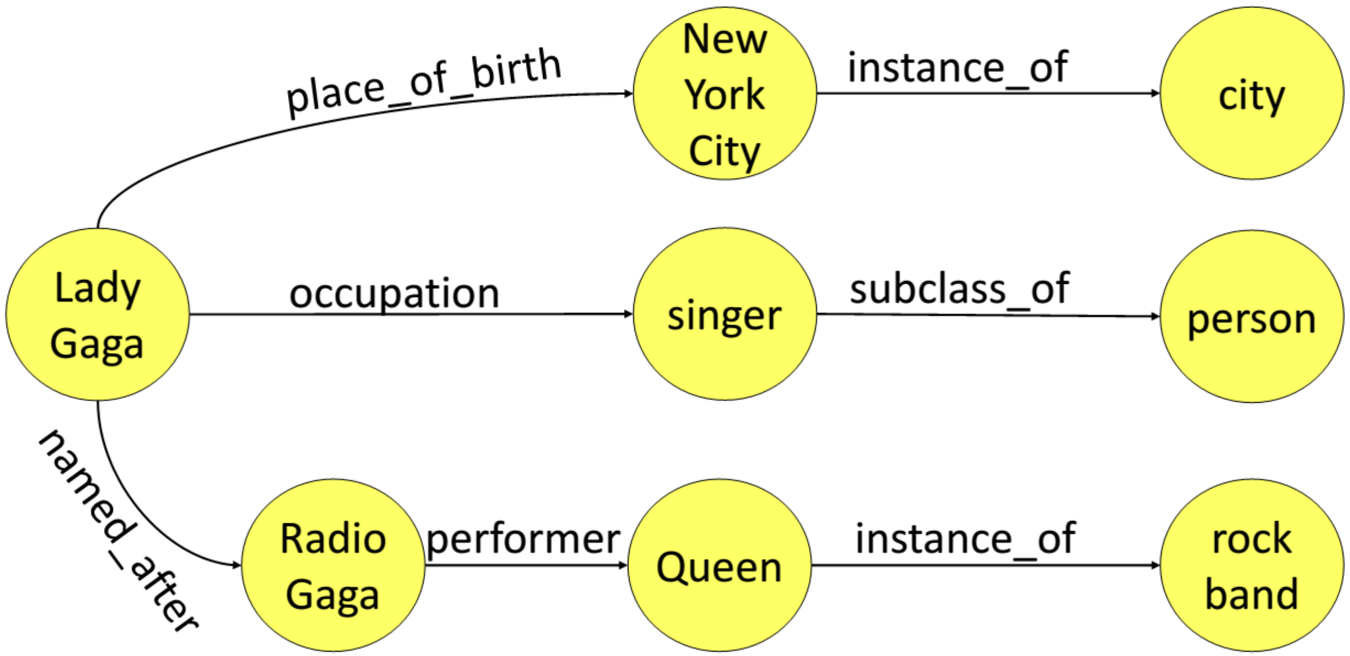
- Training: learn which properties are indicative of given lexical inference relation (e.g. "is a")



- Inference: $x \rightarrow y$ if there is a path of indicative edges from x to y

Results

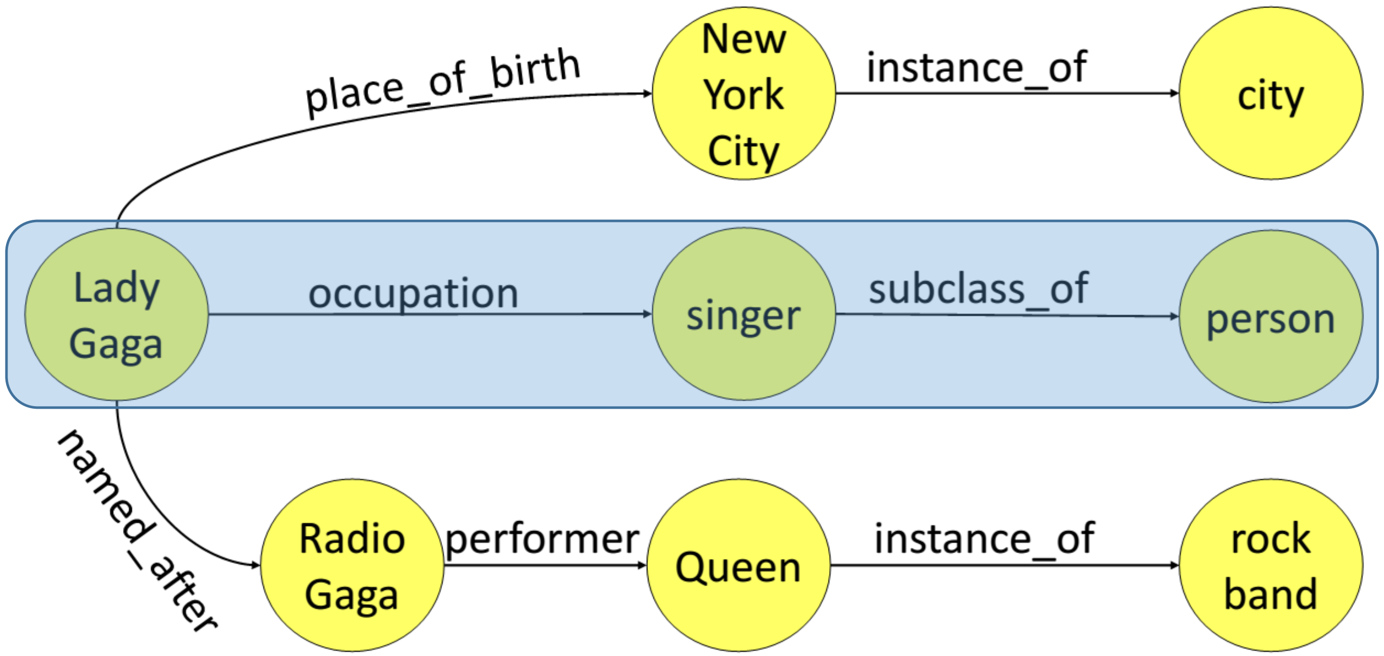
- We replicate WordNet-based methods for common nouns



Results

- We replicate WordNet-based methods for common nouns
- We extract high-precision inferences including proper-names:

$$\mathit{Lady\ Gaga} \rightarrow \mathit{person}$$



Results

- Non-trivial resource relations are learned:

occupation	$Daniel\ Radcliffe \rightarrow actor$
gender	$Louisa\ May\ Alcott \rightarrow woman$
position in sports team	$Jason\ Collins \rightarrow center$

Results

- Non-trivial resource relations are learned:

occupation	$Daniel\ Radcliffe \rightarrow actor$
gender	$Louisa\ May\ Alcott \rightarrow woman$
position in sports team	$Jason\ Collins \rightarrow center$

- We complement corpus-based methods in high-precision scenarios

Improving Hypernymy Detection with an Integrated Path-based and Distributional Method

Vered Shwartz, Yoav Goldberg, and Ido Dagan

Submitted to ACL 2016

Hypernymy Detection

- We focus on detecting hypernymy relations, which are common in inference:
  - $(apple, fruit)$
  - $(Barack\ Obama, president)$

Outline – Hypernymy Detection

- Prior Methods
- Our Method
- Evaluation

Outline – Hypernymy Detection

- Prior Methods
- Our Method
- 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
    - e.g. $elevator$ and $lift$ will both appear next to $down$, up, $building$, $floor$, 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 $\boldsymbol{down}$, $\boldsymbol{up}$, $\boldsymbol{building}$, $\boldsymbol{floor}$, and $\boldsymbol{stairs}$

- Measuring word similarity:
  - Represent words as distributional vectors

0	0	…	12	0	…	43	0	…	0
			↑ down			↑ up			

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 $down$, up, $building$, $floor$, and $stairs$

- Measuring word similarity:
  - Represent words as distributional vectors

0	0	…	12	0	…	43	0	…	0
			down			up			

  - Measure the distance between the vectors (e.g. cosine similarity)

Unsupervised Distributional Methods

- But…
  - Word similarity != lexical inference

Unsupervised Distributional Methods

- But…
  - Word similarity != lexical inference
  - Antonyms are similar e.g. small, big

Unsupervised Distributional Methods

- But…
  - Word similarity != lexical inference
  - Antonyms are similar e.g. small, big
  - Mutually exclusive terms are also similar e.g. football, basketball

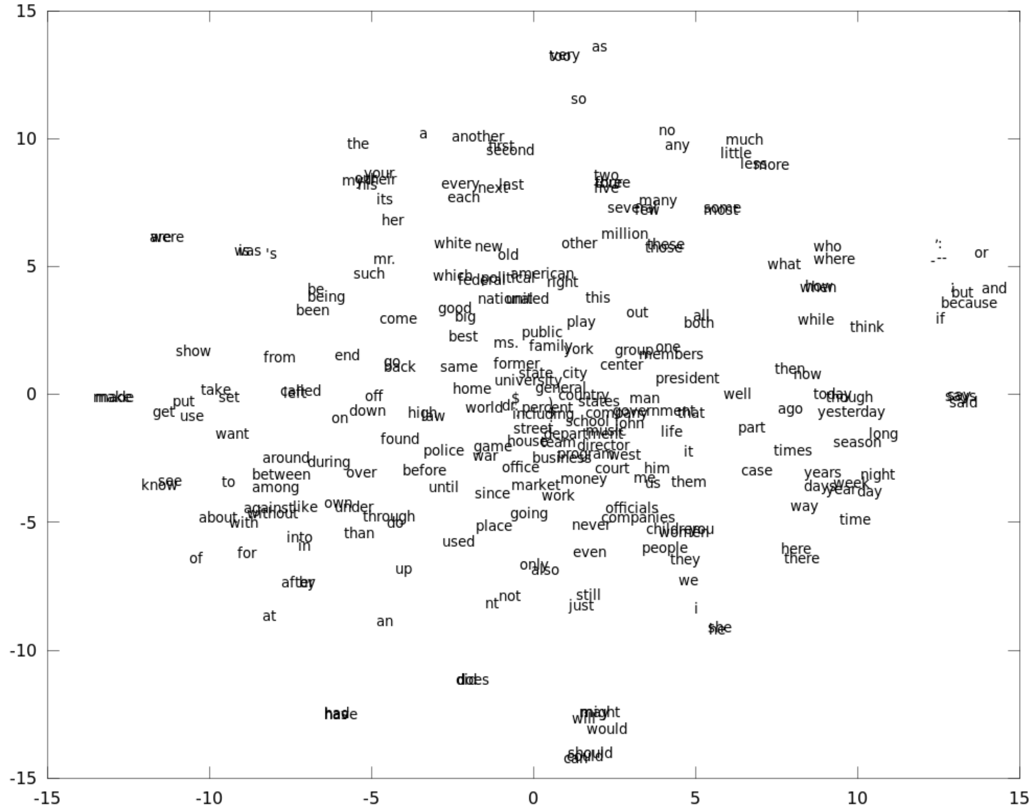
Unsupervised Distributional Methods

- But…
  - Word similarity != lexical inference
  - Antonyms are similar — e.g. small, big
  - Mutually exclusive terms are also similar — e.g. football, basketball

- Directional similarity
  - Inclusion: If $x \rightarrow y$, then the contexts of x are expected to be possible contexts for y (Weeds and Weir, 2003; Kotlerman et. al, 2010)
  - Generality: the most typical linguistic contexts of a hypernym are less informative than those of its hyponyms (Santus et al., 2014; Rimell, 2014).

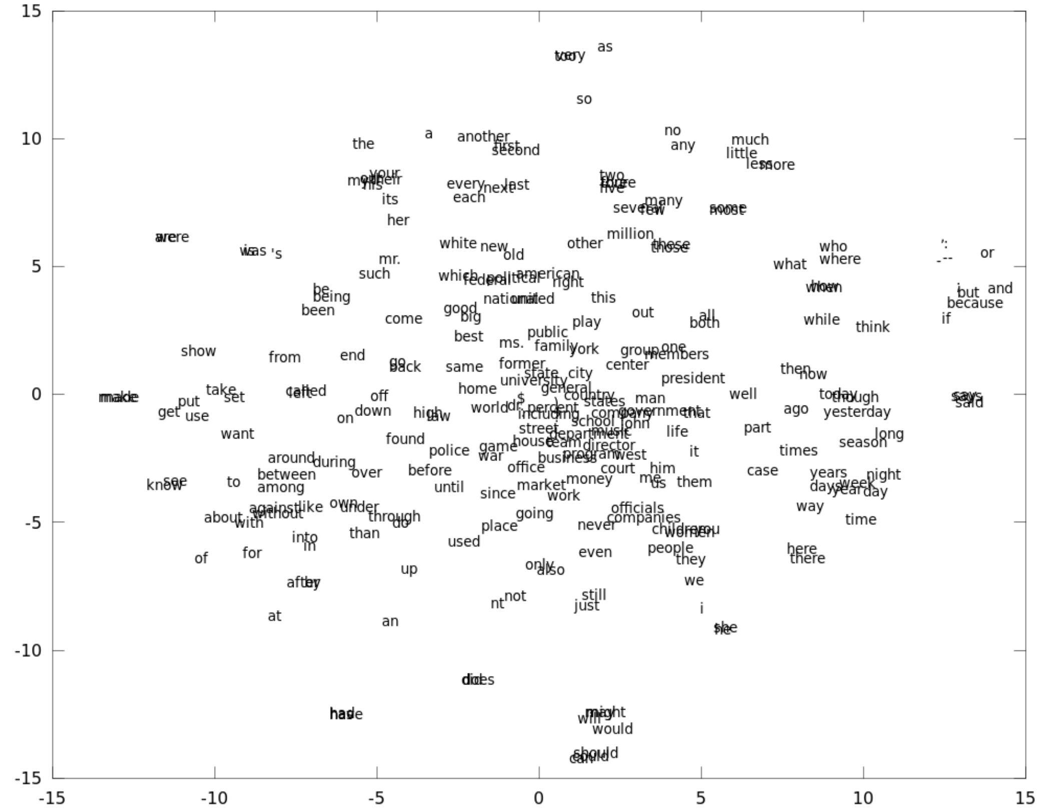
Supervised Distributional Methods

- Based on Word Embeddings
  - Distributional vectors are high-dimensional and sparse



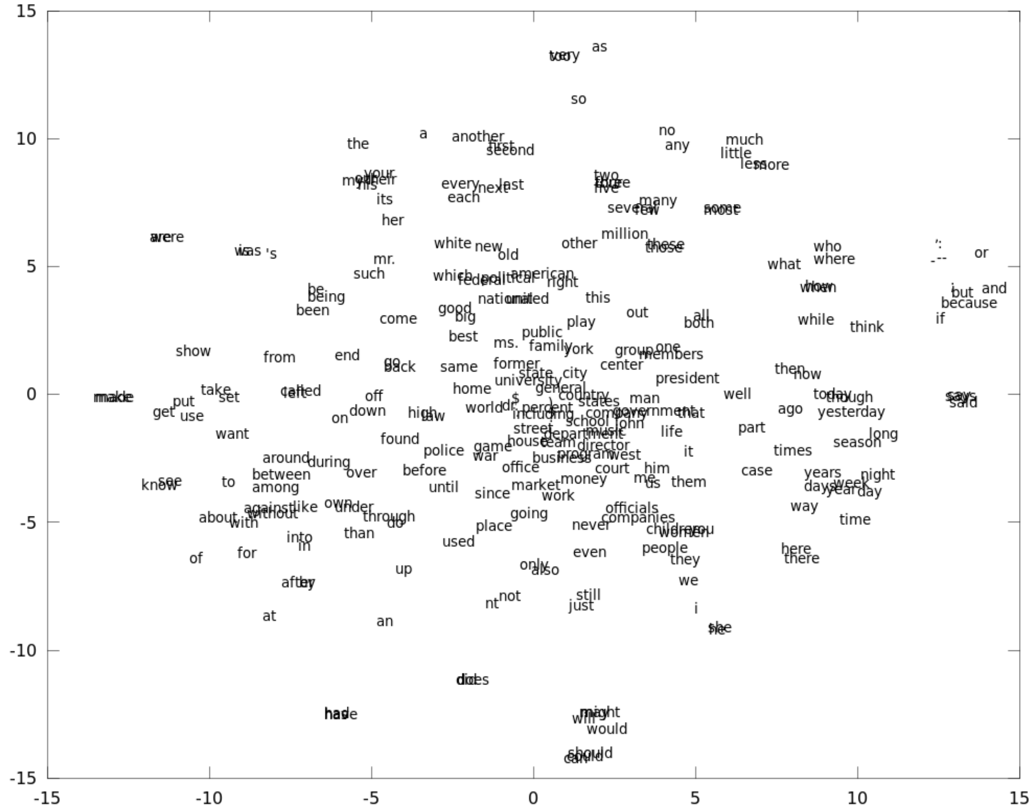
Supervised Distributional Methods

- Based on Word Embeddings
  - Distributional vectors are high-dimensional and sparse
  - Word embeddings are dense and low-dimensional - more efficient



Supervised Distributional Methods

- Based on Word Embeddings
  - Distributional vectors are high-dimensional and sparse
  - Word embeddings are dense and low-dimensional - more efficient
  - Similar words are still close to each other in the vector space
  - Bengio et al. (2003), word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014)



Supervised Distributional Methods

- Represent (x, y) as a combination of each term embeddings vector:
  - Concatenation $\vec{x} \oplus \vec{y}$ (Baroni et al., 2012)
  - Difference $\vec{y} - \vec{x}$ (Roller et al., 2014; Fu et al.,2014; Weeds et al., 2014)
  - Similarity $\vec{x} \cdot \vec{y}$

- Train a classifier over these vectors to predict entailment / hypernymy

Supervised Distributional Methods

- Represent (x, y) as a combination of each term embeddings vector:
  - Concatenation $\vec{x} \oplus \vec{y}$ (Baroni et al., 2012)
  - Difference $\vec{y} - \vec{x}$ (Roller et al., 2014; Fu et al.,2014; Weeds et al., 2014)
  - Similarity $\vec{x} \cdot \vec{y}$

- Train a classifier over these vectors to predict entailment / hypernymy
- Achieved high performance

Supervised Distributional Methods

- Represent (x, y) as a combination of each term embeddings vector:
  - Concatenation $\vec{x} \oplus \vec{y}$ (Baroni et al., 2012)
  - Difference $\vec{y} - \vec{x}$ (Roller et al., 2014; Fu et al.,2014; Weeds et al., 2014)
  - Similarity $\vec{x} \cdot \vec{y}$

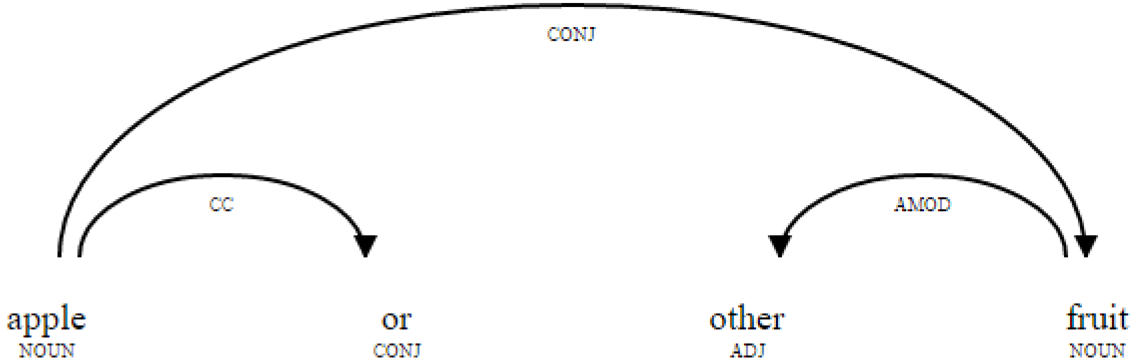
- Train a classifier over these vectors to predict entailment / hypernymy
- Achieved high performance
- However, these methods don't learn anything about the relation between x and y – they only learn characteristics of each term (Levy et al., 2015).

Path-based approach

- Based on **joint** occurrences of *x* and *y*

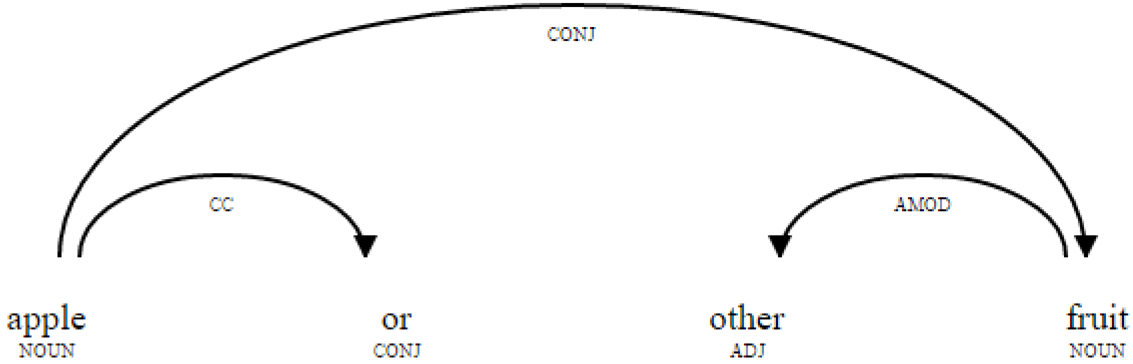
Path-based approach

- Based on **joint** occurrences of *x* and *y*
- **lexico-syntactic paths** = dependency paths or textual patterns, with POS tags and lemma



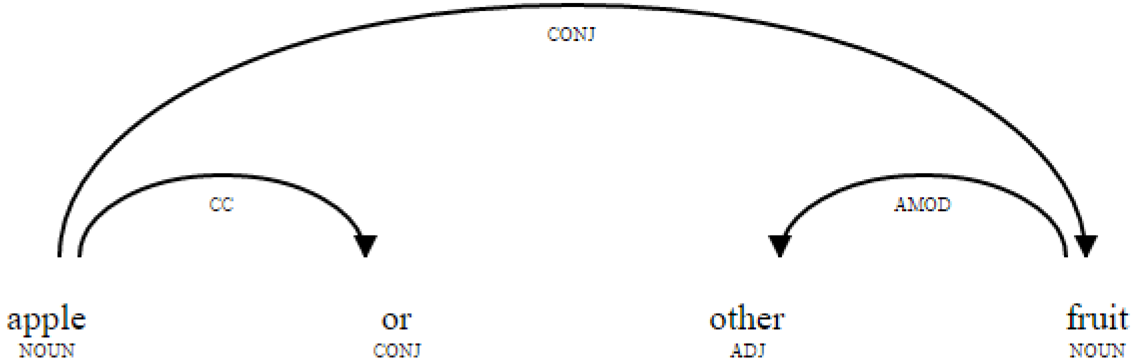
Path-based approach

- Based on **joint** occurrences of *x* and *y*
- **lexico-syntactic paths** = dependency paths or textual patterns, with POS tags and lemma
- Some patterns indicate semantic relations between terms:
  - e.g. X or other Y indicates that X is of type Y



Path-based approach

- Based on **joint** occurrences of *x* and *y*
- **lexico-syntactic paths** = dependency paths or textual patterns, with POS tags and lemma
- Some patterns indicate semantic relations between terms:
  - e.g. X or other Y indicates that X is of type Y
- If x and y hold a certain semantic relation, they are expected to occur in the corpus as the arguments of such patterns
  - e.g. apple or other fruit



Hearst Patterns

- Hearst (1992) - automatic acquisition of hypernyms

Hearst Patterns

- Hearst (1992) - automatic acquisition of hypernyms

- Found a few indicative patterns based on occurrences of known hypernyms in the corpus:

Y such as X

such Y as X

X or other Y

X and other Y

Y including X

Y, especially X

Snow et al. (2004)

- Supervised method to recognize hypernymy
  - Predict whether y is a hypernym of x

Snow et al. (2004)

- Supervised method to recognize hypernymy
  - Predict whether y is a hypernym of x
  - Supervision: set of known hyponym/hypernym pairs

Snow et al. (2004)

- Supervised method to recognize hypernymy
  - Predict whether y is a hypernym of x
  - Supervision: set of known hyponym/hypernym pairs
  - Features: all dependency paths between x and y in a corpus

0	0	…	12	0	…	43	0	…	0

"x and other y" *"such y as x"*

Snow et al. (2004)

- Supervised method to recognize hypernymy
  - Predict whether y is a hypernym of x
  - Supervision: set of known hyponym/hypernym pairs
  - Features: all dependency paths between x and y in a corpus

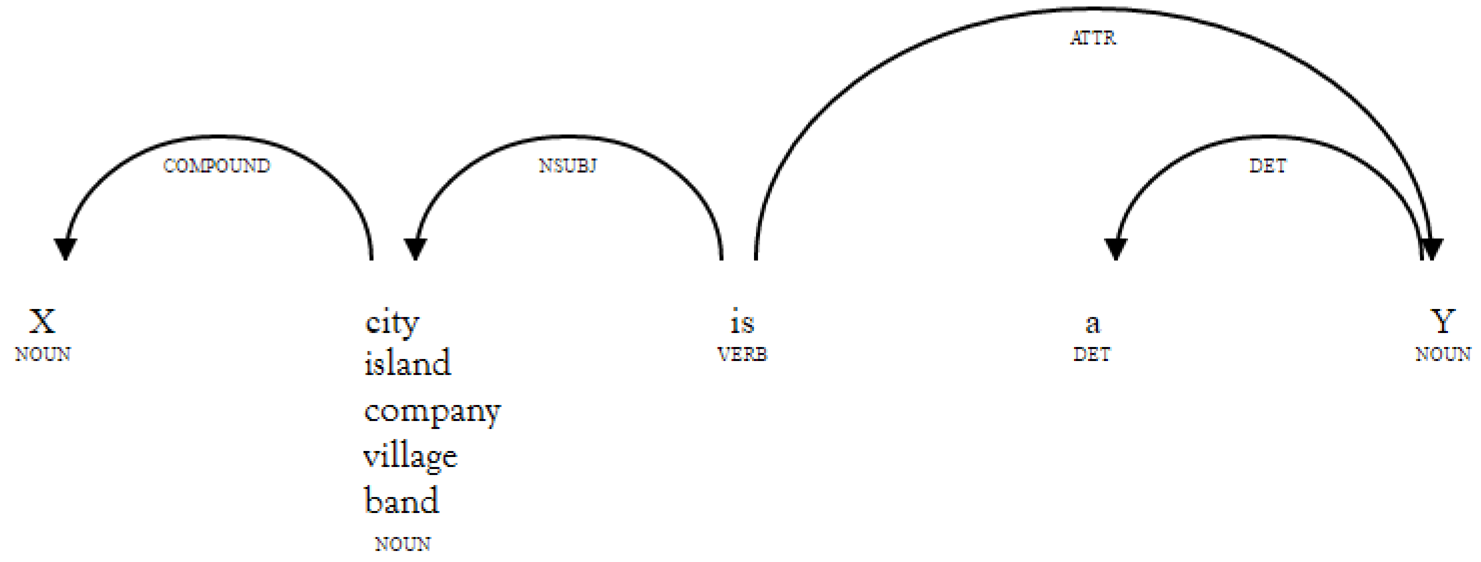
0	0	…	12	0	…	43	0	…	0

"x and other y" *"such y as x"*

  - Successfully restores Hearst patterns (and adds many more)
  - Used for analogy identification, taxonomy creation, etc.

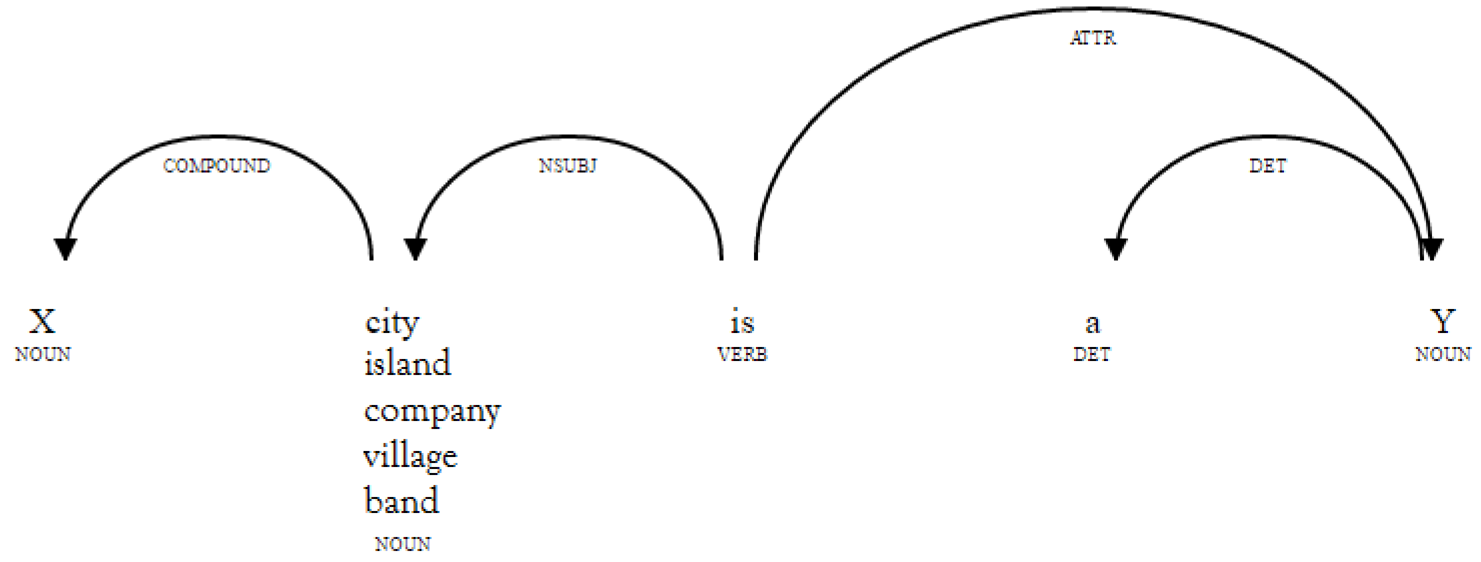
Problem with lexico-syntactic paths

- The feature space is too sparse:



Problem with lexico-syntactic paths

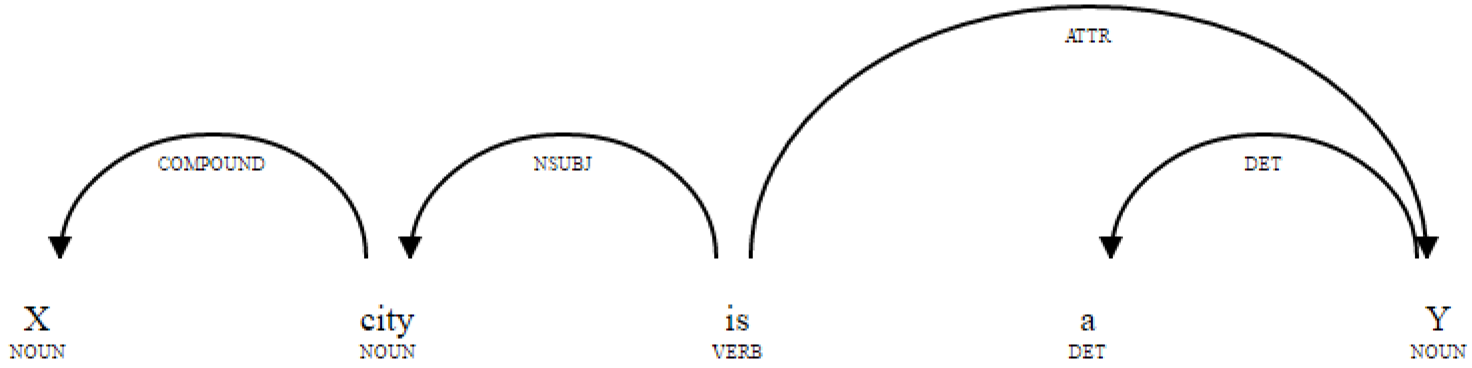
- The feature space is too sparse:



- Some words along the path don’t change the meaning

PATTY

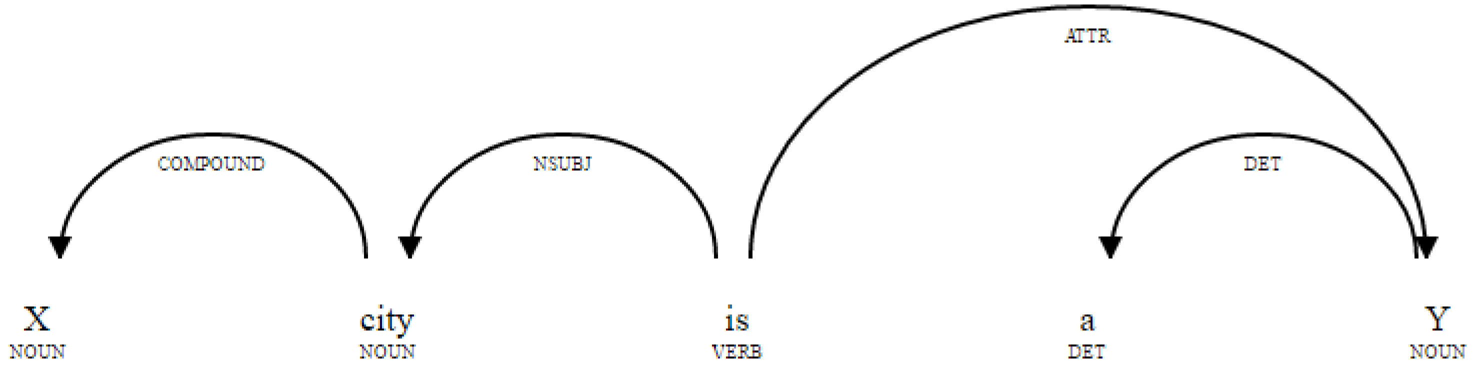
- A taxonomy created from free text (Nakashole et al., 2012)
- The relation between terms is based on the dependency paths between them



PATTY

- A taxonomy created from free text (Nakashole et al., 2012)
- The relation between terms is based on the dependency paths between them

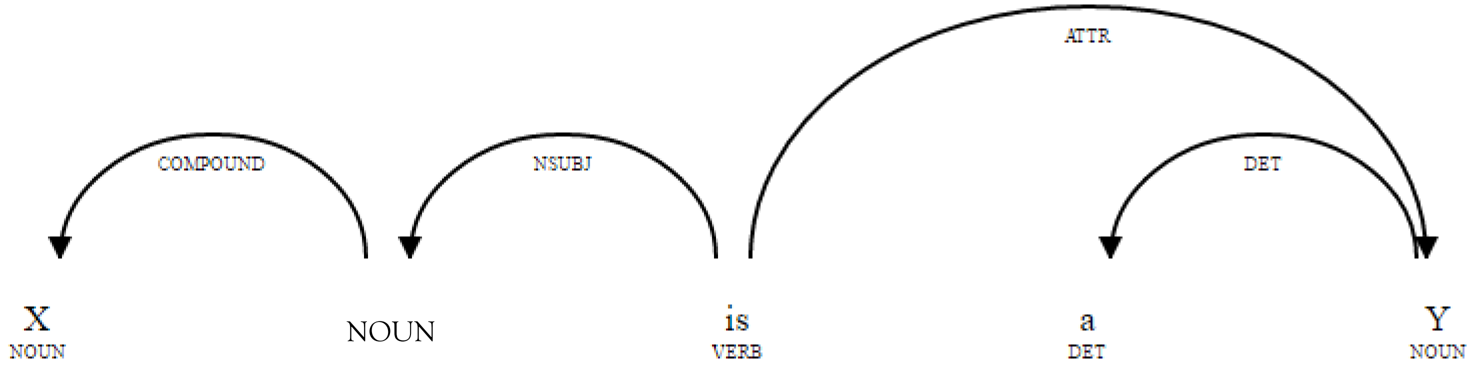
- Paths are generalized – a word might be replaced by:



PATTY

- A taxonomy created from free text (Nakashole et al., 2012)
- The relation between terms is based on the dependency paths between them

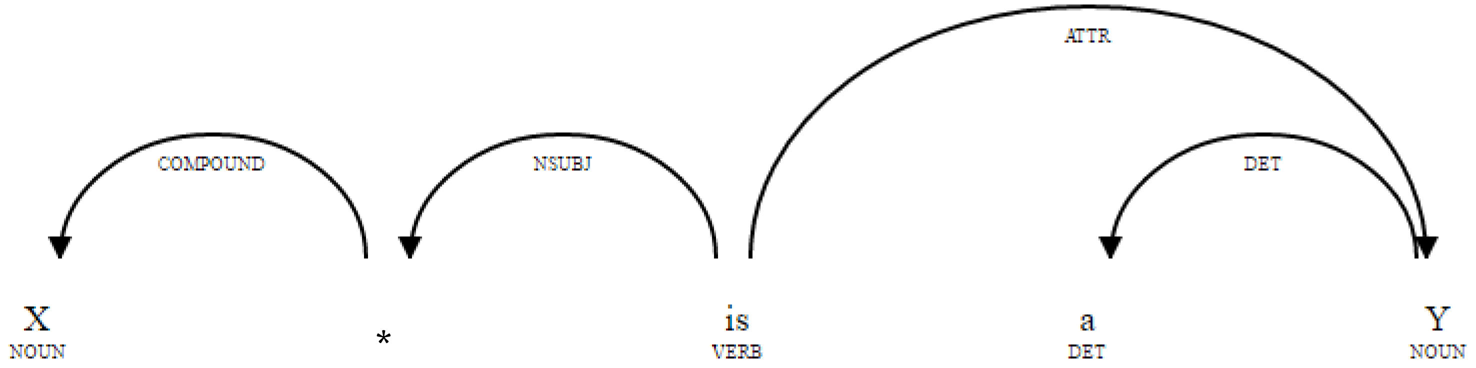
- Paths are generalized – a word might be replaced by:
  - its POS tag



PATTY

- A taxonomy created from free text (Nakashole et al., 2012)
- The relation between terms is based on the dependency paths between them

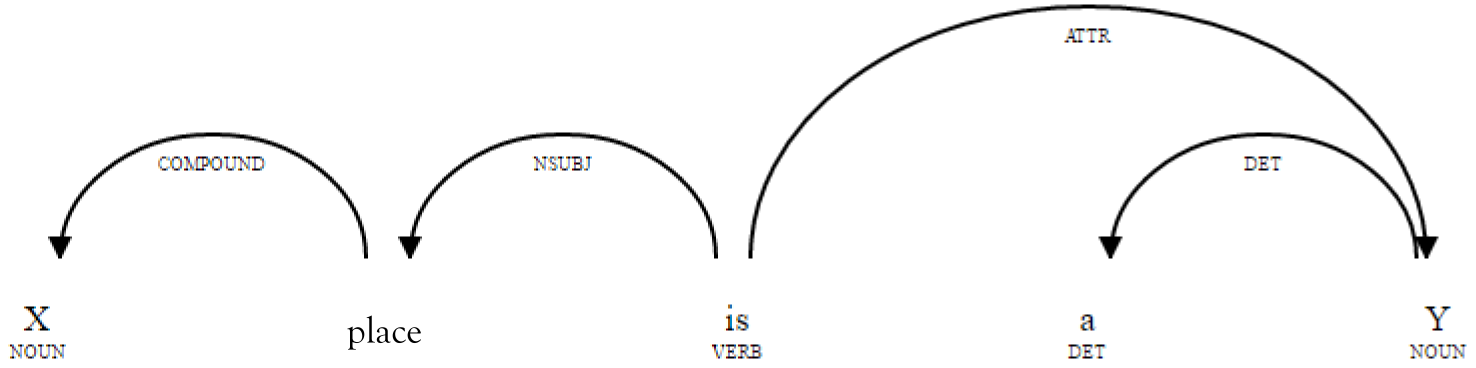
- Paths are generalized – a word might be replaced by:
  - its POS tag
  - a wild card



PATTY

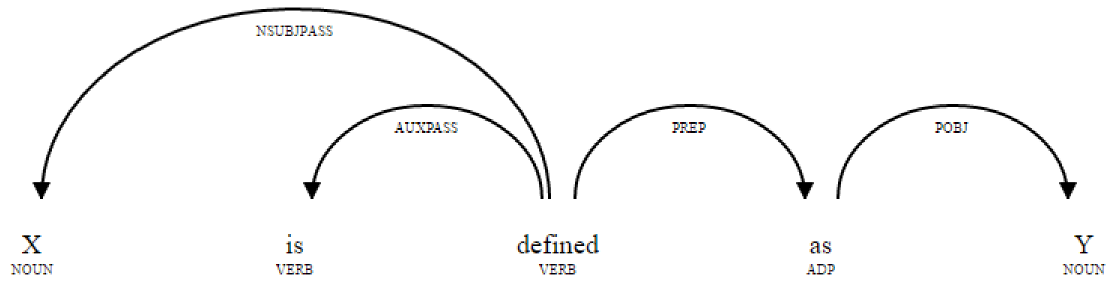
- A taxonomy created from free text (Nakashole et al., 2012)
- The relation between terms is based on the dependency paths between them

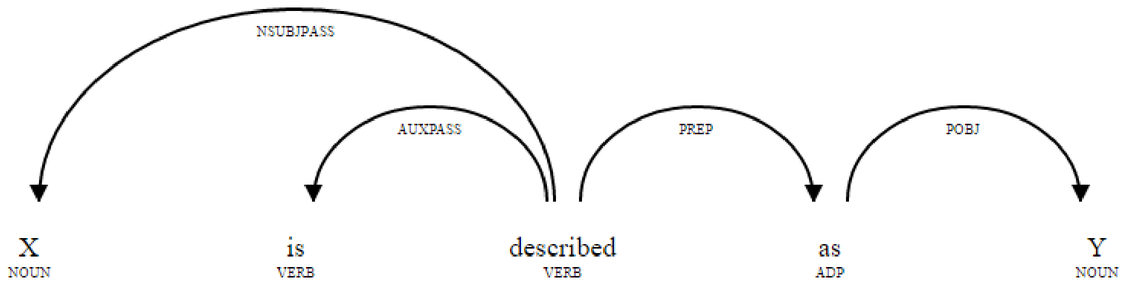
- Paths are generalized – a word might be replaced by:
  - its POS tag
  - a wild card
  - its ontological type



Needed: better path representation

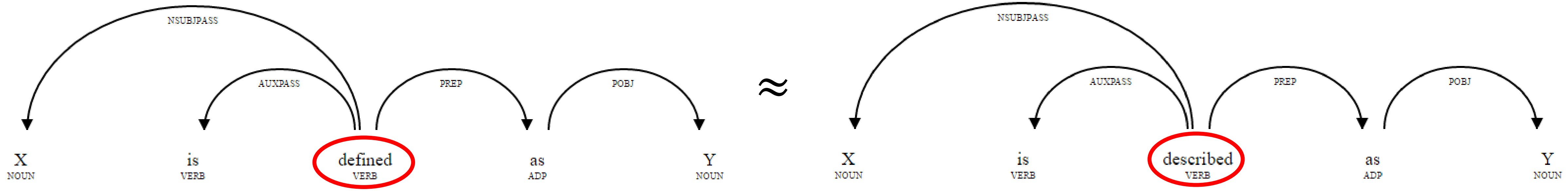
- Idea: learn semantic generalizations of paths





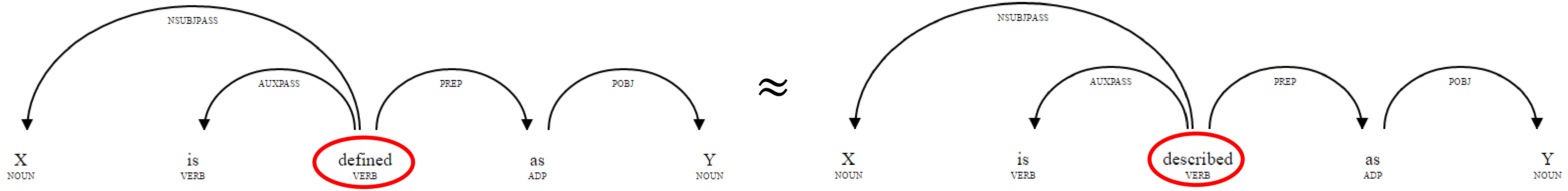
Needed: better path representation

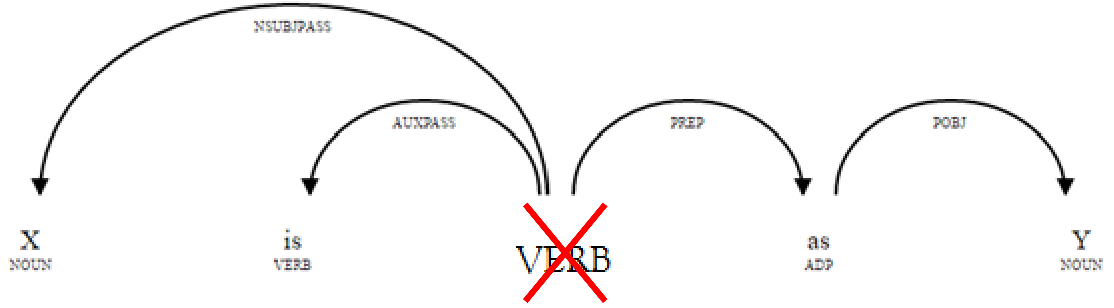
- Idea: learn semantic generalizations of paths



Needed: better path representation

- Idea: learn semantic generalizations of paths

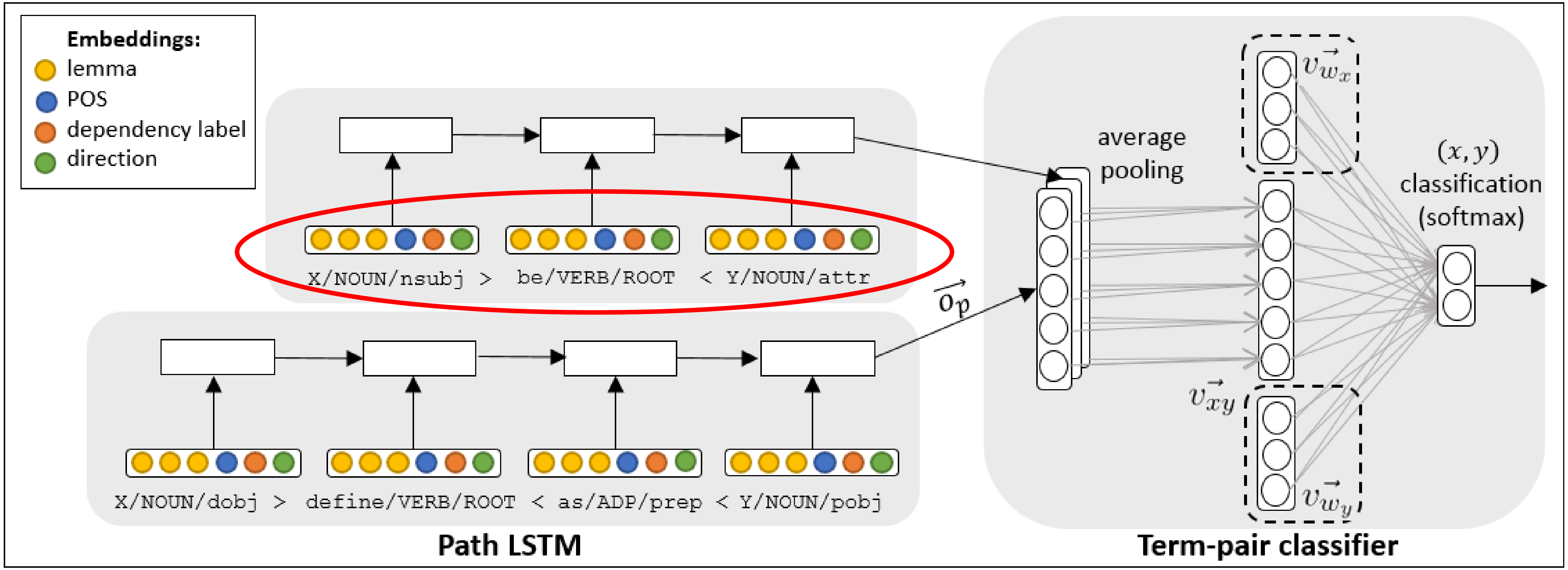




Outline – Hypernymy Detection

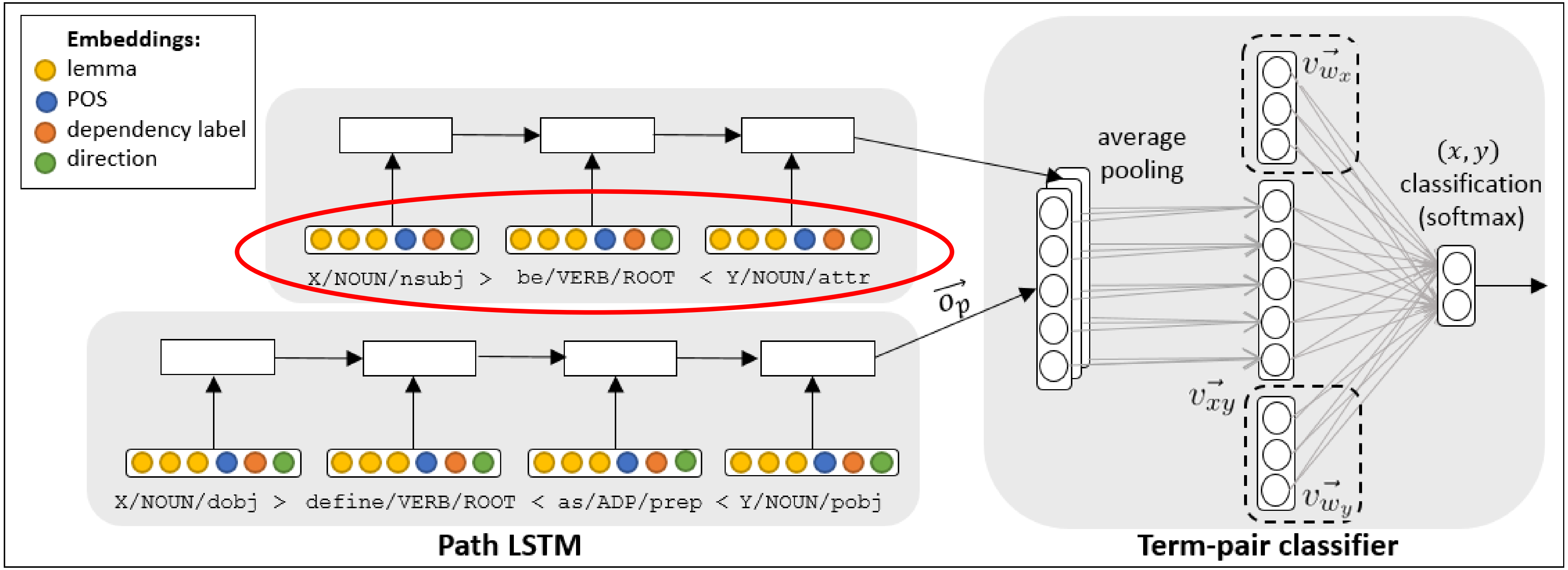
- Prior Methods
- Our Method
- Evaluation

LSTM-based hypernymy detection



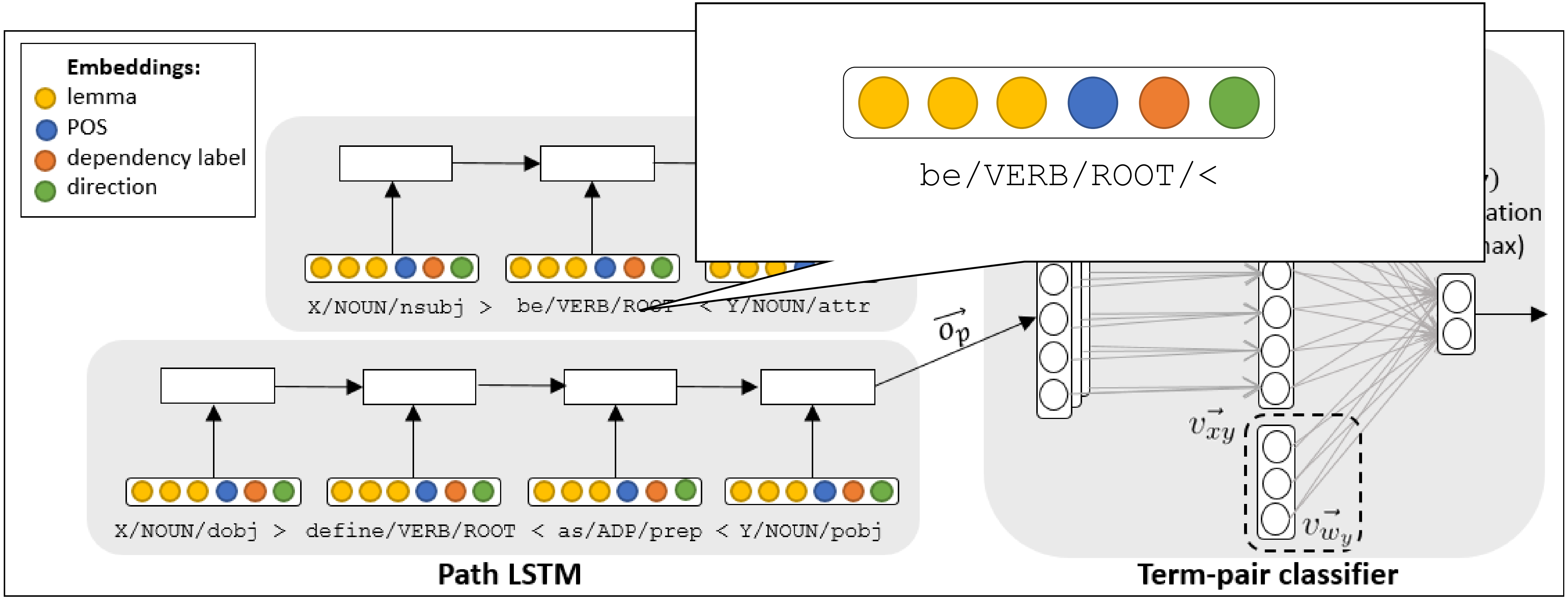
- Process each path edge-by-edge, using an LSTM

LSTM-based hypernymy detection



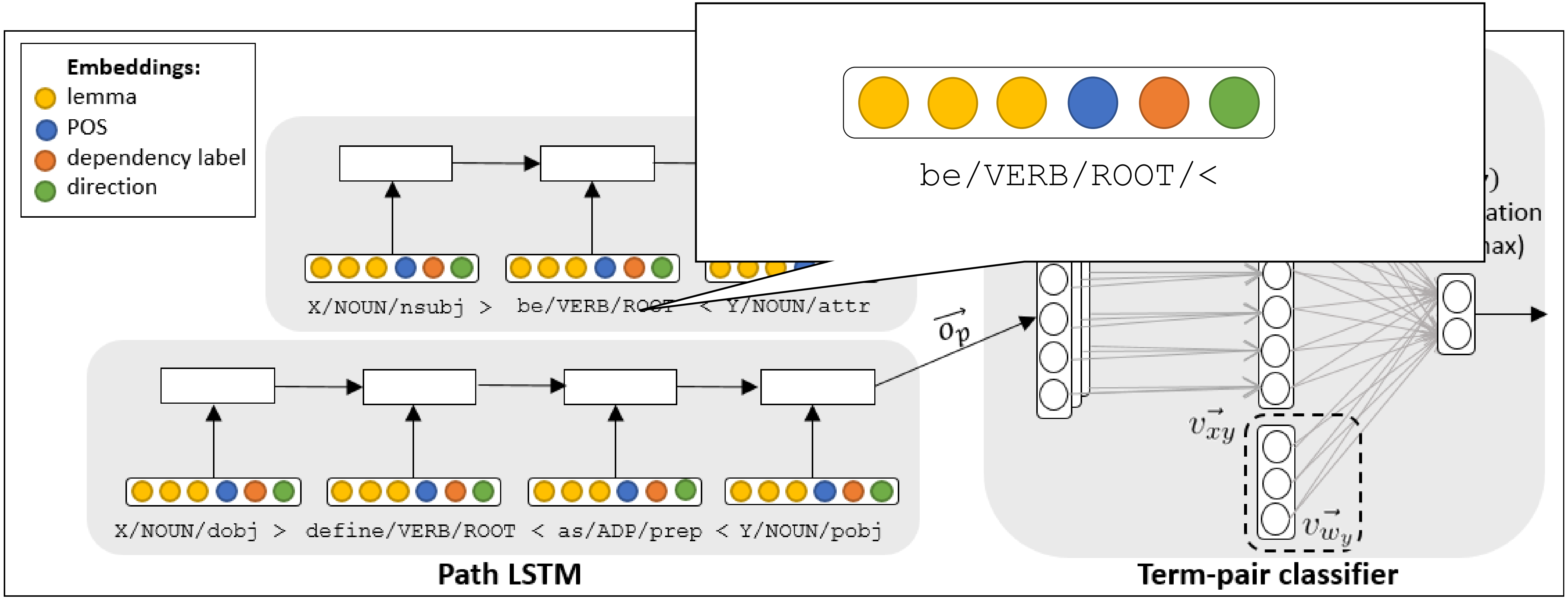
- Process each path edge-by-edge, using an LSTM
  - The encoder may focus on edges that are more informative for the classification task, while ignoring others

LSTM-based hypernymy detection



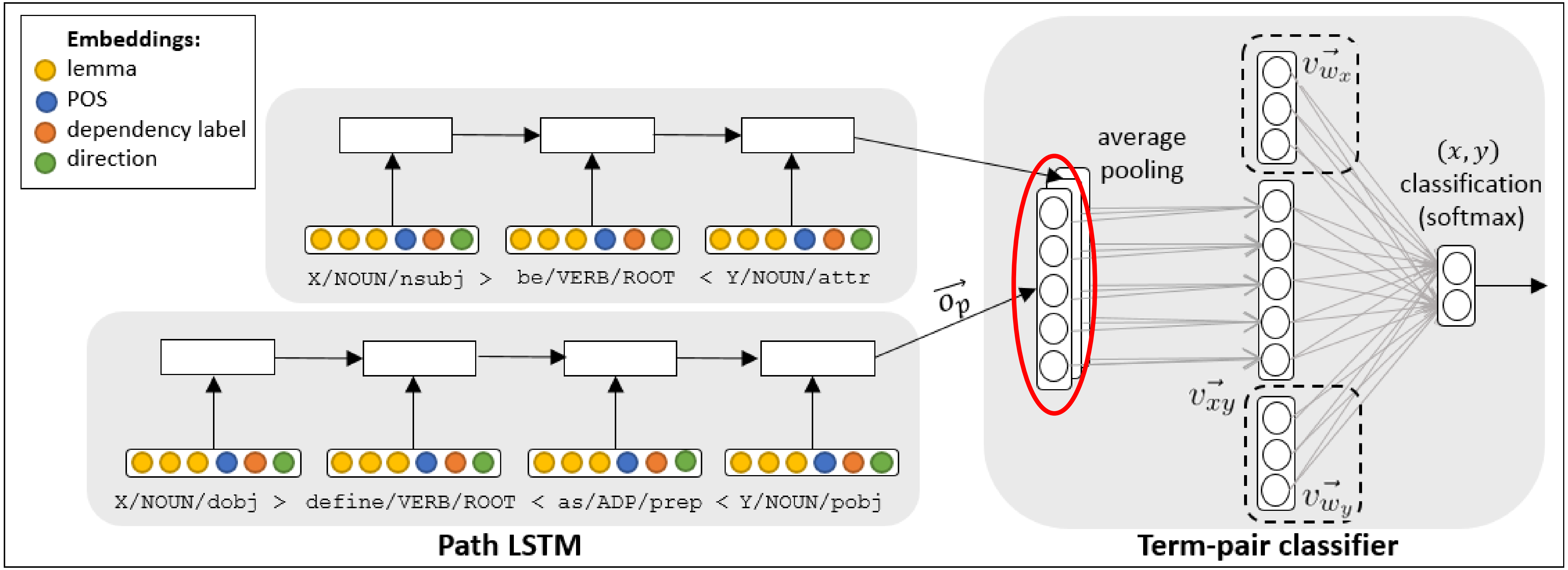
- Represent each edge as a concatenation of:
  - Lemma vector
  - Part-of-speech vector
  - Dependency label vector
  - Direction vector

LSTM-based hypernymy detection



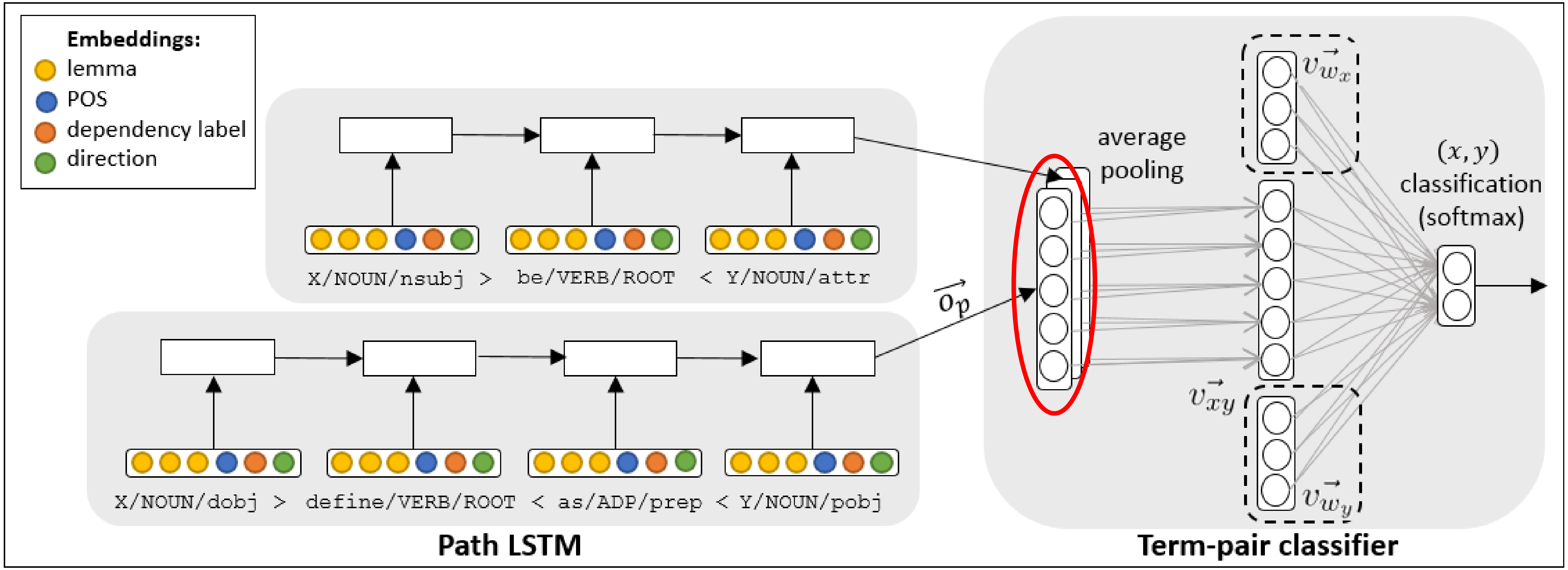
- Represent each edge as a concatenation of:
  - Lemma vector
  - Part-of-speech vector
  - Dependency label vector
  - Direction vector
- Learn embeddings of each component

LSTM-based hypernymy detection



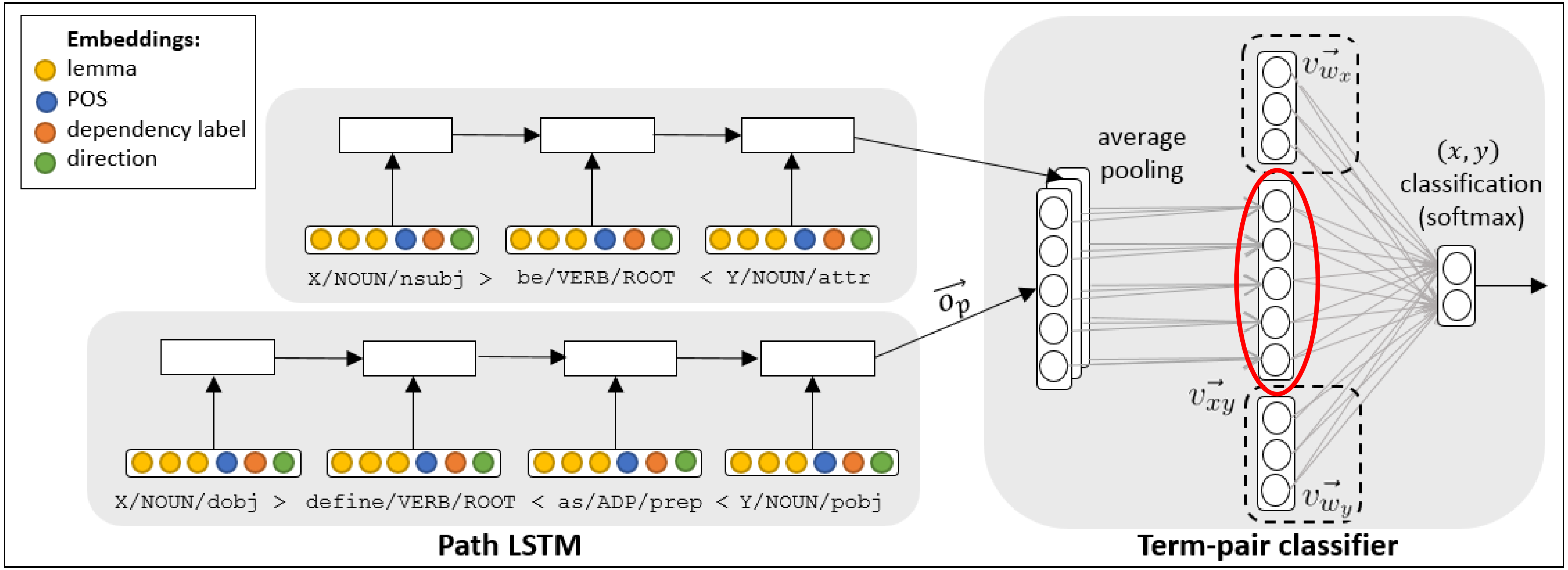
- Use the LSTM output as the path vector

LSTM-based hypernymy detection



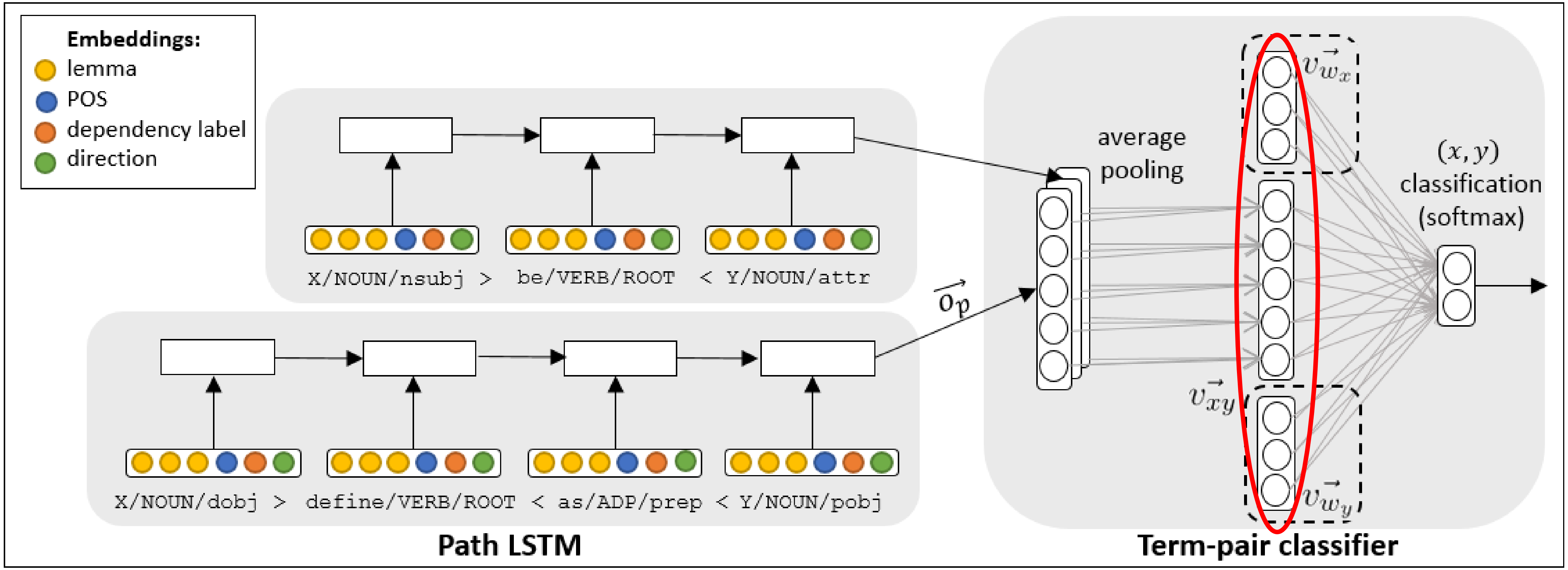
- Use the LSTM output as the path vector
- Each term-pair has multiple paths

LSTM-based hypernymy detection



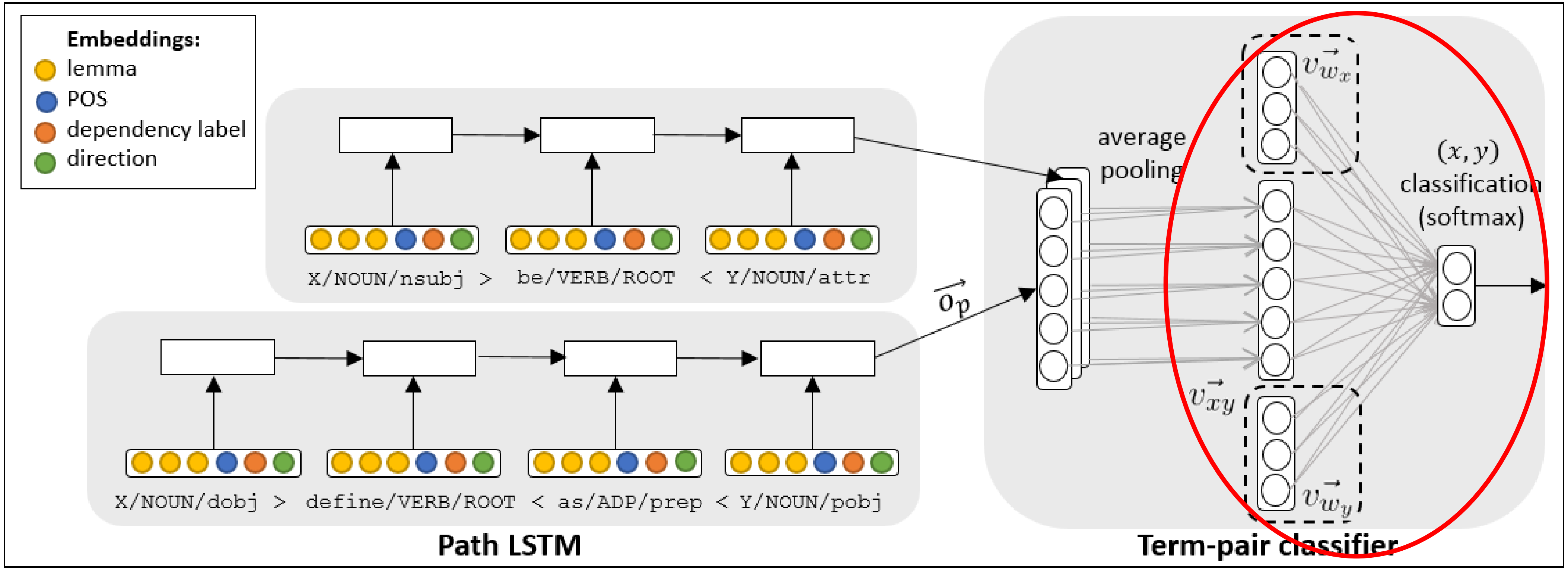
- Use the LSTM output as the path vector
- Each term-pair has multiple paths
  - Compute the averaged path embedding

LSTM-based hypernymy detection



- Each pair (x, y) is represented using the concatenation of:
  - x's embedding vector
  - the averaged path vector
  - y's embedding vector

LSTM-based hypernymy detection



- This vector is used as the input of a network that predicts whether y is a hypernym of x

Outline – Hypernymy Detection

- Prior Methods
- Our Method
- Evaluation

Experimental Settings

- Dataset:
  - Positive instances - hypernymy relations from resources:

resource	**relations**
WordNet	instance hypernym, hypernym
DBPedia	type
Wikidata	subclass of, instance of
Yago	subclass of

  - Negative instances – other relations from these resources
  - Filtering: pairs must occur in the corpus in at least 2 different paths
  - Similar to Snow et al. (2004)

Experimental Settings

- Train / Test / Validation split:
  - Random (70% - 25% - 5%)
  - Lexical:
    - Avoiding lexical memorization (Levy et al., 2015)
    - Distinct vocabulary in each set

Results

		random split			lexical split		
method		**precision**	**recall**	$\boldsymbol{F_1}$	**precision**	**recall**	$\boldsymbol{F_1}$
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
	LSTM (this paper)	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	SLQS (Santus et al., 2014)	0.246	0.213	0.228	0.270	0.222	0.243
	Best supervised (concatenation)	0.901	0.637	0.746	0.754	0.551	0.637
Combined	LSTM-Integrated (this paper)	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

Results

		random split			lexical split		
method		**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
	LSTM (this paper)	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	SLQS (Santus et al., 2014)	0.246	0.213	0.228	0.270	0.222	0.243
	Best supervised (concatenation)	0.901	0.637	0.746	0.754	0.551	0.637
Combined	LSTM-Integrated (this paper)	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

- Path-based:
  - Our method outperforms the baselines

Results

		random split			lexical split		
method		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
	LSTM (this paper)	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	SLQS (Santus et al., 2014)	0.246	0.213	0.228	0.270	0.222	0.243
	Best supervised (concatenation)	0.901	0.637	0.746	0.754	0.551	0.637
Combined	LSTM-Integrated (this paper)	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

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

Results

		random split			lexical split		
method		**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
	LSTM (this paper)	0.811	0.716	0.761	0.691	**0.632**	0.660
Distributional	SLQS (Santus et al., 2014)	0.246	0.213	0.228	0.270	0.222	0.243
	Best supervised (concatenation)	0.901	0.637	0.746	0.754	0.551	0.637
Combined	LSTM-Integrated (this paper)	**0.913**	**0.890**	**0.901**	**0.809**	0.617	**0.700**

- Path-based:
  - Our method outperforms the baselines
  - The generalizations yield improved recall
- The combined method outperforms both path-based and distributional methods

Analysis – Path Representation

- Snow's method finds certain common paths:

  X company is a Y

  X ltd is a Y

Analysis – Path Representation

- 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

- 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

- Our method makes fine-grained generalizations:

    X (association|co.|company|corporation| foundation|group|inc.|international|limited|ltd.) is a Y

Future Work

The Next Challenge

- Recognizing lexical inferences within context:

The Next Challenge

- Recognizing lexical inferences within context:

  - Detecting the correct sense of the term (e.g. *apple*) within the given context

The Next Challenge

- Recognizing lexical inferences within context:

  - Detecting the correct sense of the term (e.g. *apple*) within the given context
  - Basing the entailment decision on the sentence and the semantic relation:
    - I ate an *apple* => I ate a *fruit*
    - I hate *fruit* => I hate *apples*

First step: fine-grained semantic relations

- Idea: extend our hypernymy-detection method to support multi-class classification

First step: fine-grained semantic relations

- Idea: extend our hypernymy-detection method to support multi-class classification
  - e.g. WordNet relations: hypernym, meronym, antonym, synonym, …

First step: fine-grained semantic relations

- Idea: extend our hypernymy-detection method to support multi-class classification
  - e.g. WordNet relations: hypernym, meronym, antonym, synonym, …
  - Natural logic relations (MacCartney and Manning, 2007):

$\equiv$	Equivalence	is the same as
$\sqsubset$	Forward Entailment	is more specific than
$\sqsupset$	Reverse Entailment	is more general than
$\hat{}$	Negation	is the exact opposite of
$\vert$	Alternation	is mutually exclusive with
$\sim$	Other-Related	is related in some other way to
$\#$	Independence	is not related to

First step: fine-grained semantic relations

- Idea: extend our hypernymy-detection method to support multi-class classification
  - e.g. WordNet relations: hypernym, meronym, antonym, synonym, …
  - Natural logic relations (MacCartney and Manning, 2007):

$\equiv$	Equivalence	is the same as
$\sqsubset$	Forward Entailment	is more specific than
$\sqsupset$	Reverse Entailment	is more general than
$\hat{}$	Negation	is the exact opposite of
$\mid$	Alternation	is mutually exclusive with
$\sim$	Other-Related	is related in some other way to
$\#$	Independence	is not related to

  - BLESS relations: coord, mero, hyper, attr, event, random

Thanks!

References

[1] Vered Shwartz, Omer Levy, Ido Dagan, and Jacob Goldberger. Learning to Exploit Structured Resources for Lexical Inference. CoNLL 2015.

[2] Zellig S. Harris *Distributional structure.* Word. 1954.

[3] Julie Weeds and David Weir. *A general framework for distributional similarity.* EMNLP 2003.

[4] Lili Kotlerman et al. *Directional distributional similarity for lexical inference*. Natural Language Engineering 16.04: 359-389. 2010.

[5] Enrico Santus et al. *Chasing Hypernyms in Vector Spaces with Entropy*. EACL 2014.

[6] Laura Rimell. *Distributional Lexical Entailment by Topic Coherence*. EACL 2014.

[7] Yoshua Bengio et al., *A neural probabilistic language model*, The Journal of Machine Learning Research, 2003.

[8] Tomas Mikolov et. al *Efficient estimation of word representations in vector space*. CoRR, 2013.

[9] Jeffrey Pennington et al. *GloVe: Global Vectors for Word Representation*. EMNLP 2014.

[10] Marco Baroni et al. *Entailment above the word level in distributional semantics*. EACL 2012.

[11] Stephen Roller et al. *Inclusive yet selective: Supervised distributional hypernymy detection*. COLING 2014.

[12] Ruiji Fu et al. *Learning semantic hierarchies via word embeddings*. ACL 2014.

[13] Julie Weeds et al. *Learning to distinguish hypernyms and co-hyponyms*. COLING 2014.

[14] Omer Levy et al. *Do supervised distributional methods really learn lexical inference relations?* NAACL 2015.

[15] Marti A. Hearst *Automatic acquisition of hyponyms from large text corpora*. ACL, 1992.

[16] Rion Snow et al. *Learning syntactic patterns for automatic hypernym discovery.* Advances in Neural Information Processing Systems 17. 2004.

[17] Ndapandula Nakashole et al. *PATTY: A taxonomy of relational patterns with semantic types.* EMNLP 2012.

[18] Bill MacCartney and Christopher D Manning. *Natural logic for textual inference*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. ACL 2007.