Acquiring Lexical Semantic Knowledge

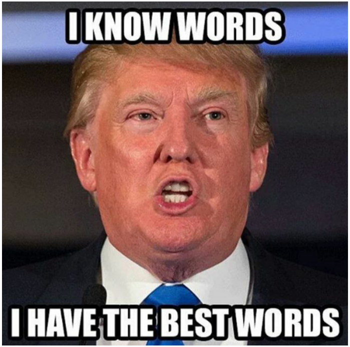
Vered Shwartz

Natural Language Processing Lab, Bar-Ilan University

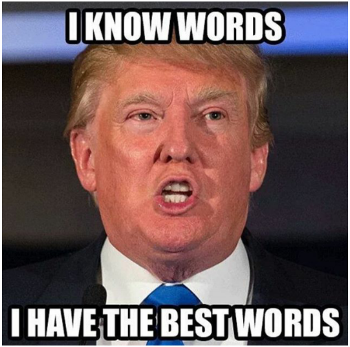
B I U
N L P

Bar-Ilan
University

What is "lexical knowledge"?

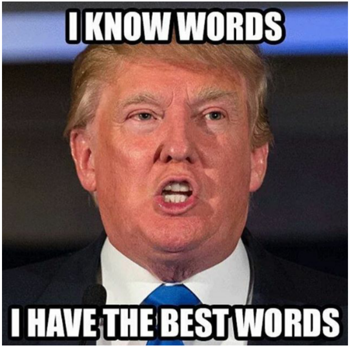


What is “lexical knowledge”?



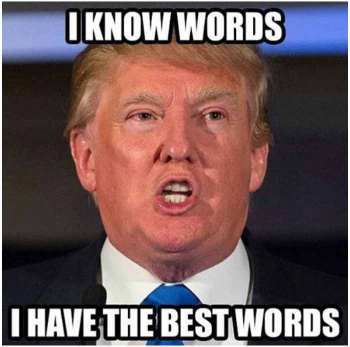
- Knowledge about lexical items (words, MWEs)

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 all problems)

- Provide semantic representations of words

Word Embeddings
(are not the solution for all problems)

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

Word Embeddings
(are not the solution for all problems)

- 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 all problems)

- 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 all problems)

- 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

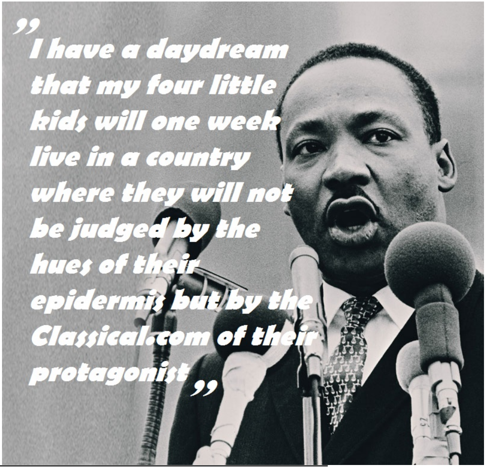
Word Embeddings
(are not the solution for all problems)

- 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

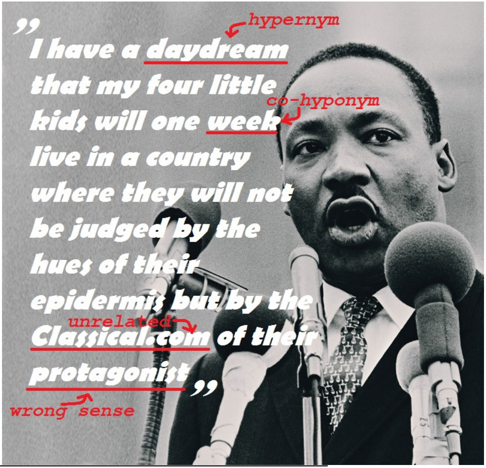
- To illustrate, take famous texts and replace nouns with their word2vec neighbours:[1]



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

Word Embeddings

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



[1] More 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

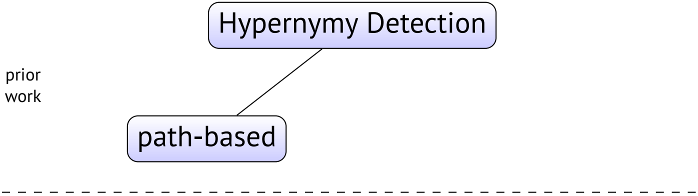
- Hypernymy
  - The hyponym is a subclass of / instance of the hypernym
  - *(cat, animal)*, *(Google, company)*

The Hypernymy Detection Task

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

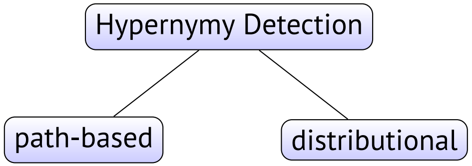
- Given two terms, *x* and *y*, decide whether *y* is a hypernym of *x*
  - in some senses of *x* and *y*, e.g. *(apple, fruit), (apple, company)*

Corpus-based Hypernymy Detection

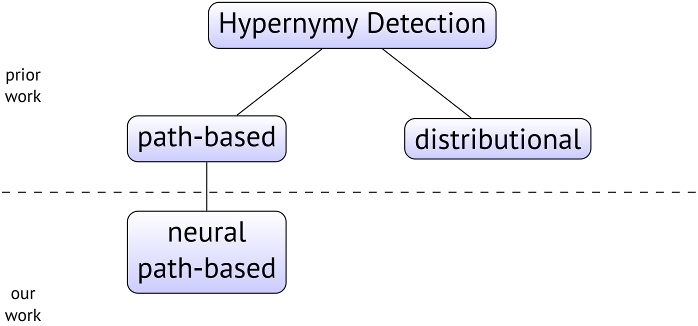


Corpus-based Hypernymy Detection

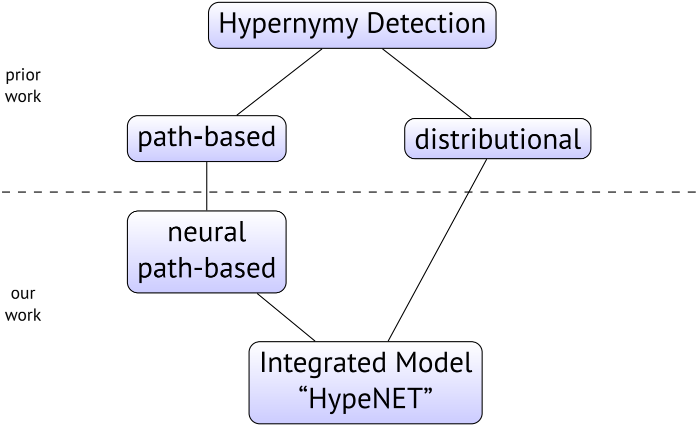
prior
work



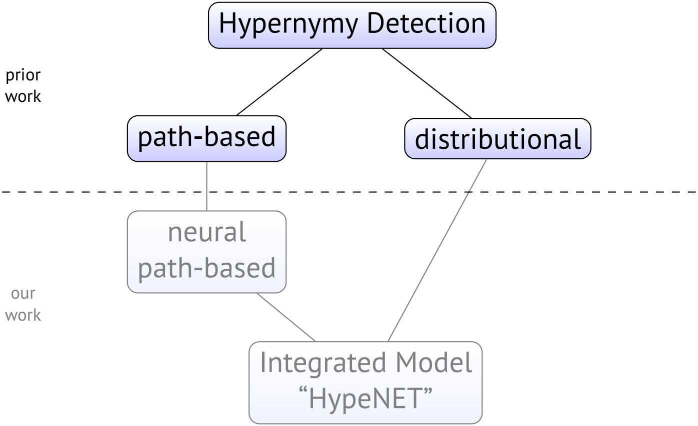
Corpus-based Hypernymy Detection



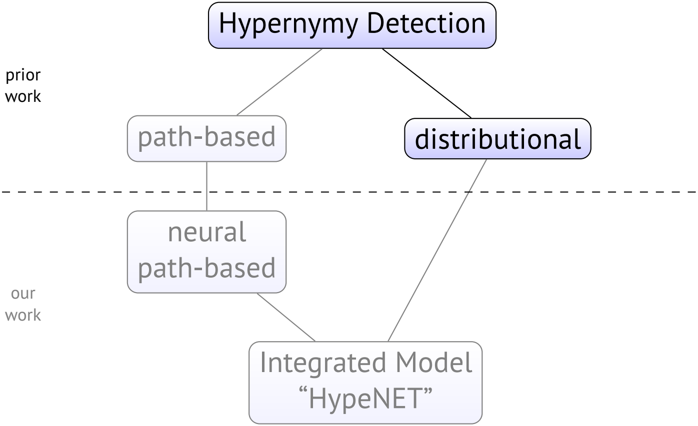
Corpus-based Hypernymy Detection



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

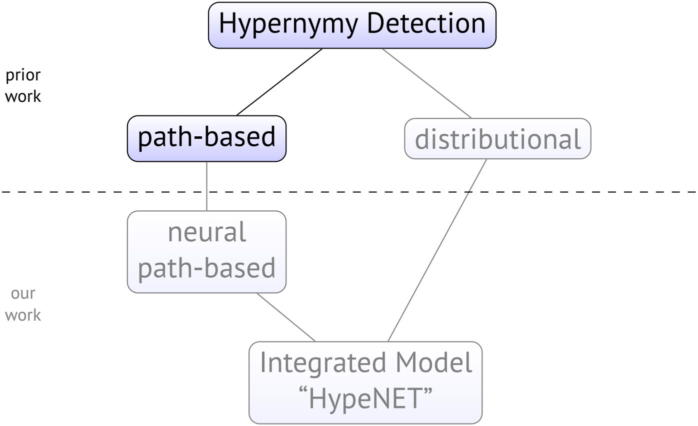
- Recognize the relation between words based on their *separate* occurrences in the corpus
- Train a classifier to predict hypernymy using the terms' embeddings:
  - Concatenation $\vec{x} \oplus \vec{y}$ [Baroni et al., 2012]
  - Difference $\vec{y} - \vec{x}$ [Roller et al., 2014, Weeds et al., 2014]
- 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., 2015a]: *"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

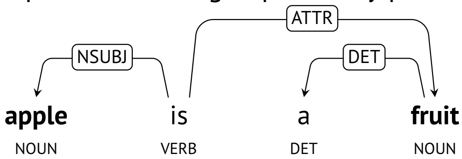
- Recognize the relation between *x* and *y* based on their *joint* occurrences in the corpus

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*

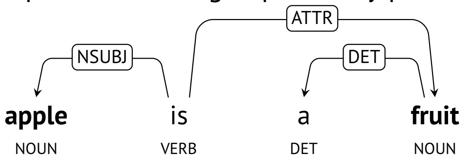
Path-based Approach

- 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

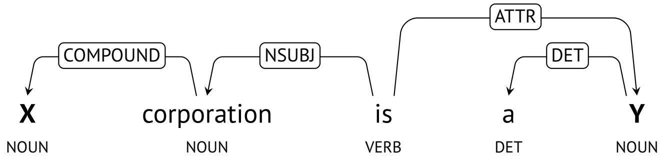
- The feature space is too sparse:

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

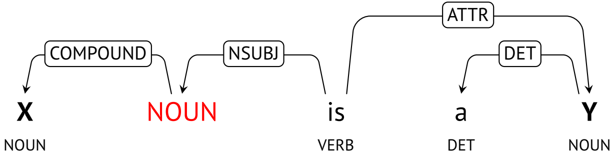
Path-based Approach Issues

- 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:



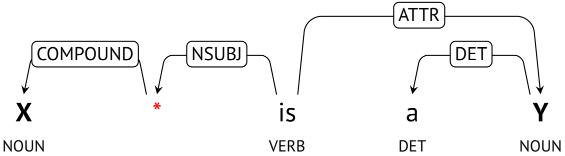
Path-based Approach Issues

- 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



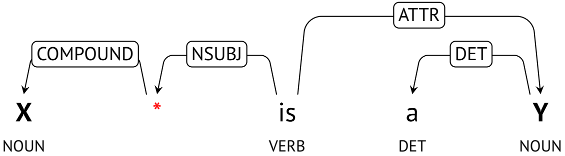
Path-based Approach Issues

- 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

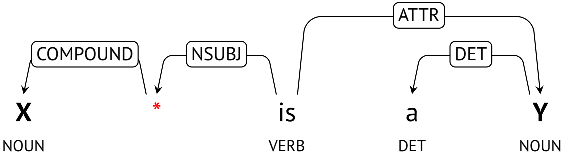
- The feature space is too sparse:
  - Similar paths share no information:
    X inc. is a Y
    X group is a Y
    X organization is a Y
- PATTY [Nakashole et al., 2012] generalized paths, by replacing a word by:
  - a wild-card



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

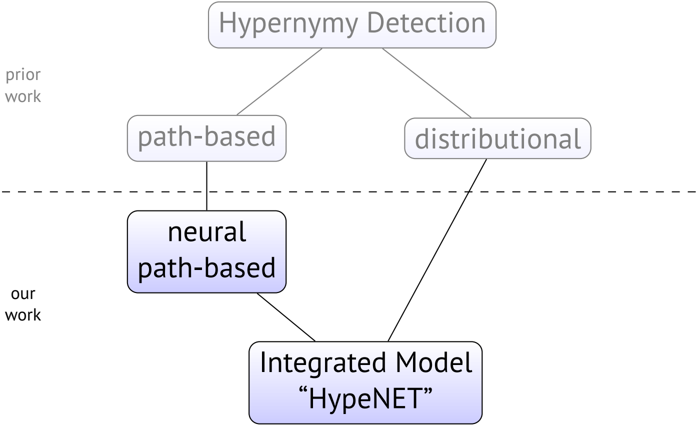
Path-based Approach Issues

- 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

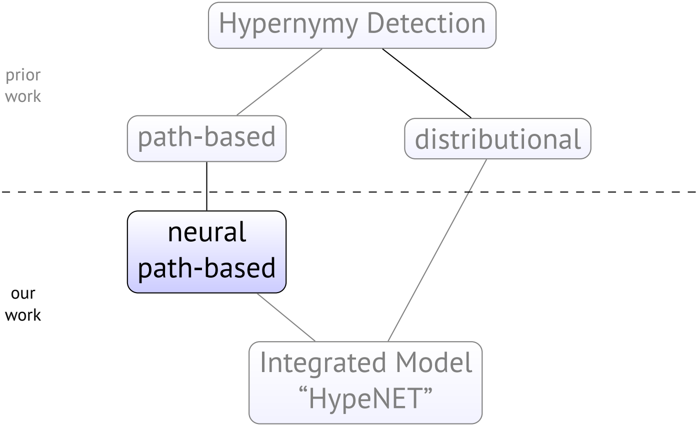


- Some of these generalizations are too general:
  - X is defined as Y $\approx$ X is described as Y via X is VERB as Y
  - X is defined as Y $\neq$ X is rejected as Y

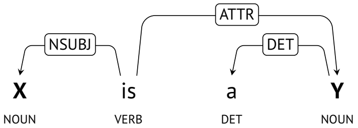
HypeNET: Integrated Path-based and Distributional Method [Shwartz et al., 2016]



First Step: Improving Path Representation



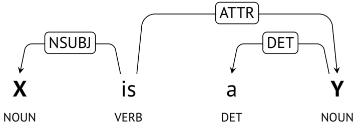
Path Representation



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

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

Path Representation

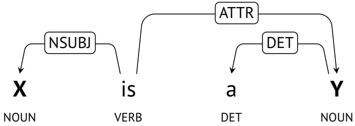


1. Split each path to edges, each edge consists of 4 components:

   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



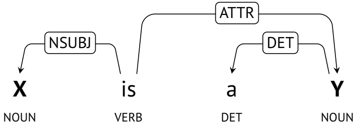
1. Split each path to edges, each edge consists of 4 components:

   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



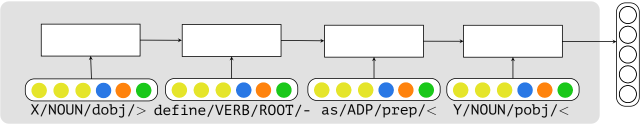
1. Split each path to edges, each edge consists of 4 components:

   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:

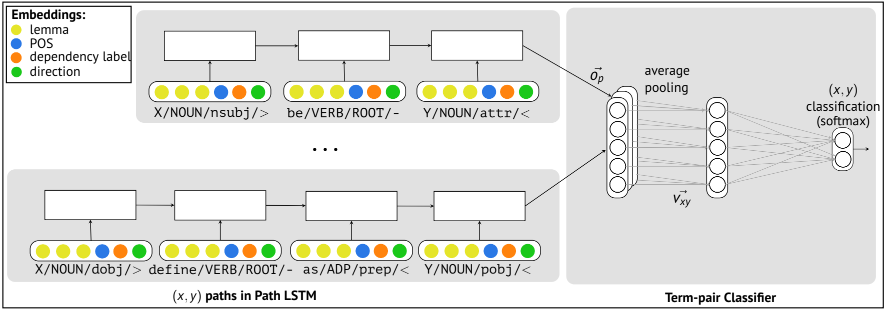
   [dependent lemma ; dependent POS ; dependency label ; direction]

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

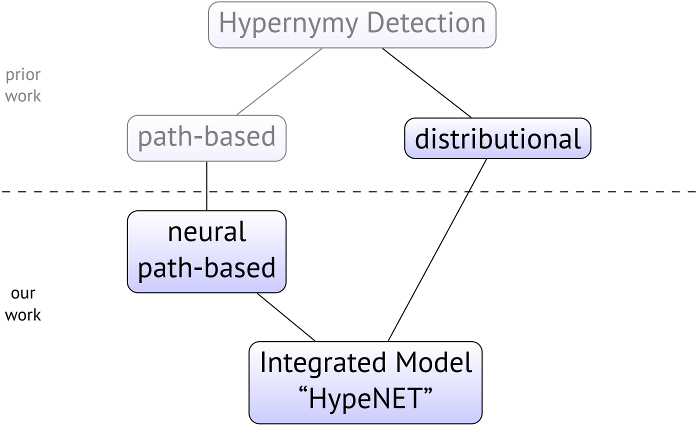


Term-pair Classification

- 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):



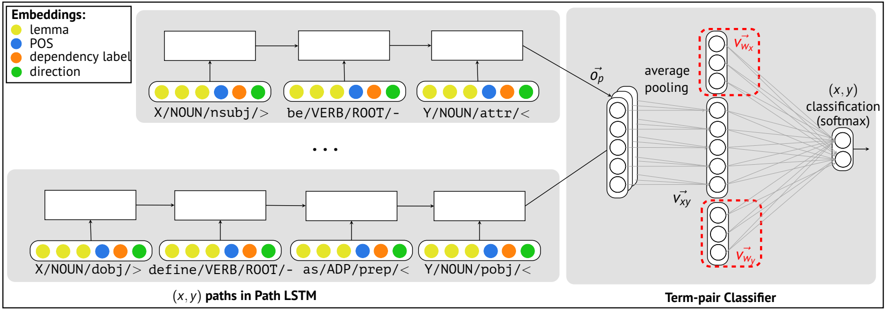
Second Step: Integrating Distributional Information



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**

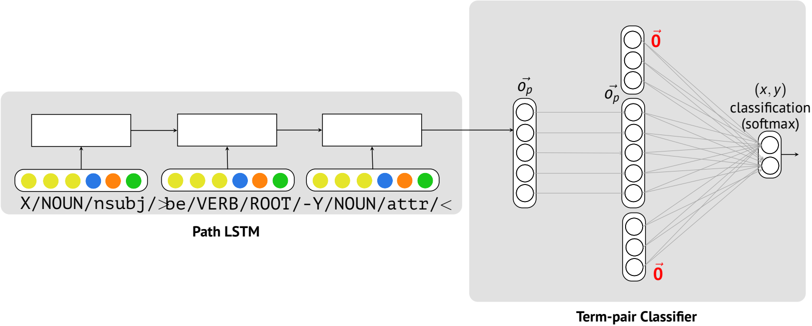
- The integrated method substantially outperforms both path-based and distributional methods

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

Analysis - Path Representation (2/2)

- Snow's method finds certain common paths:

  X company is a Y
  X ltd is a Y

Analysis - Path Representation (2/2)

- Snow's method finds certain common paths:

  X company is a Y

  X ltd is a Y

- PATTY-style generalizations find 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

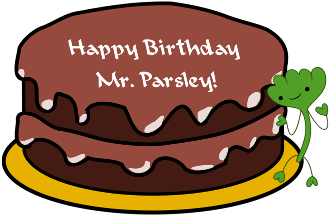
- Noun-compounds are prevalent in English, but most are rare
- We easily interpret noun-compounds we've never seen before
- What is a "*parsley cake*"?

1.

cake with/from parsley

(from `http://www.bazekalim.com`)

2.



cake for parsley

Interpreting new Noun Compounds

- What can cake be made of?

Corpus of Contemporary American English

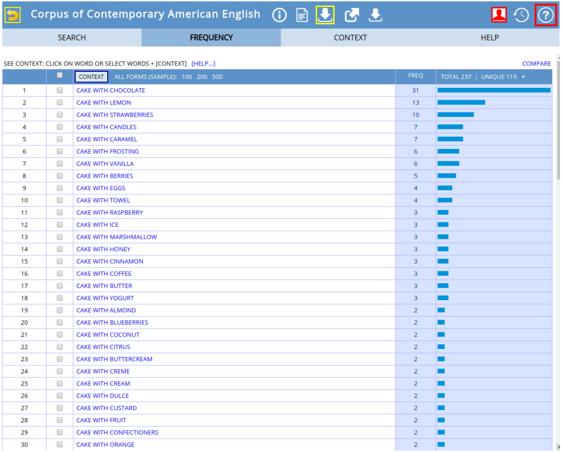
SEARCH | FREQUENCY | CONTEXT | HELP

SEE CONTEXT: CLICK ON WORD OR SELECT WORDS + [CONTEXT] [HELP...] COMPARE

	CONTEXT ALL FORMS (SAMPLE): 100 200 500	FREQ	TOTAL 237 | UNIQUE 119 +
1	CAKE WITH CHOCOLATE	31	
2	CAKE WITH LEMON	13	
3	CAKE WITH STRAWBERRIES	10	
4	CAKE WITH CANDLES	7	
5	CAKE WITH CARAMEL	7	
6	CAKE WITH FROSTING	6	
7	CAKE WITH VANILLA	6	
8	CAKE WITH BERRIES	5	
9	CAKE WITH EGGS	4	
10	CAKE WITH TOWEL	4	
11	CAKE WITH RASPBERRY	3	
12	CAKE WITH ICE	3	
13	CAKE WITH MARSHMALLOW	3	
14	CAKE WITH HONEY	3	
15	CAKE WITH CINNAMON	3	
16	CAKE WITH COFFEE	3	
17	CAKE WITH BUTTER	3	
18	CAKE WITH YOGURT	3	
19	CAKE WITH ALMOND	2	
20	CAKE WITH BLUEBERRIES	2	
21	CAKE WITH COCONUT	2	
22	CAKE WITH CITRUS	2	
23	CAKE WITH BUTTERCREAM	2	
24	CAKE WITH CREME	2	
25	CAKE WITH CREAM	2	
26	CAKE WITH DULCE	2	
27	CAKE WITH CUSTARD	2	
28	CAKE WITH FRUIT	2	
29	CAKE WITH CONFECTIONERS	2	
30	CAKE WITH ORANGE	2	

Interpreting new Noun Compounds

- What can cake be made of?



	CONTEXT ALL FORMS (SAMPLE): 100 200 500	FREQ
1	CAKE WITH CHOCOLATE	31
2	CAKE WITH LEMON	13
3	CAKE WITH STRAWBERRIES	10
4	CAKE WITH CANDLES	7
5	CAKE WITH CARAMEL	7
6	CAKE WITH FROSTING	6
7	CAKE WITH VANILLA	6
8	CAKE WITH BERRIES	5
9	CAKE WITH EGGS	4
10	CAKE WITH TOWEL	4
11	CAKE WITH RASPBERRY	3
12	CAKE WITH ICE	3
13	CAKE WITH MARSHMALLOW	3
14	CAKE WITH HONEY	3
15	CAKE WITH CINNAMON	3
16	CAKE WITH COFFEE	3
17	CAKE WITH BUTTER	3
18	CAKE WITH YOGURT	3
19	CAKE WITH ALMOND	2
20	CAKE WITH BLUEBERRIES	2
21	CAKE WITH COCONUT	2
22	CAKE WITH CITRUS	2
23	CAKE WITH BUTTERCREAM	2
24	CAKE WITH CREME	2
25	CAKE WITH CREAM	2
26	CAKE WITH DULCE	2
27	CAKE WITH CUSTARD	2
28	CAKE WITH FRUIT	2
29	CAKE WITH CONFECTIONERS	2
30	CAKE WITH ORANGE	2

TOTAL 237 | UNIQUE 119 +

- Parsley (sort of) fits into this distribution

Interpreting new Noun Compounds

- What can cake be made of?

Corpus of Contemporary American English

SEARCH | FREQUENCY | CONTEXT | HELP

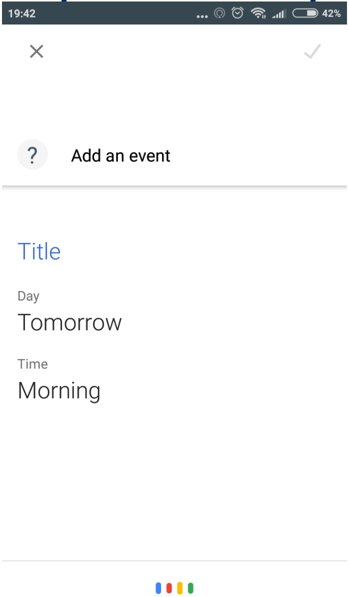
SEE CONTEXT: CLICK ON WORD OR SELECT WORDS + [CONTEXT] [HELP...] COMPARE

	CONTEXT ALL FORMS (SAMPLE): 100 200 500	FREQ	TOTAL 237 | UNIQUE 119 +
1	CAKE WITH CHOCOLATE	31	
2	CAKE WITH LEMON	13	
3	CAKE WITH STRAWBERRIES	10	
4	CAKE WITH CANDLES	7	
5	CAKE WITH CARAMEL	7	
6	CAKE WITH FROSTING	6	
7	CAKE WITH VANILLA	6	
8	CAKE WITH BERRIES	5	
9	CAKE WITH EGGS	4	
10	CAKE WITH TOWEL	4	
11	CAKE WITH RASPBERRY	3	
12	CAKE WITH ICE	3	
13	CAKE WITH MARSHMALLOW	3	
14	CAKE WITH HONEY	3	
15	CAKE WITH CINNAMON	3	
16	CAKE WITH COFFEE	3	
17	CAKE WITH BUTTER	3	
18	CAKE WITH YOGURT	3	
19	CAKE WITH ALMOND	2	
20	CAKE WITH BLUEBERRIES	2	
21	CAKE WITH COCONUT	2	
22	CAKE WITH CITRUS	2	
23	CAKE WITH BUTTERCREAM	2	
24	CAKE WITH CREME	2	
25	CAKE WITH CREAM	2	
26	CAKE WITH DULCE	2	
27	CAKE WITH CUSTARD	2	
28	CAKE WITH FRUIT	2	
29	CAKE WITH CONFECTIONERS	2	
30	CAKE WITH ORANGE	2	

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

We need Computers to Interpret Noun-Compounds

create a morning meeting

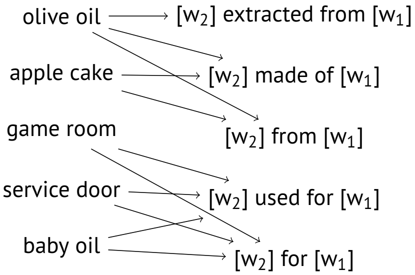


Noun-Compound Interpretation Tasks

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

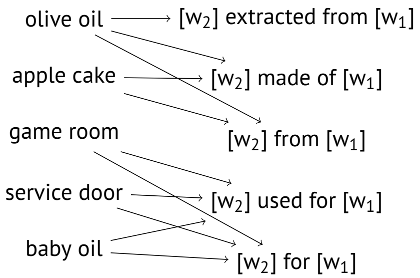
Noun-Compound Paraphrasing

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



Noun-Compound Paraphrasing

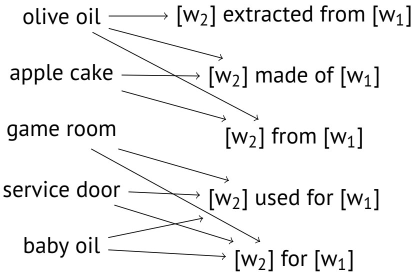
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



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

Noun-Compound Paraphrasing

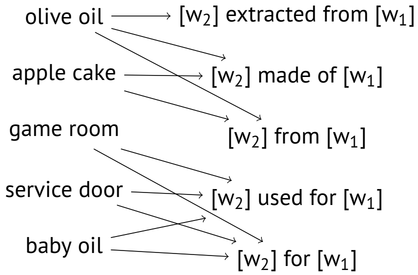
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



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

Noun-Compound Paraphrasing

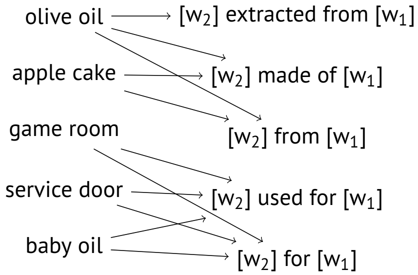
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



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

Noun-Compound Paraphrasing

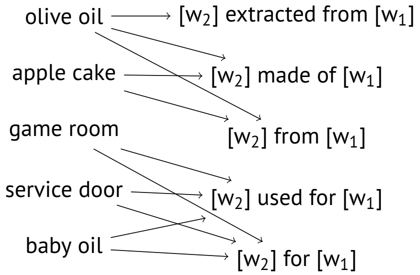
- To multiple prepositional and verbal paraphrases [Nakov and Hearst, 2006]



- 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. "[w_2] extracted from [w_1]" $\subset$ "[w_2] made of [w_1]"

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. "[w_2] extracted from [w_1]" $\subset$ "[w_2] made of [w_1]"

- 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 = apple, w_2 = cake, p = "[w$_2$] 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 = apple, w_2 = cake, p = "[w$_2$] 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*?

Multi-task Reformulation

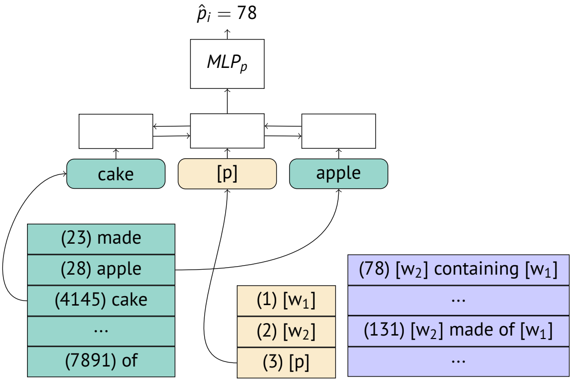
- Previous approaches: predict a paraphrase for a given NC
- Our model: multi-task learning problem
- Training example {w_1 = apple, w_2 = cake, p = "[w_2] made of [w_1]"}
    1. Predict a paraphrase p for a given NC $w_1 w_2$:
       What is the relation between *apple* and *cake*?
    2. Predict w_1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

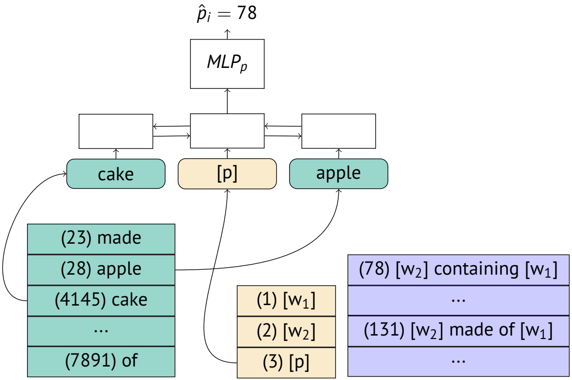
What is the relation between *apple* and *cake*?



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

Main Task (1): Predicting Paraphrases

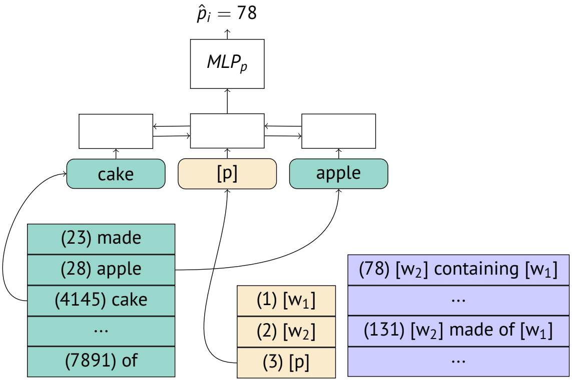
What is the relation between *apple* and *cake*?



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

Main Task (1): Predicting Paraphrases

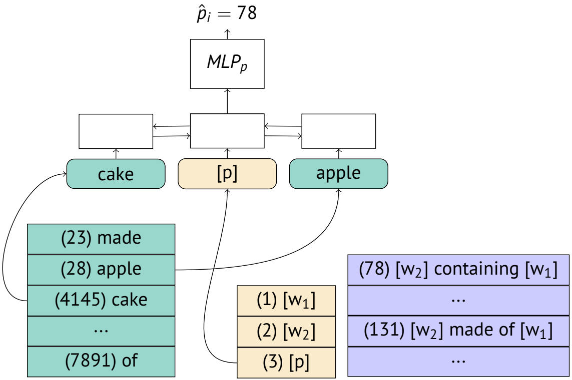
What is the relation between *apple* and *cake*?



- 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

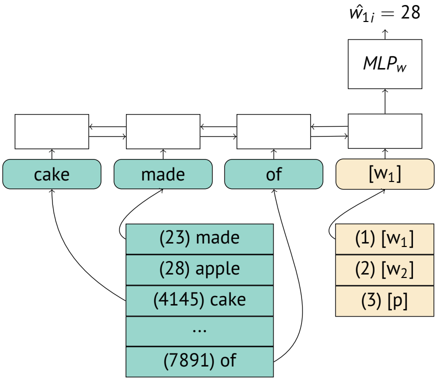
What is the relation between *apple* and *cake*?



- 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

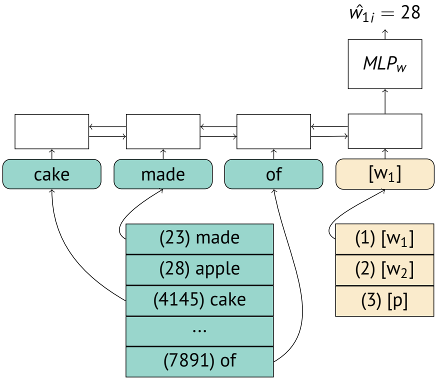
What can *cake* be made of?



- Encode placeholder in "cake made of $[w_1]$" using biLSTM

Helper Task (2): Predicting Missing Constituents

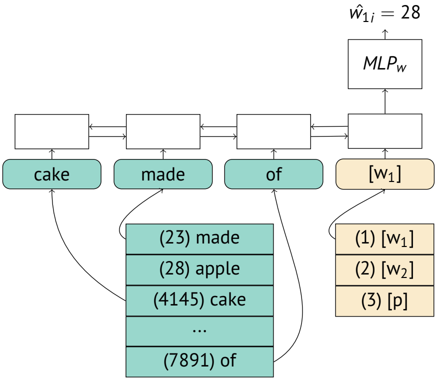
What can *cake* be made of?



- Encode placeholder in "cake made of $[w_1]$" 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 $[w_1]$" using biLSTM
- Predict an index in the word vocabulary
- (2) Generalizes paraphrases:
  "$[w_2]$ containing $[w_1]$" 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. “[w_2] verb prep [w_1]”)

Training Data

- Collected from Google N-grams
- Input:
  - Set of NCs
  - Templates of POS tags (e.g. "[w_2] verb prep [w_1]")
- Weighting by frequency and length

Training Data

- Collected from Google N-grams
- Input:
  - Set of NCs
  - Templates of POS tags (e.g. “[w_2] verb prep [w_1]”)
- Weighting by frequency and length
- 136,609 instances

Evaluation: Paraphrasing

Model

- Predict top *k* paraphrases for each noun compound

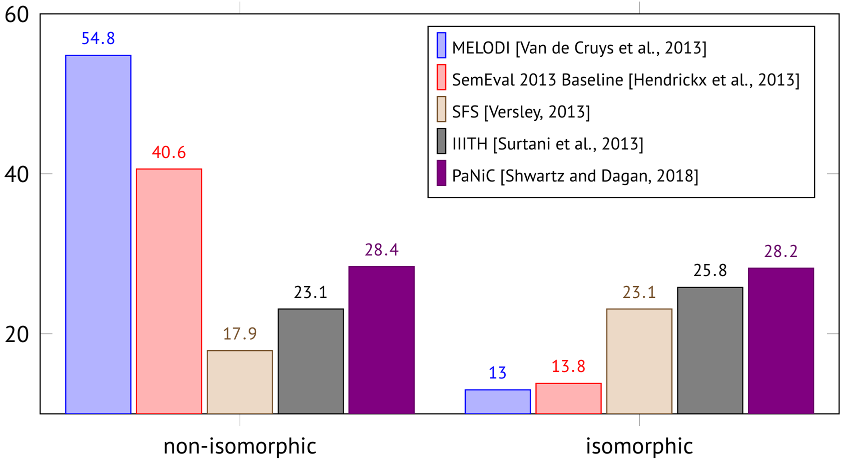
Model

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

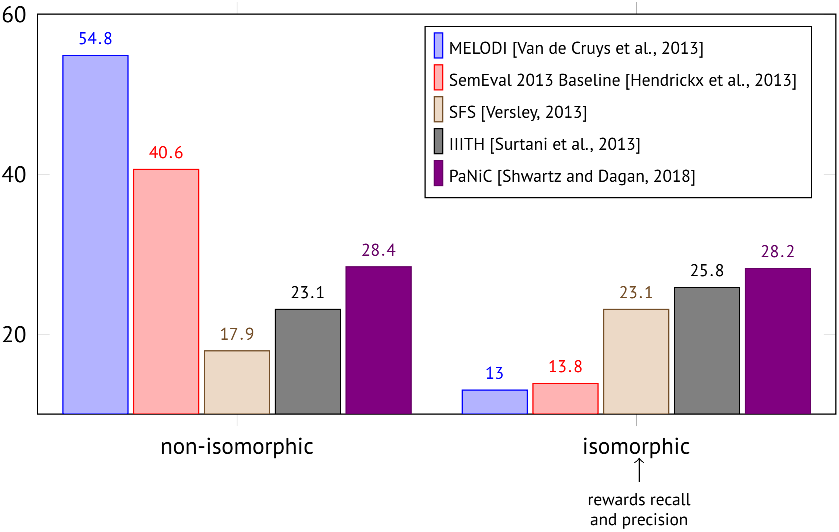
Model

- 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

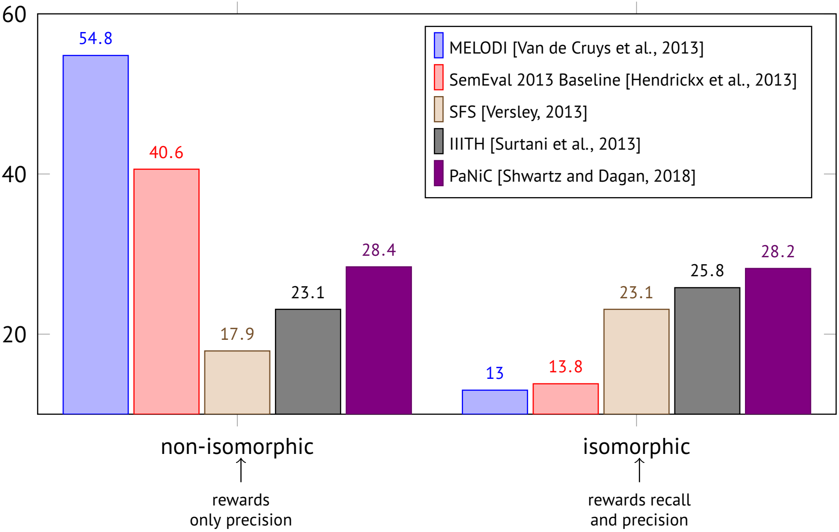
Results


60
40
20
54.8
40.6
17.9
23.1
28.4
13
13.8
23.1
25.8
28.2
MELODI [Van de Cruys et al., 2013]
SemEval 2013 Baseline [Hendrickx et al., 2013]
SFS [Versley, 2013]
IIITH [Surtani et al., 2013]
PaNiC [Shwartz and Dagan, 2018]
non-isomorphic
isomorphic

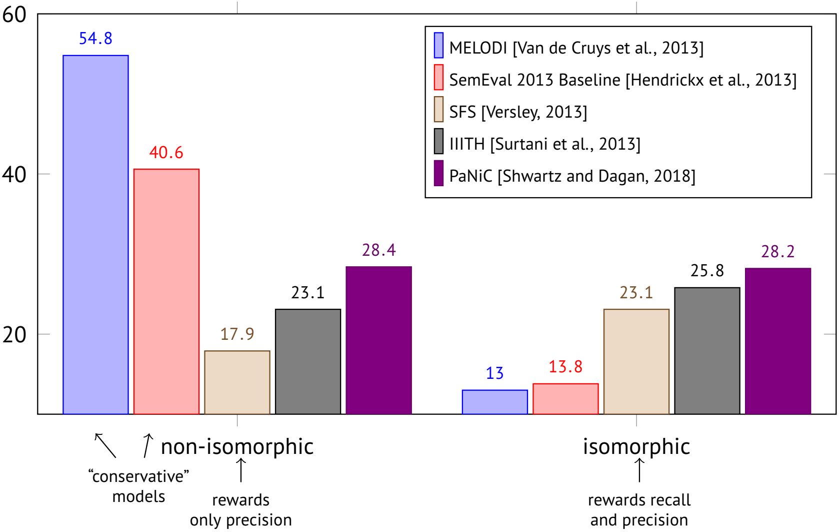
Results


60
40
20
54.8
40.6
17.9
23.1
28.4
13
13.8
23.1
25.8
28.2
MELODI [Van de Cruys et al., 2013]
SemEval 2013 Baseline [Hendrickx et al., 2013]
SFS [Versley, 2013]
IIITH [Surtani et al., 2013]
PaNiC [Shwartz and Dagan, 2018]
non-isomorphic
isomorphic
rewards recall
and precision

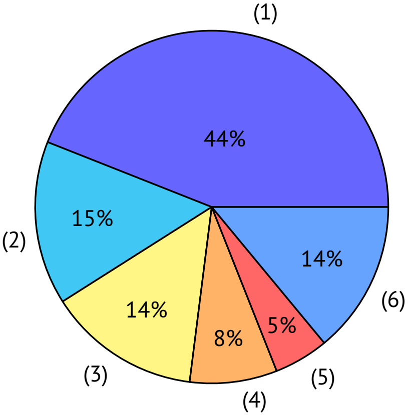
Results


60
40
20
54.8
40.6
17.9
23.1
28.4
13
13.8
23.1
25.8
28.2
MELODI [Van de Cruys et al., 2013]
SemEval 2013 Baseline [Hendrickx et al., 2013]
SFS [Versley, 2013]
IIITH [Surtani et al., 2013]
PaNiC [Shwartz and Dagan, 2018]
non-isomorphic
isomorphic
rewards only precision
rewards recall and precision

Results

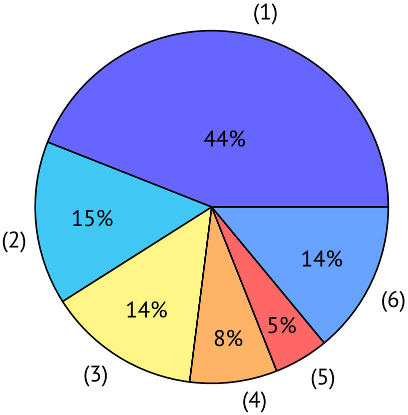

60
40
20
54.8
40.6
17.9
23.1
28.4
13
13.8
23.1
25.8
28.2
MELODI [Van de Cruys et al., 2013]
SemEval 2013 Baseline [Hendrickx et al., 2013]
SFS [Versley, 2013]
IIITH [Surtani et al., 2013]
PaNiC [Shwartz and Dagan, 2018]
non-isomorphic
isomorphic
"conservative" models
rewards only precision
rewards recall and precision

Error Analysis
False Positive



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

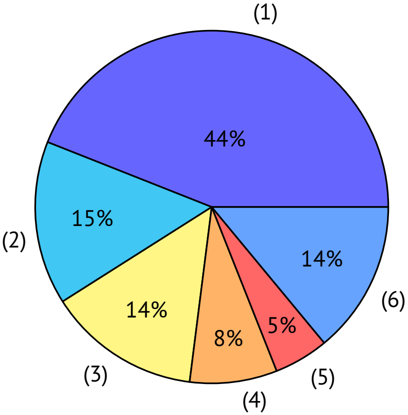
Error Analysis
False Positive



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

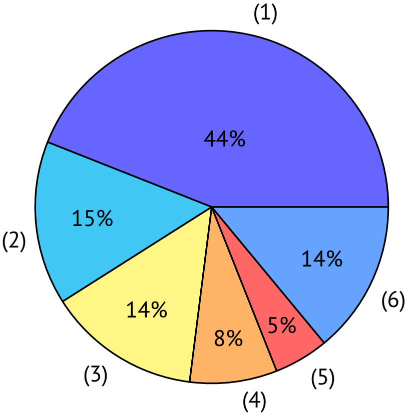
Error Analysis

False Positive



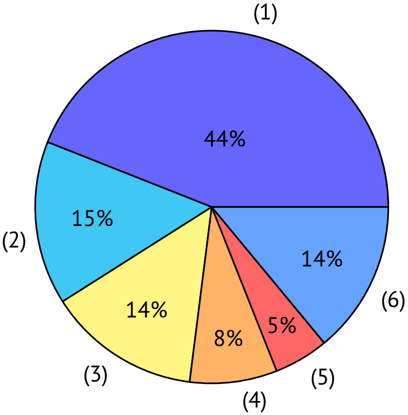
1. Valid, missing from gold-standard ("discussion by group")
2. Too specific ("life *of women in* community")
3. Incorrect prepositions
   E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby 's head" $\Rightarrow$ "oil from baby"

Error Analysis
False Positive



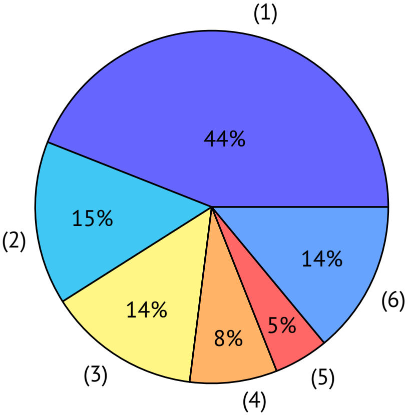
1. Valid, missing from gold-standard
   ("discussion by group")
2. Too specific
   ("life *of women in* community")
3. Incorrect prepositions
   E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby 's head" ⇒ "oil from baby"
4. Syntactic errors

Error Analysis
False Positive



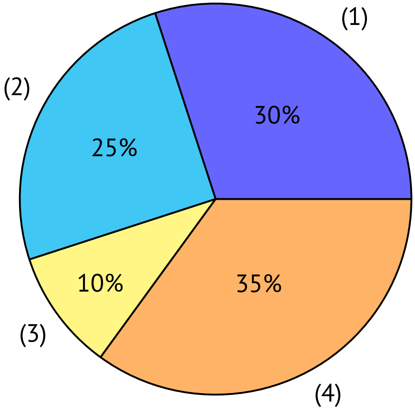
1. Valid, missing from gold-standard ("discussion by group")
2. Too specific ("life *of women in* community")
3. Incorrect prepositions
   E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby 's head" ⇒ "oil from baby"
4. Syntactic errors
5. Borderline grammatical ("force of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

Error Analysis
False Positive



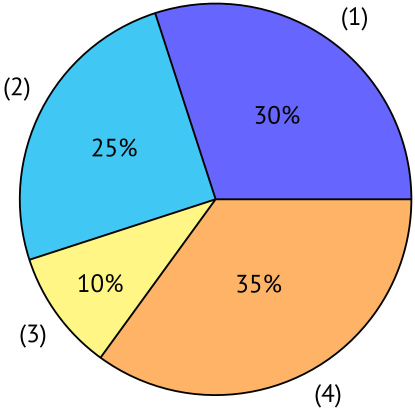
1. Valid, missing from gold-standard ("discussion by group")
2. Too specific ("life *of women in* community")
3. Incorrect prepositions
   E.g., n-grams don't respect syntactic structure: "rinse away the oil from baby 's head" ⇒ "oil from baby"
4. Syntactic errors
5. Borderline grammatical ("force of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

Error Analysis
False Negative



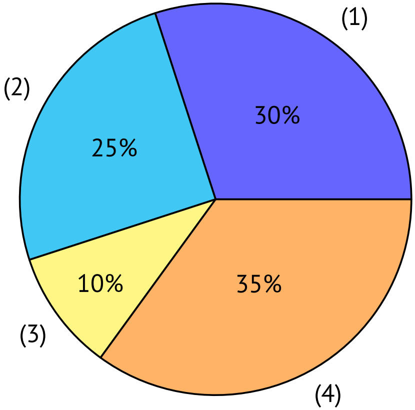
1. Long paraphrase ($n > 5$)

Error Analysis
False Negative



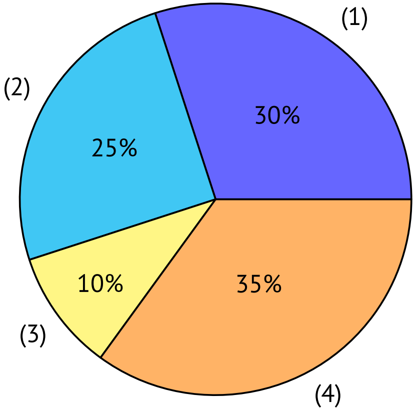
1. Long paraphrase ($n > 5$)
2. Determiners
   (“mutation of **a** gene”)

Error Analysis
False Negative



1. Long paraphrase ($n > 5$)
2. Determiners
   (“mutation of **a** gene”)
3. Inflected constituents
   (“holding of share**s**”)

Error Analysis
False Negative



1. Long paraphrase ($n > 5$)
2. Determiners
   (“mutation of **a** gene”)
3. Inflected constituents
   (“holding of share**s**”)
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.

[Dima, 2016] Dima, C. (2016). *Proceedings of the 1st Workshop on Representation Learning for NLP*, chapter On the Compositionality and Semantic Interpretation of English Noun Compounds, pages 27–39. Association for Computational Linguistics.

[Dinu et al., 2013] Dinu, G., Pham, N. T., and Baroni, M. (2013). General estimation and evaluation of compositional distributional semantic models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 50–58, Sofia, Bulgaria. Association for Computational Linguistics.

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

[Levy et al., 2015b] Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015b). Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado. Association for Computational Linguistics.

[Mitchell and Lapata, 2010] Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

[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.

References II

[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.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. 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.

[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.

References III

[Shwartz and Waterson, 2018] Shwartz, V. and Waterson, C. (2018). Olive oil is made of olives, baby oil is made for babies: Interpreting noun compounds using paraphrases in a neural model. In *The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, New Orleans, Louisiana.

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

[Socher et al., 2012] Socher, R., Huval, B., Manning, D. C., and Ng, Y. A. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

[Surtani et al., 2013] Surtani, N., Batra, A., Ghosh, U., and Paul, S. (2013). Iiit-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.

[Tratz, 2011] Tratz, S. (2011). *Semantically-enriched parsing for natural language understanding*. University of Southern California.

[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.

[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.

References IV

[Zanzotto et al., 2010] Zanzotto, F. M., Korkontzelos, I., Fallucchi, F., and Manandhar, S. (2010). Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1263–1271. Association for Computational Linguistics.

Noun-Compound Classification

Noun-Compound Classification

Given a noun-compound $w_1 w_2$, classify the relation between the head w_2 and the modifier w_1 to one of a set of pre-defined relations

source

ground attack

olive oil

apple cake

part of

sea bass

boat whistle

rotor head

purpose

baby oil

game room

service door

non-compositional

horse radish

baby sitting

hot dog

Current SOTA in NC Classification

- Compute a vector for $w_1 w_2$ as a function of w_1 and w_2's vectors
  - $vec(olive\ oil) = f(vec(olive), vec(oil))$

Current SOTA in NC Classification

- Compute a vector for $w_1 w_2$ as a function of w_1 and w_2's vectors
  - $vec(olive\ oil) = f(vec(olive), vec(oil))$
  - Many ways to learn f [Mitchell and Lapata, 2010, Zanzotto et al., 2010, Dinu et al., 2013, Socher et al., 2012]

Current SOTA in NC Classification

- Compute a vector for $w_1 w_2$ as a function of w_1 and w_2's vectors
  - $vec(olive\ oil) = f(vec(olive), vec(oil))$
  - Many ways to learn f [Mitchell and Lapata, 2010, Zanzotto et al., 2010, Dinu et al., 2013, Socher et al., 2012]
- Use the noun-compound representation as a feature vector for classification [Dima, 2016]

Current SOTA in NC Classification

- Compute a vector for $w_1 w_2$ as a function of w_1 and w_2's vectors
  - $vec(olive\ oil) = f(vec(olive), vec(oil))$
  - Many ways to learn f [Mitchell and Lapata, 2010, Zanzotto et al., 2010, Dinu et al., 2013, Socher et al., 2012]

- Use the noun-compound representation as a feature vector for classification [Dima, 2016]
  - Best performance is achieved when $f(w_1, w_2) = [w_1; w_2]$ 🤔

Current SOTA in NC Classification

- Compute a vector for $w_1 w_2$ as a function of w_1 and w_2's vectors
  - $vec(olive\ oil) = f(vec(olive), vec(oil))$
  - Many ways to learn f [Mitchell and Lapata, 2010, Zanzotto et al., 2010, Dinu et al., 2013, Socher et al., 2012]

- Use the noun-compound representation as a feature vector for classification [Dima, 2016]
  - Best performance is achieved when $f(w_1, w_2) = [w_1; w_2]$ 🤔
  - [Dima, 2016]: There is a **lexical memorization** issue

Olive Oil is Made of Olives, Baby Oil is Made for Babies:
Interpreting Noun Compounds using Paraphrases in a Neural Model [Shwartz and Waterson, 2018]

- The task is similar to **semantic relation classification**

Olive Oil is Made of Olives, Baby Oil is Made for Babies:
Interpreting Noun Compounds using Paraphrases in a Neural Model [Shwartz and Waterson, 2018]

- The task is similar to **semantic relation classification**
- Difference: we are interested in the relation between *olive* and *oil* in the context of the noun-compound, not in general

Olive Oil is Made of Olives, Baby Oil is Made for Babies:
Interpreting Noun Compounds using Paraphrases in a Neural Model [Shwartz and Waterson, 2018]

- The task is similar to **semantic relation classification**
- Difference: we are interested in the relation between *olive* and *oil* in the context of the noun-compound, not in general
- We apply lessons learned from semantic relation classification to noun-compound interpretation:

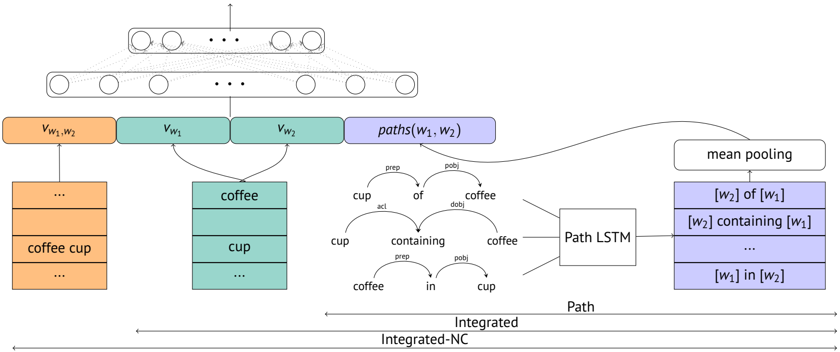
Olive Oil is Made of Olives, Baby Oil is Made for Babies:
Interpreting Noun Compounds using Paraphrases in a Neural Model [Shwartz and Waterson, 2018]

- The task is similar to **semantic relation classification**
- Difference: we are interested in the relation between *olive* and *oil* in the context of the noun-compound, not in general
- We apply lessons learned from semantic relation classification to noun-compound interpretation:
  - Use paraphrases as features, using LexNET

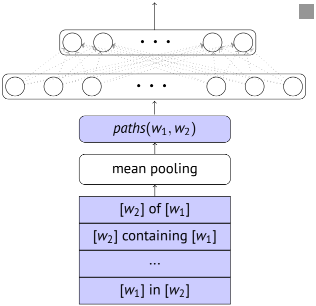
Olive Oil is Made of Olives, Baby Oil is Made for Babies:
Interpreting Noun Compounds using Paraphrases in a Neural Model [Shwartz and Waterson, 2018]

- The task is similar to **semantic relation classification**
- Difference: we are interested in the relation between *olive* and *oil* in the context of the noun-compound, not in general
- We apply lessons learned from semantic relation classification to noun-compound interpretation:
  - Use paraphrases as features, using LexNET
  - Use lexical splits to disable lexical memorization

Overall Architecture

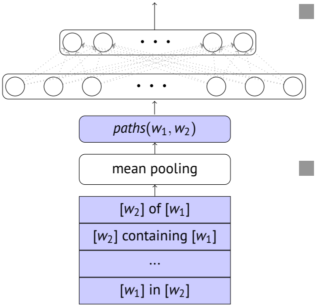


Path-based



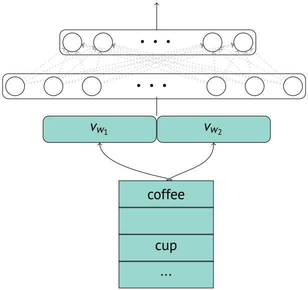
- Each dependency path connecting w_1 and w_2 in the corpus is encoded as in HypeNET [Shwartz et al., 2016]

Path-based



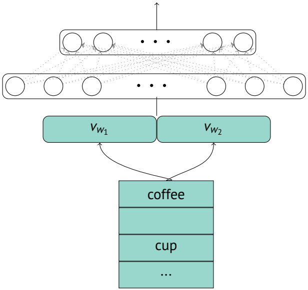
- Each dependency path connecting w_1 and w_2 in the corpus is encoded as in HypeNET [Shwartz et al., 2016]
- Motivation: semantic generalization of paths
  - $[w_2]$ *obtained from* $[w_1]$
  - $[w_2]$ *extracted from* $[w_1]$

Distributional



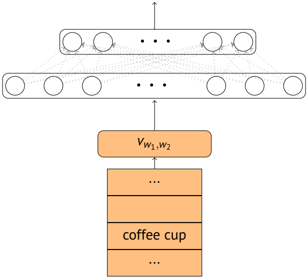
- Learn “prior probability” of relations for each individual word

Distributional



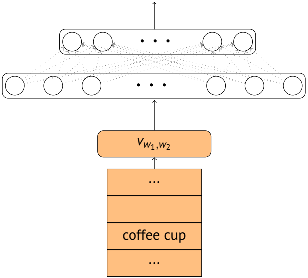
- Learn "prior probability" of relations for each individual word
- e.g. Substance-Material-Ingredient for edible w_1s:
  - *vanilla pudding*
  - *apple cake*

Distributional - Noun Compound



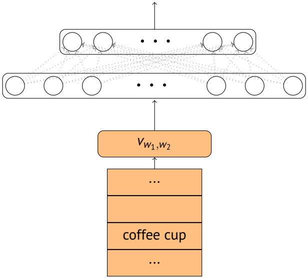
- Each NC has an *observed* vector

Distributional - Noun Compound



- Each NC has an *observed* vector
- Learned with GloVe [Pennington et al., 2014] by replacing NCs with a single token (e.g. *vanilla_pudding*).

Distributional - Noun Compound



- Each NC has an *observed* vector
- Learned with GloVe [Pennington et al., 2014] by replacing NCs with a single token (e.g. *vanilla_pudding*).

- Motivation: cluster NCs that appear in similar contexts
  - e.g. *gun violence* and *abortion rights* appear in news, both are Topic

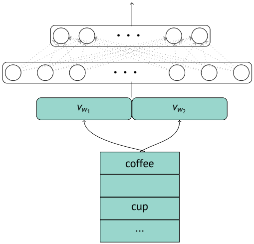
Evaluation - Datasets

- Dataset: [Tratz, 2011]
  - 19,158 instances
  - 37 relations (fine-grained) / 12 relations (coarse-grained)

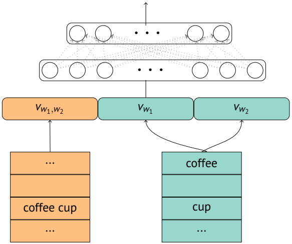
Evaluation - Datasets

- Dataset: [Tratz, 2011]
  - 19,158 instances
  - 37 relations (fine-grained) / 12 relations (coarse-grained)
- Datatset splits:
  - Random 75:20:5 (like previous work)
  - Lexical-full [Levy et al., 2015b]
  - Lexical-head
  - Lexical-mod

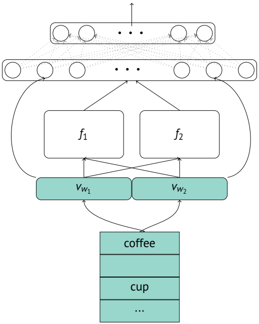
Evaluation - Baselines



Dist



Dist-NC



Compositional
[Dima, 2016]

Evaluation - Results

Dataset	Split	Best Baseline	Path	Int	Int-NC
Tratz-fine	Rand	**0.725**	0.538	0.714	0.692
	Lex_{head}	0.458	0.448	**0.510**	0.478
	Lex_{mod}	0.607	0.472	**0.613**	0.600
	Lex_{full}	0.363	0.423	0.421	**0.429**
Tratz-coarse	Rand	**0.775**	0.586	0.736	0.712
	Lex_{head}	0.538	0.518	**0.569**	0.548
	Lex_{mod}	0.645	0.548	**0.646**	0.632
	Lex_{full}	0.409	0.472	0.475	**0.478**

- Random split: distributional/compositional baselines outperform all other methods, by memorizing words.

Evaluation - Results

Dataset	Split	Best Baseline	Path	Int	Int-NC
Tratz-fine	Rand	**0.725**	0.538	0.714	0.692
	Lex_{head}	0.458	0.448	**0.510**	0.478
	Lex_{mod}	0.607	0.472	**0.613**	0.600
	Lex_{full}	0.363	0.423	0.421	**0.429**
Tratz-coarse	Rand	**0.775**	0.586	0.736	0.712
	Lex_{head}	0.538	0.518	**0.569**	0.548
	Lex_{mod}	0.645	0.548	**0.646**	0.632
	Lex_{full}	0.409	0.472	0.475	**0.478**

- Lexical split: our methods perform better.

Evaluation - Results

Dataset	Split	Best Baseline	Path	Int	Int-NC
Tratz-fine	Rand	**0.725**	0.538	0.714	0.692
	Lex_{head}	0.458	0.448	**0.510**	0.478
	Lex_{mod}	0.607	0.472	**0.613**	0.600
	Lex_{full}	0.363	0.423	0.421	**0.429**
Tratz-coarse	Rand	**0.775**	0.586	0.736	0.712
	Lex_{head}	0.538	0.518	**0.569**	0.548
	Lex_{mod}	0.645	0.548	**0.646**	0.632
	Lex_{full}	0.409	0.472	0.475	**0.478**

- The performance gap is larger in lexical-full.

Evaluation - Results

Dataset	Split	Best Baseline	Path	Int	Int-NC
Tratz-fine	Rand	**0.725**	0.538	0.714	0.692
	Lex_{head}	0.458	0.448	**0.510**	0.478
	Lex_{mod}	0.607	0.472	**0.613**	0.600
	Lex_{full}	0.363	0.423	0.421	**0.429**
Tratz-coarse	Rand	**0.775**	0.586	0.736	0.712
	Lex_{head}	0.538	0.518	**0.569**	0.548
	Lex_{mod}	0.645	0.548	**0.646**	0.632
	Lex_{full}	0.409	0.472	0.475	**0.478**

- There is usually no gain from adding the NC embeddings.

Analysis
Which relations can the path-based model learn?

relation	path	examples
measure	$[w_2]$ varies by $[w_1]$	*state limit*
	2,560 $[w_1]$ portion of $[w_2]$	*acre estate*
personal title	$[w_2]$ Anderson $[w_1]$/title	*Mrs. Brown*
	$[w_2]$ Sheridan $[w_1]$/title	*Gen. Johnson*
create-provide-generate-sell	$[w_2]$ produce $[w_1]$	*food producer*
	$[w_2]$ manufacture $[w_1]$	*engine plant*
time-of1	$[w_2]$ begin $[w_1]$	*morning program*
	$[w_2]$ held Saturday $[w_1]$	*afternoon meeting*
substance-material - ingredient	$[w_2]$ made of wood and $[w_1]$	*marble table*
	$[w_2]$ material includes type of $[w_1]$	*steel pipe*

Analysis

Which relations CAN'T the path-based model learn?

- lexicalized has no indicative paths! (e.g. *soap opera*)

Analysis
Which relations CAN'T the path-based model learn?

- lexicalized has no indicative paths! (e.g. *soap opera*)
- partial_attribute_transfer (e.g. *bullet train*) has few indicative paths (e.g. "*train as fast as a bullet*")

Analysis

Which relations CAN'T the path-based model learn?

- lexicalized has no indicative paths! (e.g. *soap opera*)
- partial_attribute_transfer (e.g. *bullet train*) has few indicative paths (e.g. "*train as fast as a bullet*")
- Confusion between relations with subtle difference (e.g. various topic relations)

Analysis
Why didn't the NC embeddings help?

Test NC		Most Similar NC	
NC	**Label**	**NC**	**Label**
majority party	equative	*minority party*	whole+part_or_member_of
enforcement director	objective	*enforcement chief*	perform&engage_in
fire investigator	objective	*fire marshal*	organize&supervise&authority
stabilization plan	objective	*stabilization program*	perform&engage_in
investor sentiment	experiencer-of-experience	*market sentiment*	topic_of_cognition&emotion
alliance member	whole+part_or_member_of	*alliance leader*	objective

- 81% of the noun-compounds in the test set have embeddings

Analysis

Why didn't the NC embeddings help?

Test NC		Most Similar NC	
NC	**Label**	**NC**	**Label**
majority party	equative	*minority party*	whole+part_or_member_of
enforcement director	objective	*enforcement chief*	perform&engage_in
fire investigator	objective	*fire marshal*	organize&supervise&authority
stabilization plan	objective	*stabilization program*	perform&engage_in
investor sentiment	experiencer-of-experience	*market sentiment*	topic_of_cognition&emotion
alliance member	whole+part_or_member_of	*alliance leader*	objective

- 81% of the noun-compounds in the test set have embeddings
- Most similar compounds demonstrate high quality embeddings

Analysis

Why didn't the NC embeddings help?

Test NC		Most Similar NC	
NC	**Label**	**NC**	**Label**
majority party	equative	*minority party*	whole+part_or_member_of
enforcement director	objective	*enforcement chief*	perform&engage_in
fire investigator	objective	*fire marshal*	organize&supervise&authority
stabilization plan	objective	*stabilization program*	perform&engage_in
investor sentiment	experiencer-of-experience	*market sentiment*	topic_of_cognition&emotion
alliance member	whole+part_or_member_of	*alliance leader*	objective

- 81% of the noun-compounds in the test set have embeddings
- Most similar compounds demonstrate high quality embeddings
- There are inconsistencies in the annotation