

Acquiring Lexical Semantic Knowledge

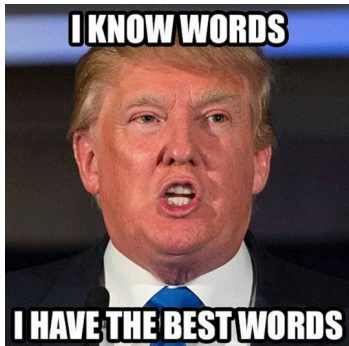
Vered Shwartz

Natural Language Processing Lab, Bar-Ilan University

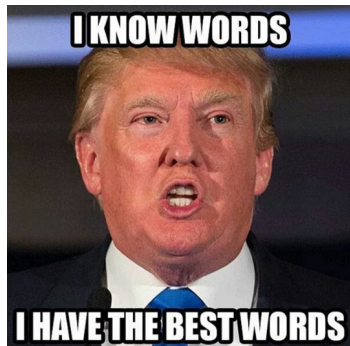
Stanford NLP Seminar, May 24, 2018



What is “lexical knowledge”?

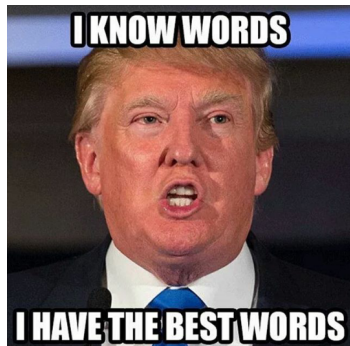


What is “lexical knowledge”?



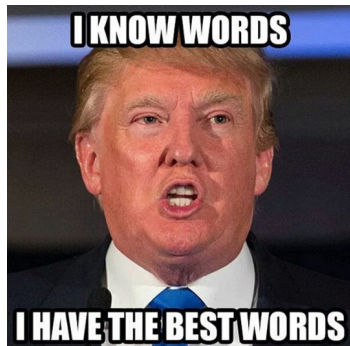
- Knowledge about lexical items (words, MWEs)

What is “lexical knowledge”?



- Knowledge about lexical items (words, MWEs)
- How do they **relate** to each other?

What is “lexical knowledge”?



- Knowledge about lexical items (words, MWEs)
- How do they **relate** to each other?
- Helpful for dealing with lexical variability in NLP applications

Example Application - 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

Huntsville is a *meronym* of Alabama, **Mississippi** is not.

Word Embeddings

(are not the solution for any problem)

- Provide semantic representations of words

Word Embeddings

(are not the solution for any problem)

- Provide semantic representations of words
- Commonly used across NLP applications with great success

Word Embeddings

(are not the solution for any problem)

- Provide semantic representations of words
- Commonly used across NLP applications with great success
- Pre-trained / learned / fine-tuned for a specific application

Word Embeddings

(are not the solution for any problem)

- Provide semantic representations of words
- Commonly used across NLP applications with great success
- Pre-trained / learned / fine-tuned for a specific application
- **Common claim:**
 - Word embeddings are all you need for lexical semantics

Word Embeddings

(are not the solution for any problem)

- Provide semantic representations of words
- Commonly used across NLP applications with great success
- Pre-trained / learned / fine-tuned for a specific application
- **Common claim:**
 - Word embeddings are all you need for lexical semantics
- **Reality:**
 - They are great in capturing general semantic relatedness

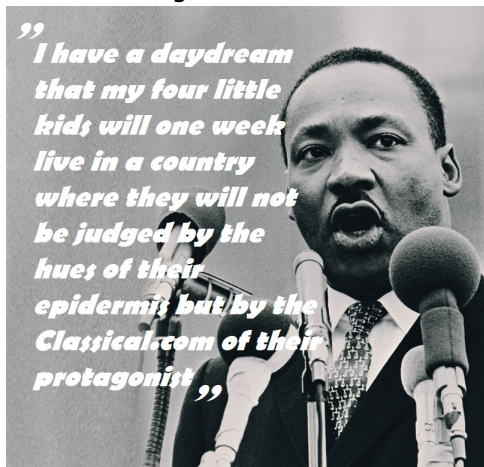
Word Embeddings

(are not the solution for any problem)

- Provide semantic representations of words
- Commonly used across NLP applications with great success
- Pre-trained / learned / fine-tuned for a specific application
- **Common claim:**
 - Word embeddings are all you need for lexical semantics
- **Reality:**
 - They are great in capturing general semantic relatedness
 - ...but they mix all semantic relations together!

Word Embeddings

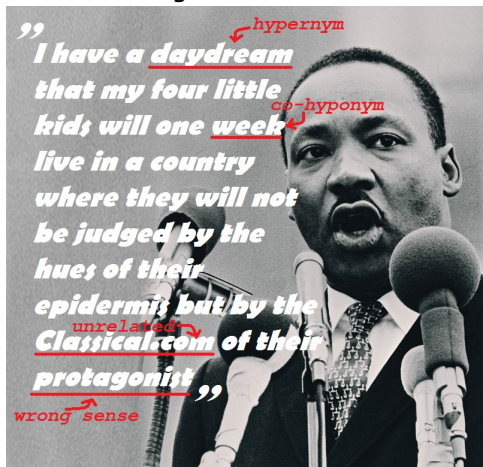
- To illustrate, take famous texts and replace nouns with their word2vec neighbours:¹



¹More examples here: <https://goo.gl/LJHzbi>

Word Embeddings

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



¹More examples here: <https://goo.gl/LJHzbi>

What's in this talk?

Recognizing Lexical Semantic Relations

Interpreting Noun Compounds

Recognizing Lexical Semantic Relations

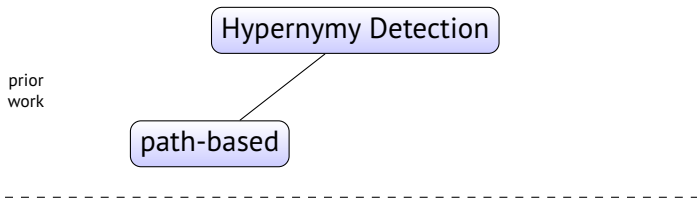
The Hypernymy Detection Task

- Hypernymy
 - The hyponym is a subclass of / instance of the hypernym
 - *(cat, animal), (Google, company)*

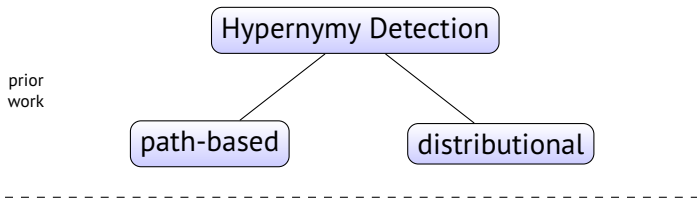
The Hypernymy Detection Task

- Hypernymy
 - The hyponym is a subclass of / instance of the hypernym
 - *(cat, animal)*, *(Google, company)*
- Given two terms, x and y , decide whether y is a hypernym of x
 - in some senses of x and y , e.g. *(apple, fruit)*, *(apple, company)*

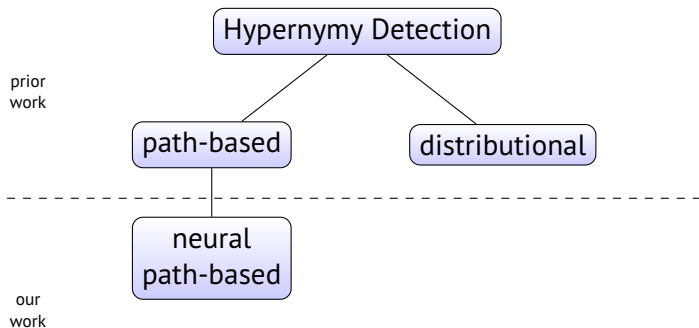
Corpus-based Hypernymy Detection



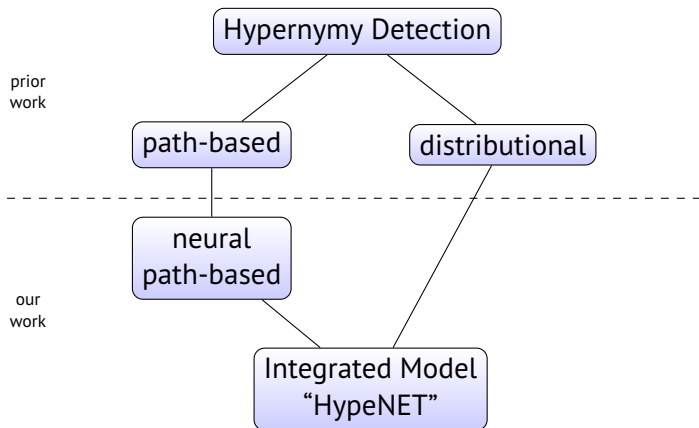
Corpus-based Hypernymy Detection



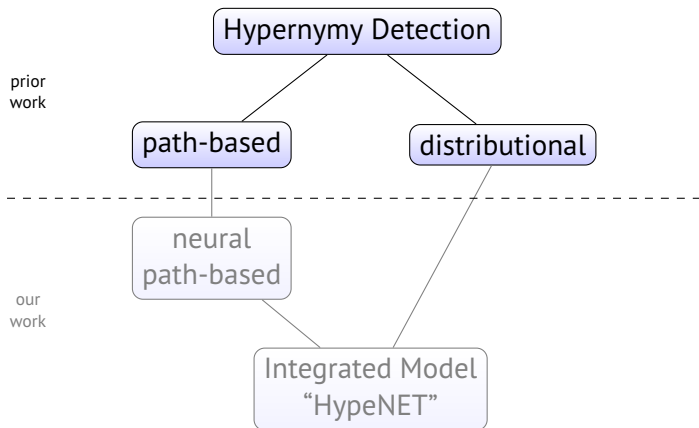
Corpus-based Hypernymy Detection



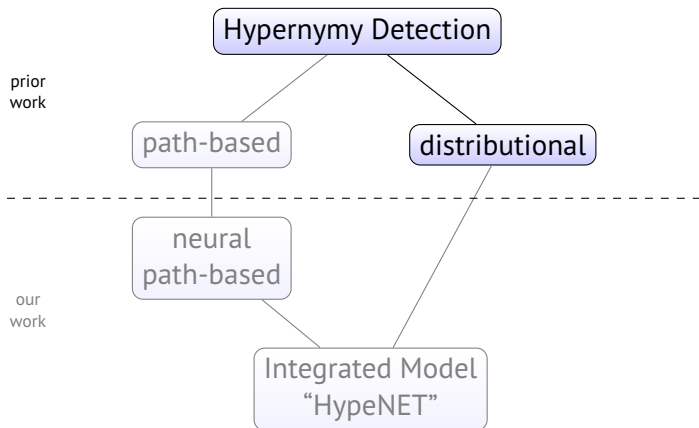
Corpus-based Hypernymy Detection



Prior Methods



Distributional Approach



Supervised Distributional Methods

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

Supervised Distributional Methods

- Recognize the relation between words based on their *separate* occurrences in the corpus
- Train a classifier to predict hypernymy using the terms' embeddings:
 - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
 - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]

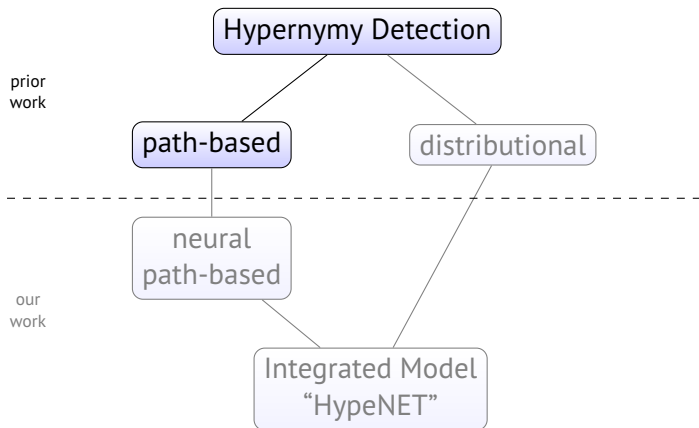
Supervised Distributional Methods

- Recognize the relation between words based on their *separate* occurrences in the corpus
- Train a classifier to predict hypernymy using the terms' embeddings:
 - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
 - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- Achieved very good results on common hypernymy detection / semantic relation classification datasets

Supervised Distributional Methods

- Recognize the relation between words based on their *separate* occurrences in the corpus
- Train a classifier to predict hypernymy using the terms' embeddings:
 - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
 - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- Achieved very good results on common hypernymy detection / semantic relation classification datasets
- [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*

Path-based Approach



Path-based Approach

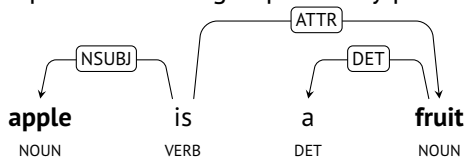
- Recognize the relation between x and y based on their *joint* occurrences in the corpus

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*

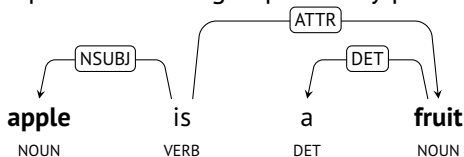
Path-based Approach

- 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:



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:



- [Snow et al., 2004]: logistic regression classifier, dependency paths as sparse features

0	0	...	58	0	...	97	0	...	0
---	---	-----	----	---	-----	----	---	-----	---

↑ ↑
X and other Y *such Y as X*

Path-based Approach Issues

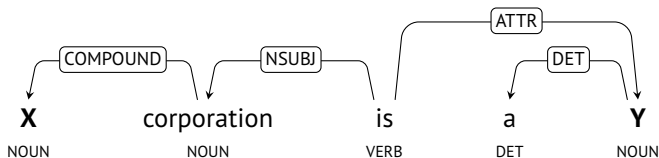
- The feature space is too sparse:

Path-based Approach Issues

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

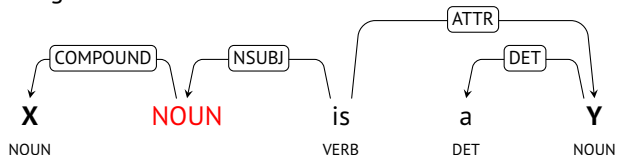
Path-based Approach Issues

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



Path-based Approach Issues

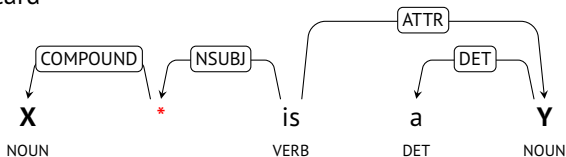
- The feature space is too sparse:
 - Similar paths share no information:
 - X inc. is a Y
 - X group is a Y
 - X organization is a Y
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
 - its POS tag



Path-based Approach Issues

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

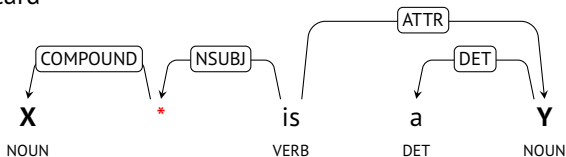
- a wild-card



Path-based Approach Issues

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

- a wild-card

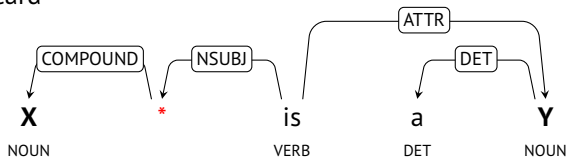


- Some of these generalizations are too general:
 - X is defined as Y $\approx$ X is described as Y via X is VERB as Y

Path-based Approach Issues

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

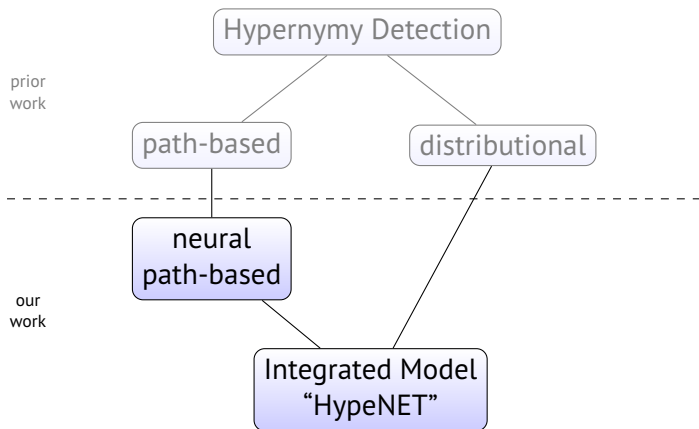
- a wild-card



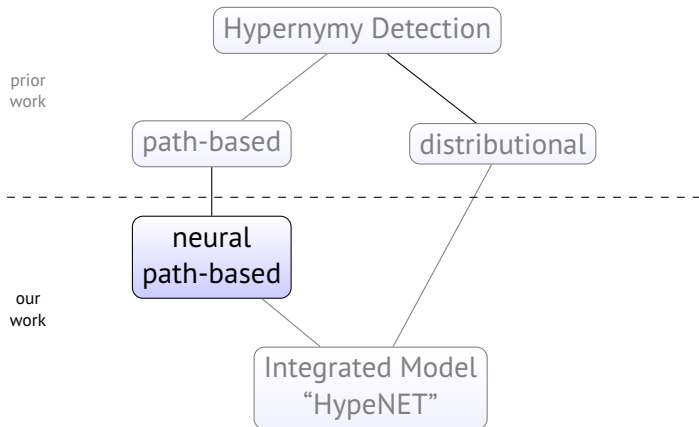
- Some of these generalizations are too general:
 - X is defined as $Y \approx$ X is described as Y via X is VERB as Y
 - X is defined as $Y \neq$ X is rejected as Y

HypeNET: Integrated Path-based and Distributional Method

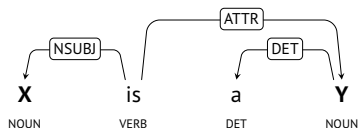
[Shwartz et al., 2016]



First Step: Improving Path Representation



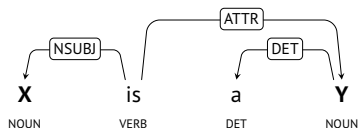
Path Representation



1. Split each path to edges, each edge consists of 4 components:

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

Path Representation

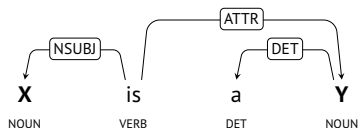


1. Split each path to edges, each edge consists of 4 components:

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

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

Path Representation



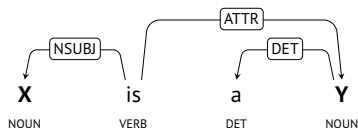
1. Split each path to edges, each edge consists of 4 components:

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

- 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:

[dependent lemma ; dependent POS ; dependency label ; direction]

Path Representation



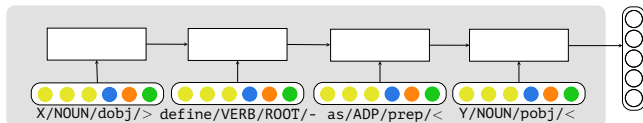
1. Split each path to edges, each edge consists of 4 components:

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

- 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:

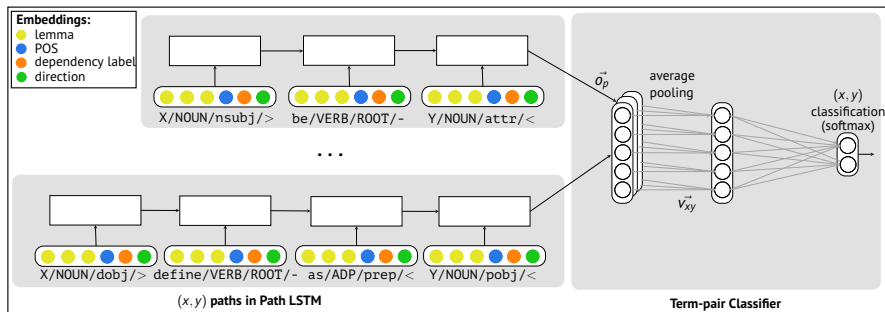
[dependent lemma ; dependent POS ; dependency label ; direction]

2. Feed the edges sequentially to an LSTM, use the last output vector as the path embedding:

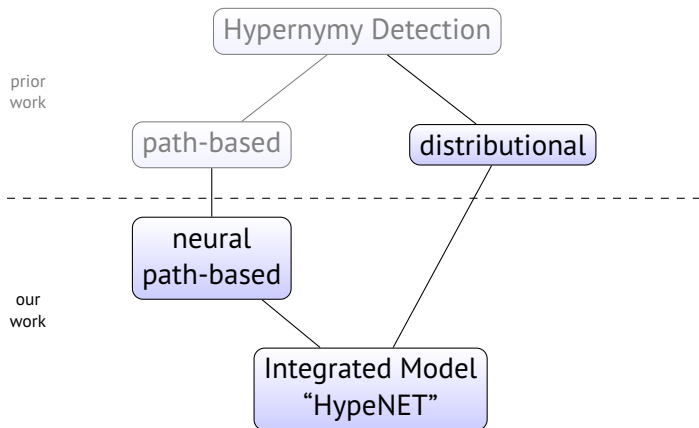


Term-pair Classification

- 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):

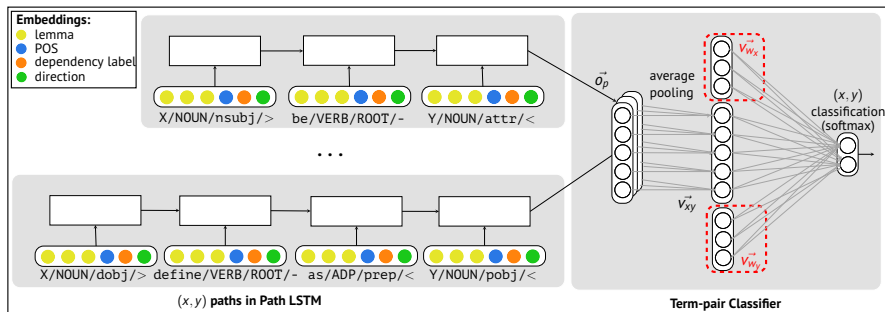


Second Step: Integrating Distributional Information



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 knowledge resources

	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 knowledge resources

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

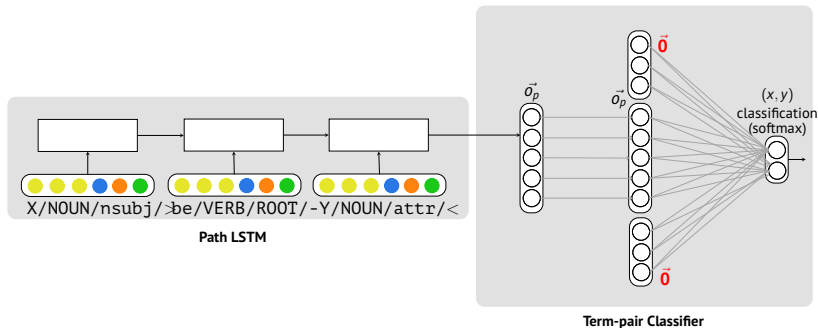
- The integrated method substantially outperforms both path-based and distributional methods

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 $\text{softmax}(W \cdot [\vec{o}, \vec{o}_p, \vec{o}])[1]$

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 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

...

LexNET - Multiple Semantic Relation Classification

[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations

LexNET - Multiple Semantic Relation Classification

[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations
- LexNET outperforms individual path-based and distributional methods

LexNET - Multiple Semantic Relation Classification

[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations
- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is prominent when:

LexNET - Multiple Semantic Relation Classification

[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations
- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is prominent when:
 - Lexical memorization is disabled (lexical split)

LexNET - Multiple Semantic Relation Classification

[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations
- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is prominent when:
 - Lexical memorization is disabled (lexical split)
 - x or y are polysemous, e.g. *mero:(piano, key)*.

LexNET - Multiple Semantic Relation Classification

[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations
- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is prominent when:
 - Lexical memorization is disabled (lexical split)
 - x or y are polysemous, e.g. *mero:(piano, key)*.
 - the relation is not prototypical, e.g. *event:(cherry, pick)*.

LexNET - Multiple Semantic Relation Classification

[Shwartz and Dagan, 2016a, Shwartz and Dagan, 2016b]

- Application of HypeNET for multiple relations
- LexNET outperforms individual path-based and distributional methods
- Path-based contribution over distributional info is prominent when:
 - Lexical memorization is disabled (lexical split)
 - 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)*.

Interpreting Noun Compounds

Noun Compounds

- Noun-compounds hold an implicit semantic relation between the head and its modifier(s).

Noun Compounds

- Noun-compounds hold an implicit semantic relation between the head and its modifier(s).
 - *apple cake*: *cake made of apples*

Noun Compounds

- Noun-compounds hold an implicit semantic relation between the head and its modifier(s).
 - *apple cake*: cake made of apples
 - *birthday cake*: cake eaten on a birthday

Noun Compounds

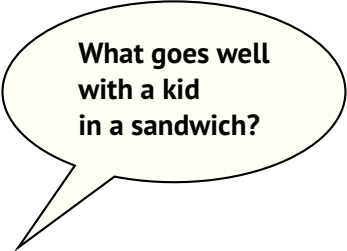
- Noun-compounds hold an implicit semantic relation between the head and its modifier(s).
 - *apple cake*: cake made of apples
 - *birthday cake*: cake eaten on a birthday
- They are like “text compression devices” [Nakov, 2013]

Noun Compounds

- Noun-compounds hold an implicit semantic relation between the head and its modifier(s).
 - *apple cake*: cake made of apples
 - *birthday cake*: cake eaten on a birthday
- They are like “text compression devices” [Nakov, 2013]
- We’re pretty good in decompressing them!

We are good at Interpreting Noun-Compounds

We are good at Interpreting Noun-Compounds



**What goes well
with a kid
in a sandwich?**

Interpreting new Noun Compounds

- Noun-compounds are prevalent in English, but most are rare

Interpreting new Noun Compounds

- Noun-compounds are prevalent in English, but most are rare
- We easily interpret noun-compounds we've never seen before

Interpreting new Noun Compounds

- Noun-compounds are prevalent in English, but most are rare
- We easily interpret noun-compounds we've never seen before
- What is a “*parsley cake*”?

Interpreting new Noun Compounds

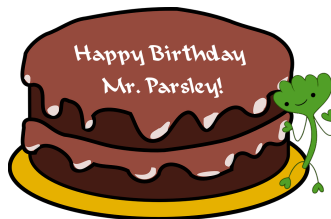
- Noun-compounds are prevalent in English, but most are rare
- We easily interpret noun-compounds we've never seen before
- What is a “*parsley cake*”?

1.

cake with/from parsley

(from <http://www.bazekalim.com>)

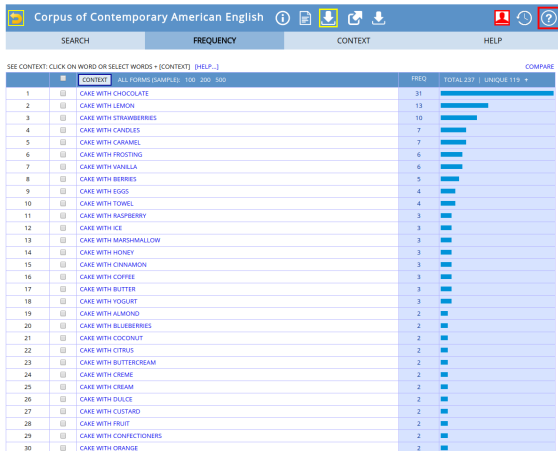
2.



cake for parsley

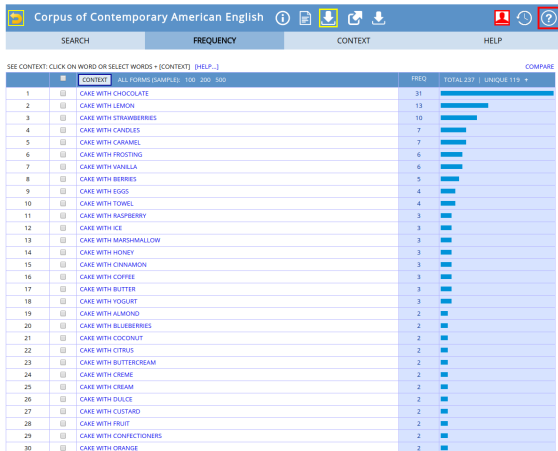
Interpreting new Noun Compounds

■ What can cake be made of?



Interpreting new Noun Compounds

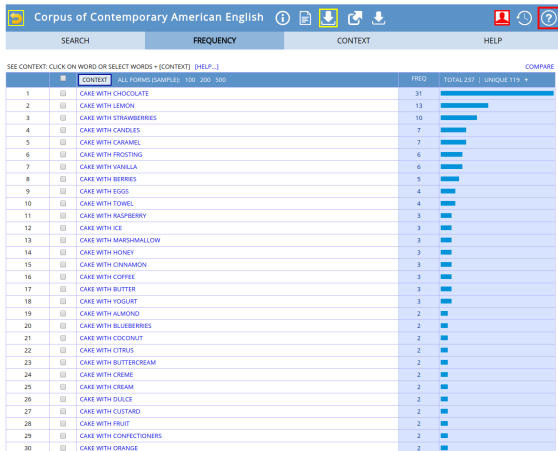
■ What can cake be made of?



■ Parsley (sort of) fits into this distribution

Interpreting new Noun Compounds

■ What can cake be made of?



- Parsley (sort of) fits into this distribution
- Similar to “selectional preferences” [Pantel et al., 2007]

We need Computers to Interpret Noun-Compounds

19:42 ... 🔔 🕒 📶 📶 🔋 42%



Add an event

create a morning meeting

Title

Day

Tomorrow

Time

Morning

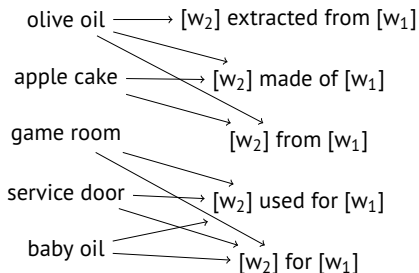


Noun-Compound Interpretation Tasks

- Compositionality Prediction
- **Noun-compound Paraphrasing**
- Noun-compound Classification

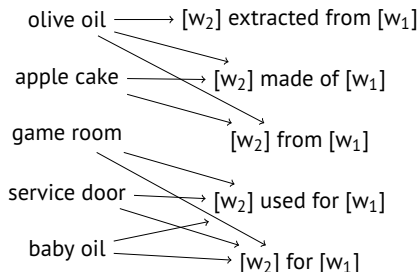
Noun-Compound Paraphrasing

- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



Noun-Compound Paraphrasing

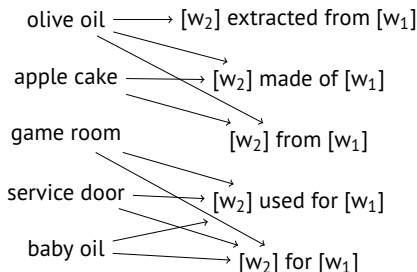
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



- SemEval 2013 task 4 [Hendrickx et al., 2013]:

Noun-Compound Paraphrasing

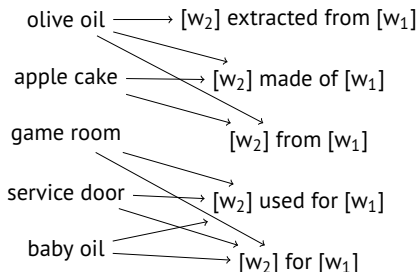
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



- SemEval 2013 task 4 [Hendrickx et al., 2013]:
 - Systems get a list of noun compounds

Noun-Compound Paraphrasing

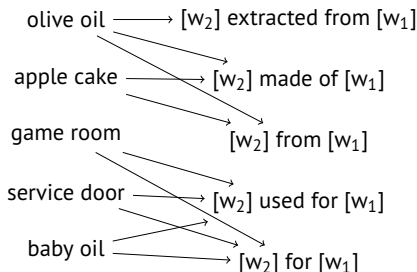
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



- SemEval 2013 task 4 [Hendrickx et al., 2013]:
 - Systems get a list of noun compounds
 - Extract paraphrases from free text

Noun-Compound Paraphrasing

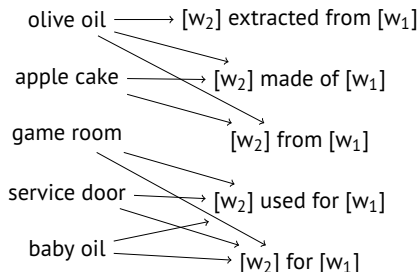
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



- SemEval 2013 task 4 [Hendrickx et al., 2013]:
 - Systems get a list of noun compounds
 - Extract paraphrases from free text
 - Rank them

Noun-Compound Paraphrasing

- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



- SemEval 2013 task 4 [Hendrickx et al., 2013]:
 - Systems get a list of noun compounds
 - Extract paraphrases from free text
 - Rank them
 - Evaluated for correlation with human judgments

Prior Work

Prior Methods

- Based on corpus occurrences of the constituents:
“cake made of apples”

Prior Methods

- Based on corpus occurrences of the constituents:
“cake made of apples”
- SemEval task participants extracted them from Google N-grams

Prior Methods

- Based on corpus occurrences of the constituents:
“cake made of apples”
- SemEval task participants extracted them from Google N-grams
- **Problems:**
 1. Many unseen NCs, no paraphrases in the corpus

Prior Methods

- Based on corpus occurrences of the constituents:
“cake made of apples”
- SemEval task participants extracted them from Google N-grams
- **Problems:**
 1. Many unseen NCs, no paraphrases in the corpus
 2. Many NCs with just a few paraphrases

Prior Methods

- Based on corpus occurrences of the constituents:
“*cake* made of *apples*”
- SemEval task participants extracted them from Google N-grams

- **Problems:**
 1. Many unseen NCs, no paraphrases in the corpus
 2. Many NCs with just a few paraphrases
- **Partial solutions:**
 1. [Van de Cruys et al., 2013]: generalize for unseen NCs with similar NCs, e.g. *pear tart* is similar to *apple cake*

Prior Methods

- Based on corpus occurrences of the constituents:
“*cake* made of *apples*”
- SemEval task participants extracted them from Google N-grams
- **Problems:**
 1. Many unseen NCs, no paraphrases in the corpus
 2. Many NCs with just a few paraphrases
- **Partial solutions:**
 1. [Van de Cruys et al., 2013]: generalize for unseen NCs with similar NCs, e.g. *pear tart* is similar to *apple cake*
 2. [Surtani et al., 2013]: learn “is-a” relations between paraphrases:
e.g. “[w₂] extracted from [w₁]” $\subset$ “[w₂] made of [w₁]”

Prior Methods

- Based on corpus occurrences of the constituents:
“cake made of apples”
- SemEval task participants extracted them from Google N-grams
- **Problems:**
 1. Many unseen NCs, no paraphrases in the corpus
 2. Many NCs with just a few paraphrases
- **Partial solutions:**
 1. [Van de Cruys et al., 2013]: generalize for unseen NCs with similar NCs, e.g. *pear tart* is similar to *apple cake*
 2. [Surtani et al., 2013]: learn “is-a” relations between paraphrases:
e.g. “[w₂] extracted from [w₁]” $\subset$ “[w₂] made of [w₁]”
- Our solution: multi-task learning to address both problems

Model

Multi-task Reformulation

- Previous approaches: predict a paraphrase for a given NC

Multi-task Reformulation

- Previous approaches: predict a paraphrase for a given NC
- Our model: multi-task learning problem

Multi-task Reformulation

- Previous approaches: predict a paraphrase for a given NC
- Our model: multi-task learning problem
- Training example $\{w_1 = \text{apple}, w_2 = \text{cake}, p = "[w_2] \text{ made of } [w_1]"\}$

Multi-task Reformulation

- Previous approaches: predict a paraphrase for a given NC
- Our model: multi-task learning problem
- Training example $\{w_1 = \text{apple}, w_2 = \text{cake}, p = "[w_2] \text{ made of } [w_1]"\}$
 1. Predict a paraphrase p for a given NC w_1w_2 :
What is the relation between *apple* and *cake*?

Multi-task Reformulation

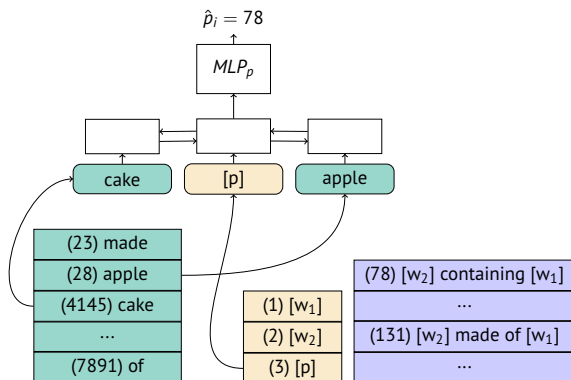
- Previous approaches: predict a paraphrase for a given NC
- Our model: multi-task learning problem
- Training example $\{w_1 = \text{apple}, w_2 = \text{cake}, p = "[w_2] \text{ made of } [w_1]"\}$
 1. Predict a paraphrase p for a given NC $w_1 w_2$:
What is the relation between *apple* and *cake*?
 2. Predict w_1 given a paraphrase p and w_2 :
What can *cake* be made of?

Multi-task Reformulation

- Previous approaches: predict a paraphrase for a given NC
- Our model: multi-task learning problem
- Training example $\{w_1 = \text{apple}, w_2 = \text{cake}, p = "[w_2] \text{ made of } [w_1]"\}$
 1. Predict a paraphrase p for a given NC $w_1 w_2$:
What is the relation between *apple* and *cake*?
 2. Predict w_1 given a paraphrase p and w_2 :
What can *cake* be made of?
 3. Predict w_2 given a paraphrase p and w_1 :
What can be made of *apple*?

Main Task (1): Predicting Paraphrases

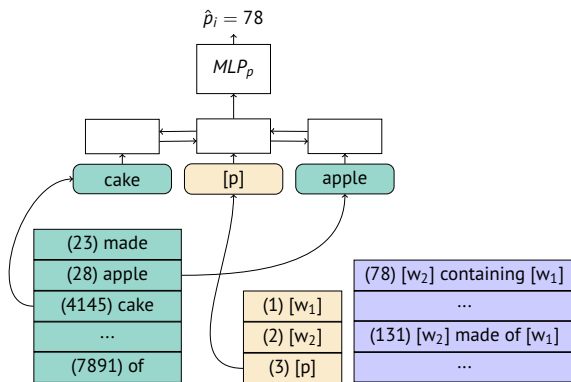
What is the relation between *apple* and *cake*?



- Encode placeholder [p] in “cake [p] apple” using biLSTM

Main Task (1): Predicting Paraphrases

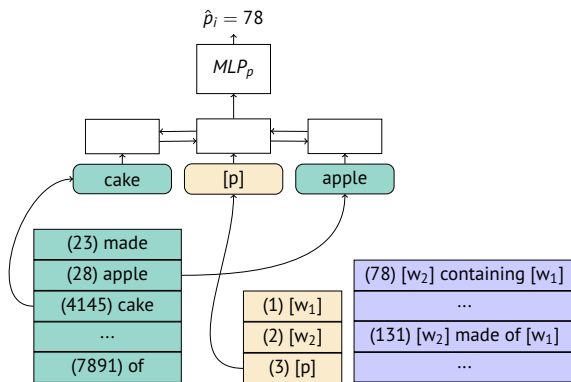
What is the relation between *apple* and *cake*?



- Encode placeholder $[p]$ in “cake $[p]$ apple” using biLSTM
- Predict an index in the paraphrase vocabulary

Main Task (1): Predicting Paraphrases

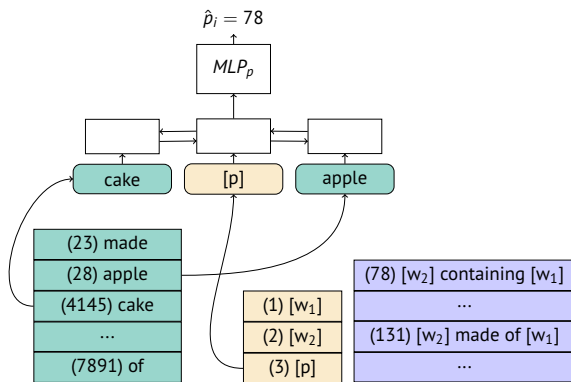
What is the relation between *apple* and *cake*?



- Encode placeholder $[p]$ in “cake $[p]$ apple” using biLSTM
- Predict an index in the paraphrase vocabulary
- Fixed word embeddings, learned placeholder embeddings

Main Task (1): Predicting Paraphrases

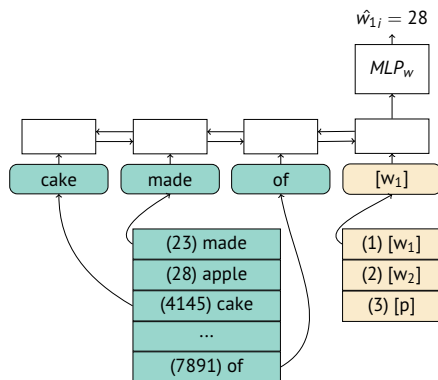
What is the relation between *apple* and *cake*?



- Encode placeholder [p] in “cake [p] apple” using biLSTM
- Predict an index in the paraphrase vocabulary
- Fixed word embeddings, learned placeholder embeddings
- (1) Generalizes NCs: *pear tart* expected to yield similar results

Helper Task (2): Predicting Missing Constituents

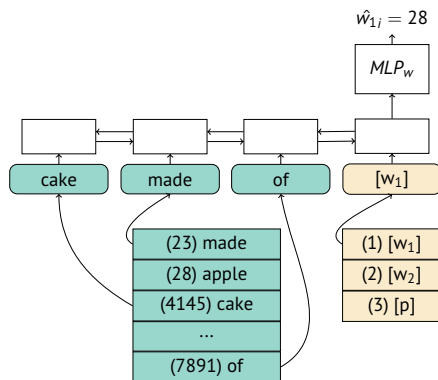
What can *cake* be made of?



- Encode placeholder in “cake made of [w₁]” using biLSTM

Helper Task (2): Predicting Missing Constituents

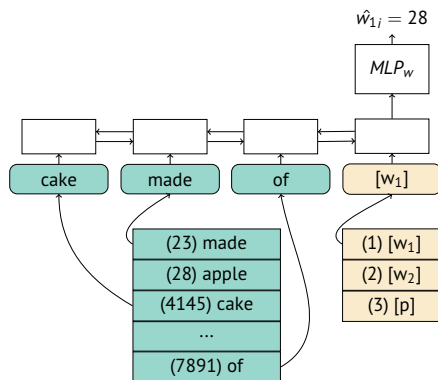
What can *cake* be made of?



- Encode placeholder in “cake made of [w₁]” using biLSTM
- Predict an index in the word vocabulary

Helper Task (2): Predicting Missing Constituents

What can *cake* be made of?



- Encode placeholder in “cake made of [w₁]” using biLSTM
- Predict an index in the word vocabulary
- (2) Generalizes paraphrases:
 “[w₂] containing [w₁]” expected to yield similar results

Training Data

- Collected from Google N-grams

Training Data

- Collected from Google N-grams
- Input:
 - Set of NCs
 - Templates of POS tags (e.g. “[w₂] verb prep [w₁]”)

Training Data

- Collected from Google N-grams
- Input:
 - Set of NCs
 - Templates of POS tags (e.g. “[w₂] verb prep [w₁]”)
- Weighting by frequency and length

Training Data

- Collected from Google N-grams
- Input:
 - Set of NCs
 - Templates of POS tags (e.g. “[w₂] verb prep [w₁]”)
- Weighting by frequency and length
- 136,609 instances

Evaluation: Paraphrasing

Model

- Predict top k paraphrases for each noun compound

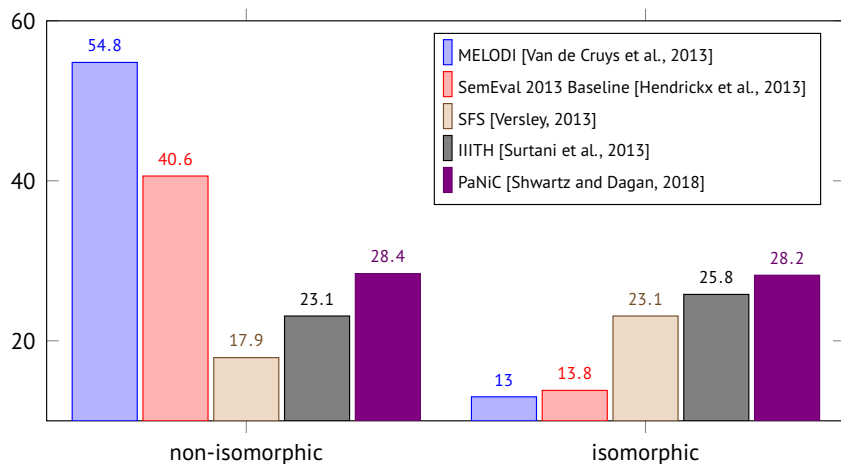
Model

- Predict top k paraphrases for each noun compound
- Learn to re-rank the paraphrases
 - to better correlate with human judgments

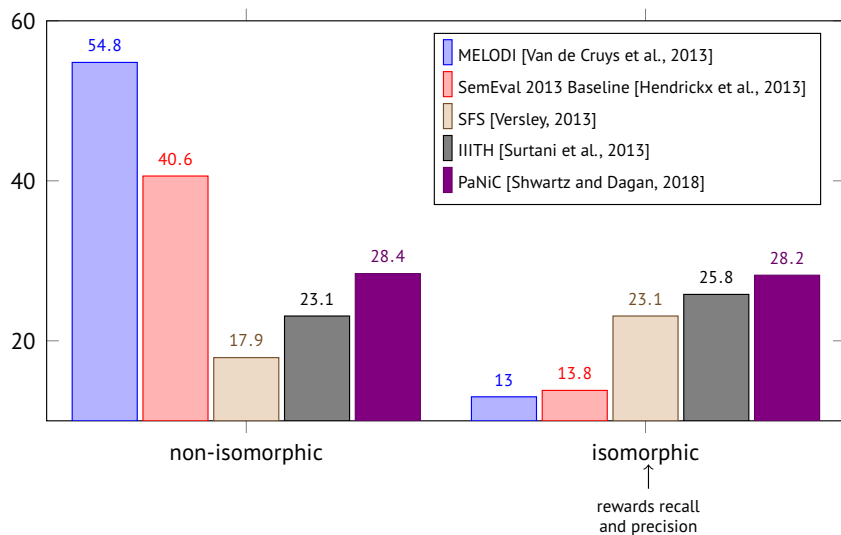
Model

- Predict top k paraphrases for each noun compound
- Learn to re-rank the paraphrases
 - to better correlate with human judgments
- SVM pair-wise ranking with the following features:
 - POS tags in the paraphrase
 - Prepositions in the paraphrase
 - Length
 - Special symbols
 - Similarity to predicted paraphrase

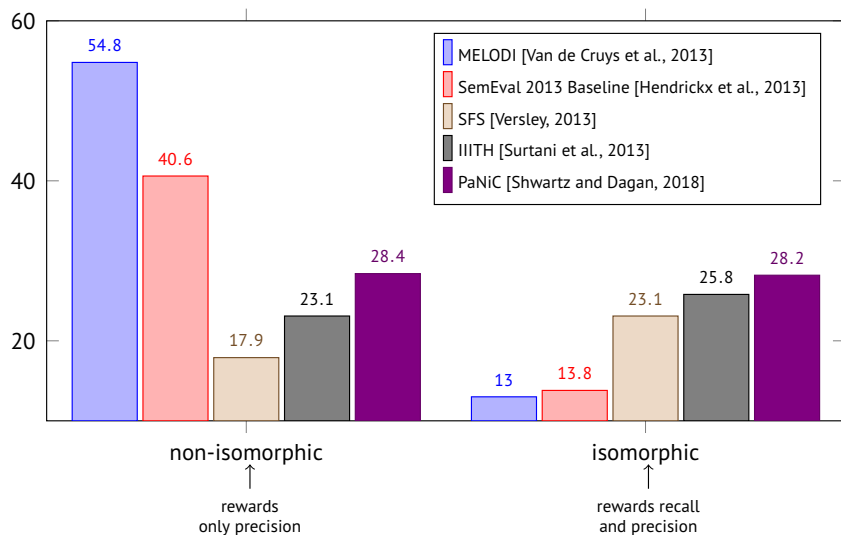
Results



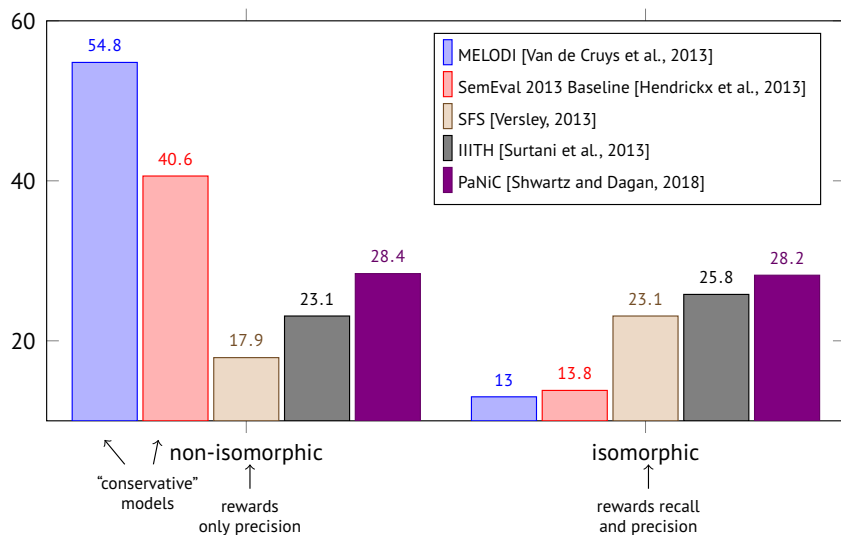
Results



Results



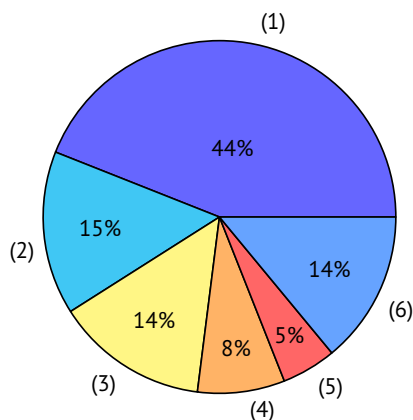
Results



Error Analysis

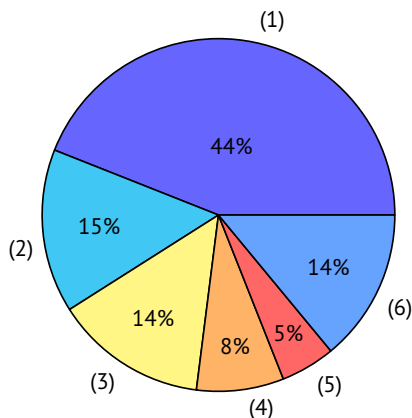
False Positive

1. Valid, missing from gold-standard
("discussion by group")



Error Analysis

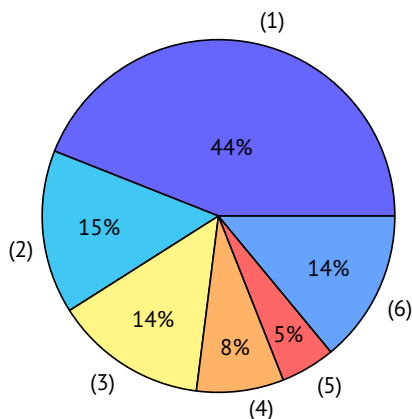
False Positive



1. Valid, missing from gold-standard
("discussion by group")
2. Too specific
("life of *women in* community")

Error Analysis

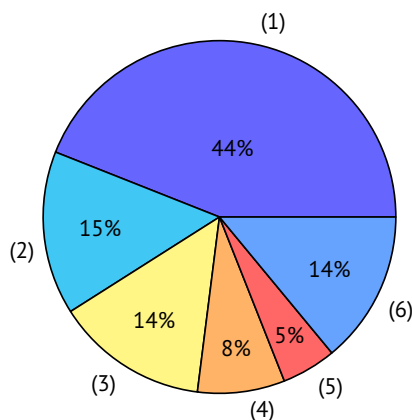
False Positive



1. Valid, missing from gold-standard
("discussion by group")
2. Too specific
("life *of women in* community")
3. Incorrect prepositions
E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby 's head" $\Rightarrow$ "oil from baby"

Error Analysis

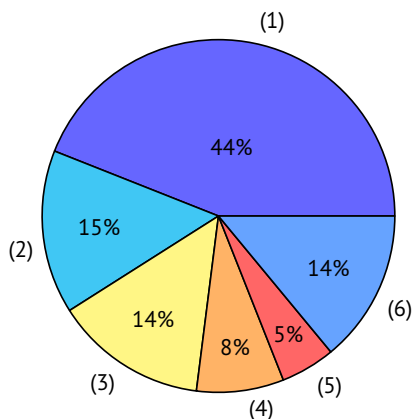
False Positive



1. Valid, missing from gold-standard
("discussion by group")
2. Too specific
("life *of women in* community")
3. Incorrect prepositions
E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby's head" $\Rightarrow$ "oil from baby"
4. Syntactic errors

Error Analysis

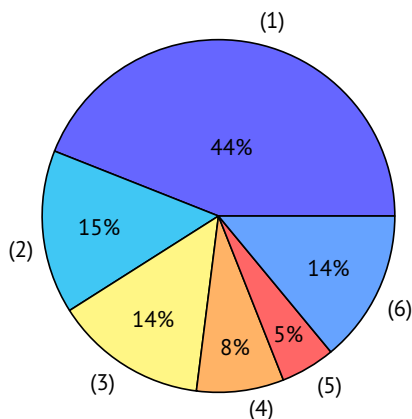
False Positive



1. Valid, missing from gold-standard
("discussion by group")
2. Too specific
("life *of women in* community")
3. Incorrect prepositions
E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby's head" $\Rightarrow$ "oil from baby"
4. Syntactic errors
5. Borderline grammatical
("force of coalition forces")

Error Analysis

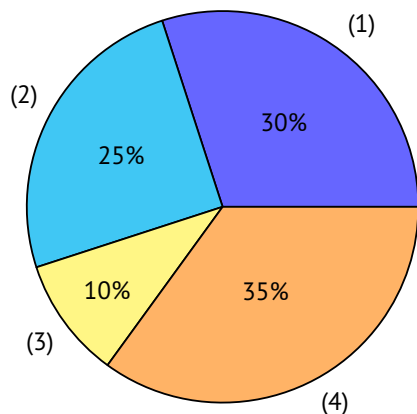
False Positive



1. Valid, missing from gold-standard
("discussion by group")
2. Too specific
("life *of women in* community")
3. Incorrect prepositions
E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby's head" $\Rightarrow$ "oil from baby"
4. Syntactic errors
5. Borderline grammatical
("force of coalition forces")
6. Other errors

Error Analysis

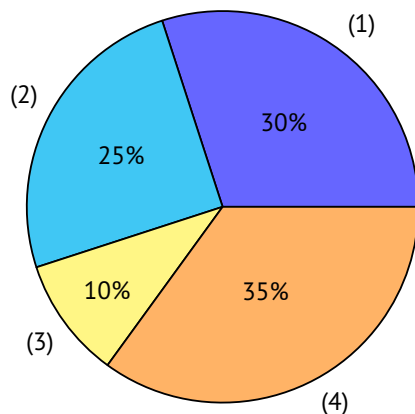
False Negative



1. Long paraphrase ($n > 5$)

Error Analysis

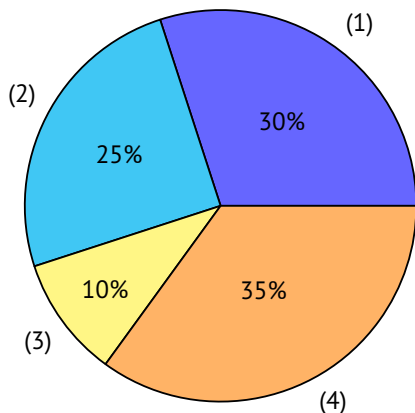
False Negative



1. Long paraphrase ($n > 5$)
2. Determiners
("mutation of **a** gene")

Error Analysis

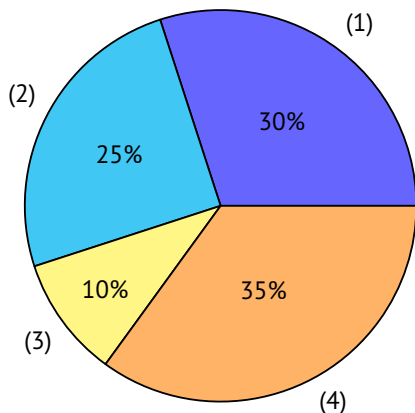
False Negative



1. Long paraphrase ($n > 5$)
2. Determiners
("mutation of **a** gene")
3. Inflected constituents
("holding of shares")

Error Analysis

False Negative



1. Long paraphrase ($n > 5$)
2. Determiners ("mutation of **a** gene")
3. Inflected constituents ("holding of shares")
4. Other errors

Recap

- Two tasks of recognizing semantic relations between nouns:

Recap

- Two tasks of recognizing semantic relations between nouns:
 - Between arbitrary nouns / constituents of a noun-compound

Recap

- Two tasks of recognizing semantic relations between nouns:
 - Between arbitrary nouns / constituents of a noun-compound
 - Classification to ontological relations / free text paraphrasing

Recap

- Two tasks of recognizing semantic relations between nouns:
 - Between arbitrary nouns / constituents of a noun-compound
 - Classification to ontological relations / free text paraphrasing
- In both tasks, integrating features from *joint* corpus occurrences improved performance

Recap

- Two tasks of recognizing semantic relations between nouns:
 - Between arbitrary nouns / constituents of a noun-compound
 - Classification to ontological relations / free text paraphrasing
- In both tasks, integrating features from *joint* corpus occurrences improved performance
- Word embeddings are a useful tool, but not the only tool!

Recap

- Two tasks of recognizing semantic relations between nouns:
 - Between arbitrary nouns / constituents of a noun-compound
 - Classification to ontological relations / free text paraphrasing
- In both tasks, integrating features from *joint* corpus occurrences improved performance
- Word embeddings are a useful tool, but not the only tool!

Thanks *Kudos* for *forthe* attending *participating*!*

* Replaced with the most similar words using word2vec

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.
- [Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *ACL*, pages 539–545.
- [Hendrickx et al., 2013] Hendrickx, I., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Szpakowicz, S., and Veale, T. (2013). Semeval-2013 task 4: Free paraphrases of noun compounds. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 138–143. Association for Computational Linguistics.
- [Levy et al., 2015] Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations. *NAACL*.
- [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.
- [Nakov, 2013] Nakov, P. (2013). On the interpretation of noun compounds: Syntax, semantics, and entailment. *Natural Language Engineering*, 19(03):291–330.
- [Nakov and Hearst, 2006] Nakov, P. and Hearst, M. (2006). Using verbs to characterize noun-noun relations. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 233–244. Springer.
- [Pantel et al., 2007] Pantel, P., Bhagat, R., Coppola, B., Chklovski, T., and Hovy, E. (2007). ISP: Learning inferential selectional preferences. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 564–571, Rochester, New York. Association for Computational Linguistics.
- [Roller et al., 2014] Roller, S., Erk, K., and Boleda, G. (2014). Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING*, pages 1025–1036.

References II

- [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 and Dagan, 2018] Shwartz, V. and Dagan, I. (2018). Paraphrase to explicate: Revealing implicit noun-compound relations. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.
- [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*.
- [Surtani et al., 2013] Surtani, N., Batra, A., Ghosh, U., and Paul, S. (2013). liit-h: A corpus-driven co-occurrence based probabilistic model for noun compound paraphrasing. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 153–157.
- [Van de Cruys et al., 2013] Van de Cruys, T., Afantenos, S., and Muller, P. (2013). Melodi: A supervised distributional approach for free paraphrasing of noun compounds. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 144–147, Atlanta, Georgia, USA. Association for Computational Linguistics.
- [Versley, 2013] Versley, Y. (2013). Sfs-tue: Compound paraphrasing with a language model and discriminative reranking. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 148–152.

References III

[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.