Higher-order Lexical Semantic Models for Non-factoid Answer Rereanking

Daniel Fried[1], Peter Jansen[1], Gustave Hahn-Powell[1], Mihai Surdeanu[1], and Peter Clark[2]

[1]University of Arizona
[2]Allen Institute for Artificial Intelligence

Task: Answer reranking

Open-domain community question answering (Yahoo! Answers)

Dining Out > United Kingdom > London Next >

Where's the best place in soho to go for breakfast?

i want to go to a nice place where i can meet a friend for breakfast in soho, london, and have a reasonable breakfast - not a stale croissant and bit of bread, nor fast food. any recommendations? personal recommendations preferred...

Update: this is going to be for about 0730, so definitely breakfast, not brunch or anything else.

Follow 5 answers

Answers Relevance

Best Answer: Flat White serves very good coffee, though I'm not sure what time they open.
http://www.london-eating.co.uk/7134.htm

Fernandez & Wells on Beak Street is also good:
http://www.fernandezandwells.com/beak.ph...

Patisserie / boulangerie chain, Paul, has a branch on Old Compton Street, though I think they open at 8am:
http://www.paul-uk.com/content/find-a-pa...

But why not treat yourself to breakfast at the Wolseley? It's frequently voted in the 'Best for Breakfast' lists in national papers, and they open at 7am:
http://www.thewolseley.com/Default.aspx

Enjoy! :)

The Elegant Epicure · 7 years ago

1 0 Comment

Patisserie Valerie is very good, and opens at 7.30:

http://www.patisserie-valerie.co.uk/loca...

adacam · 7 years ago

0 0 Comment

Bridging the lexical chasm

Limitations of lexical matching methods: short texts, different vocabularies in questions and answers

Q: `Where's the best place in soho to go for breakfast?`

A: `Fernandez & Wells has good pancakes.`

Bridging the lexical chasm

Limitations of lexical matching methods: short texts, different vocabularies in questions and answers

Q: `Where's the best place in soho to go for breakfast?`

A: `Fernandez & Wells has good pancakes.`

Bridge gap with direct associations between terms:

- monolingual alignment model
- semantic similarity from word embeddings

Chaining direct evidence

- Given lexical associations:

Q: `Where's the best place in soho to go for breakfast?`
A: `Fernandez & Wells has good pancakes.`

Q: `What goes well with pancakes?`
A: `hashbrowns and toast`

Chaining direct evidence

- Given lexical associations:

  Q: `Where's the best place in soho to go for breakfast?`
  A: `Fernandez & Wells has good pancakes.`

  Q: `What goes well with pancakes?`
  A: `hashbrowns and toast`

Chaining direct evidence

- Given lexical associations:

Q: `Where's the best place in soho to go for breakfast?`
A: `Fernandez & Wells has good pancakes.`

breakfast $\rightarrow$ *pancakes*

Q: `What goes well with pancakes?`
A: `hashbrowns and toast`

pancakes $\rightarrow$ *hashbrowns*

Chaining direct evidence

- Given lexical associations:

Q: `Where's the best place in soho to go for breakfast?`
A: `Fernandez & Wells has good pancakes.`

breakfast $\rightarrow$ *pancakes*

Q: `What goes well with pancakes?`
A: `hashbrowns and toast`

pancakes $\rightarrow$ *hashbrowns*

- Infer indirect, unseen associations:

breakfast $\rightarrow$ *pancakes* $\rightarrow$ *hashbrowns*

Chaining direct evidence

- Given lexical associations:

Q: `Where's the best place in soho to go for breakfast?`
A: `Fernandez & Wells has good pancakes.`

breakfast → *pancakes*

Q: `What goes well with pancakes?`
A: `hashbrowns and toast`

pancakes → *hashbrowns*

- Infer indirect, unseen associations:

breakfast → *pancakes* → *hashbrowns*

Q: `Where should we go for breakfast?`

Chaining direct evidence

- Given lexical associations:

Q: `Where's the best place in soho to go for breakfast?`
A: `Fernandez & Wells has good pancakes.`

breakfast → *pancakes*

Q: `What goes well with pancakes?`
A: `hashbrowns and toast`

pancakes → *hashbrowns*

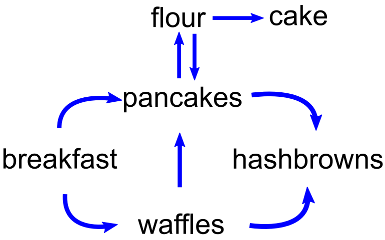
- Infer indirect, unseen associations:

breakfast → *pancakes* → *hashbrowns*

Q: `Where should we go for breakfast?`
A: `Reegee's has the best hashbrowns in town.`

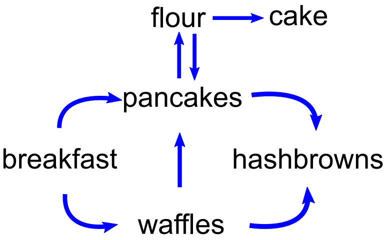
Evidence chaining as graph traversal

- Nodes are terms, edges are semantic associations
- Multiple steps give indirect associations



Evidence chaining as graph traversal

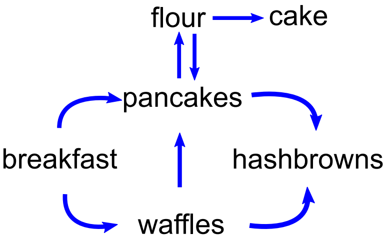
- Nodes are terms, edges are semantic associations
- Multiple steps give indirect associations



- How to build the association graph?

Evidence chaining as graph traversal

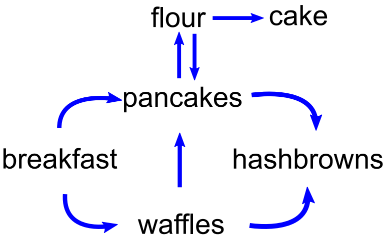
- Nodes are terms, edges are semantic associations
- Multiple steps give indirect associations



- How to build the association graph?
  first-order models: word embeddings, monolingual alignment

Evidence chaining as graph traversal

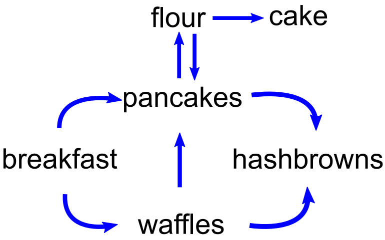
- Nodes are terms, edges are semantic associations
- Multiple steps give indirect associations



- How to build the association graph?
  first-order models: word embeddings, monolingual alignment
- How to efficiently traverse it?

Evidence chaining as graph traversal

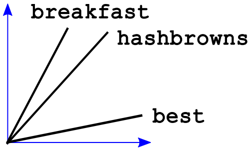
- Nodes are terms, edges are semantic associations
- Multiple steps give indirect associations



- How to build the association graph?
  first-order models: word embeddings, monolingual alignment
- How to efficiently traverse it?
  higher-order models: PageRank, conservative traversal

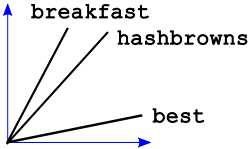
First-order embedding similarity

- Word embeddings from skip-gram model (Gigaword corpus)
- Use cosine similarity as a measure of lexical similarity



First-order embedding similarity

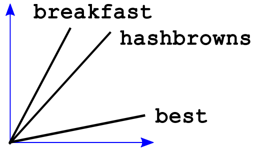
- Word embeddings from skip-gram model (Gigaword corpus)
- Use cosine similarity as a measure of lexical similarity



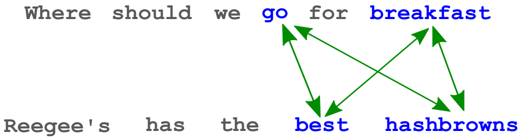
- Filter words in Q and A and compute similarity scores:

First-order embedding similarity

- Word embeddings from skip-gram model (Gigaword corpus)
- Use cosine similarity as a measure of lexical similarity



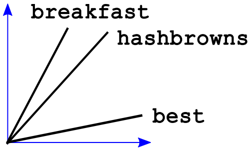
- Filter words in Q and A and compute similarity scores:



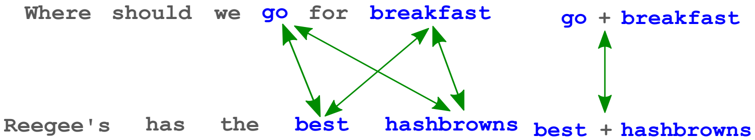
min, max, and average pairwise cosine sim

First-order embedding similarity

- Word embeddings from skip-gram model (Gigaword corpus)
- Use cosine similarity as a measure of lexical similarity



- Filter words in Q and A and compute similarity scores:



min, max, and average pairwise cosine sim

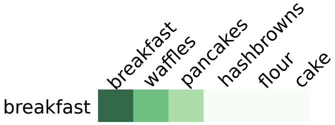
composed cosine sim

First-order alignment

- IBM Model 1: $P(Question|Answer)$
- Decomposes over alignments into $p(word_{question}|word_{answer})$

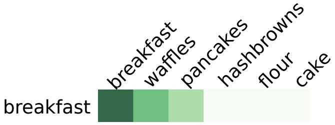
First-order alignment

- IBM Model 1: $P(Question|Answer)$
- Decomposes over alignments into $p(word_{question}|word_{answer})$
- Use vector $p(\cdot|word)$ as a distributed representation for *word*

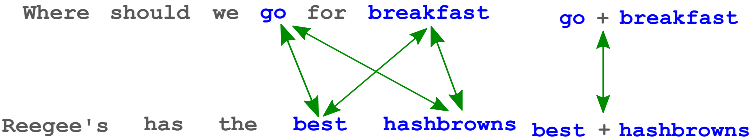


First-order alignment

- IBM Model 1: $P(Question|Answer)$
- Decomposes over alignments into $p(word_{question}|word_{answer})$
- Use vector $p(\cdot|word)$ as a distributed representation for *word*



- compare using Jensen Shannon distance (JSD)

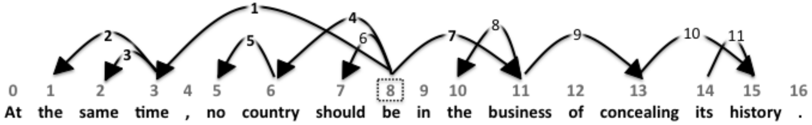


min, max, and average pairwise JSD

composed JSD

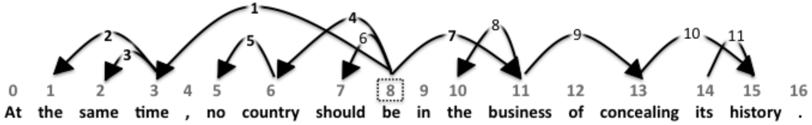
Modeling syntactic structures

- Applying alignment and embedding models beyond words
- Extract collapsed unlabelled dependencies:



Modeling syntactic structures

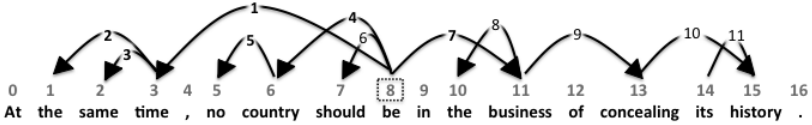
- Applying alignment and embedding models beyond words
- Extract collapsed unlabelled dependencies:




```
be_time, time_the, time_same, be_country, country_no, be_should,
be_business, business_the, business_concealing,
concealing_history, history_its
```

Modeling syntactic structures

- Applying alignment and embedding models beyond words
- Extract collapsed unlabelled dependencies:

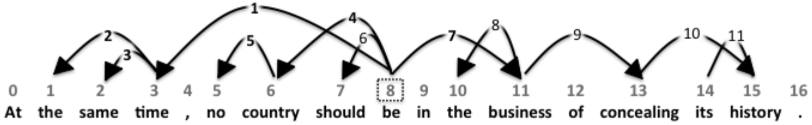


`be_time, time_the, time_same, be_country, country_no, be_should, be_business, business_the, business_concealing, concealing_history, history_its`

- Alignment model: unordered (bag-of-dependencies)
- Embedding model: skip-gram on depth-first traveral of dependency graph

Modeling syntactic structures

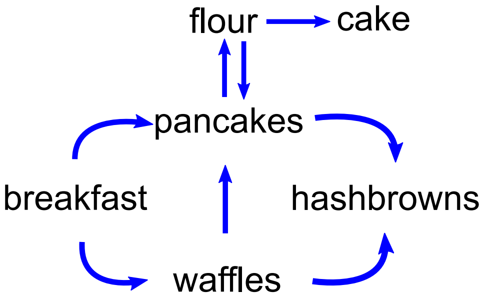
- Applying alignment and embedding models beyond words
- Extract collapsed unlabelled dependencies:



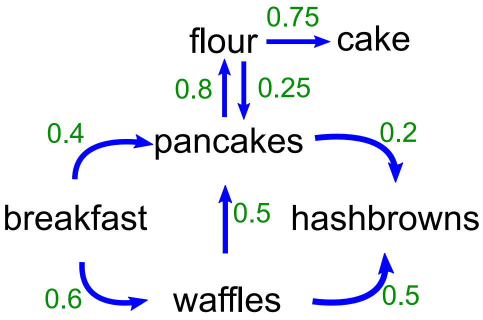
`be_time, time_the, time_same, be_country, country_no, be_should, be_business, business_the, business_concealing, concealing_history, history_its`

- Alignment model: unordered (bag-of-dependencies)
- Embedding model: skip-gram on depth-first traveral of dependency graph
- Both produce vector representations for the dependency pairs

Higher-order models: Chaining direct evidence

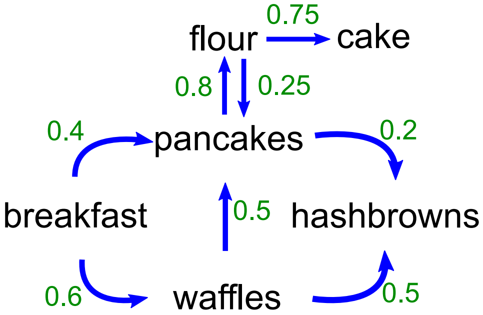


Higher-order models: Chaining direct evidence



- Edge weights are association strengths (from QA alignment probabilities, or normalized embedding similarities)

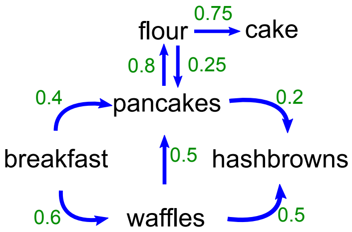
Higher-order models: Chaining direct evidence



- Edge weights are association strengths (from QA alignment probabilities, or normalized embedding similarities)

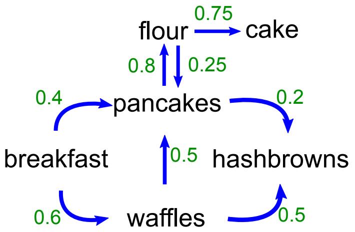
$$P(hashbrowns|breakfast; \text{1 step}) = 0$$
$$P(hashbrowns|breakfast; \text{2 steps}) = (0.4 * 0.2) + (0.6 * 0.5)$$
$$P(hashbrowns|breakfast; \text{3 steps}) = 0.4 * 0.5 * 0.5$$

Random walks on graphs

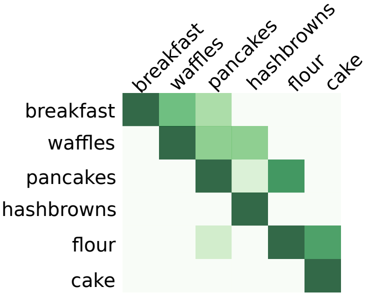


Graph

Random walks on graphs

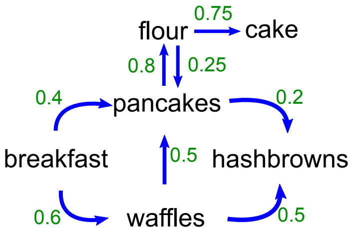


Graph

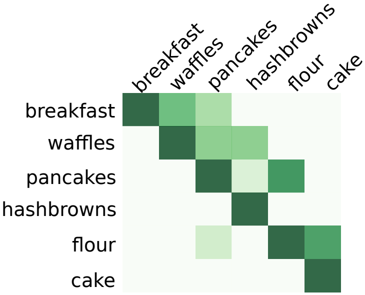


Adjacency Matrix, **A**

Random walks on graphs



Graph

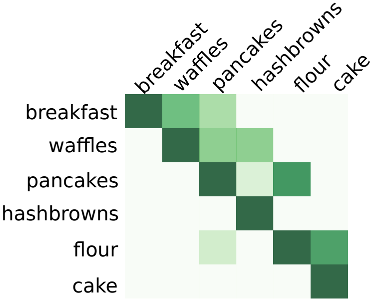


Adjacency Matrix, $\mathbf{A}$

- $\mathbf{A}^n$: probabilities of paths of length n (like PageRank)
- but long tail of association probabilities $\implies$ semantic drift

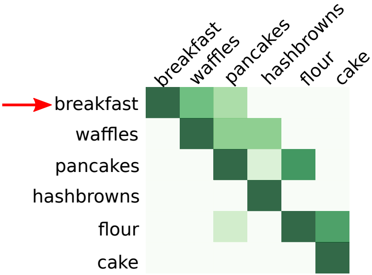
Cautious graph traversal

- Average each node's transition distribution with its k nearest neighbors (weighted by transition probabilities):
- $k = 2$:



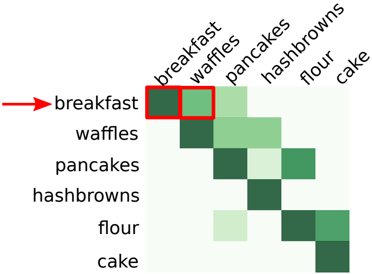
Cautious graph traversal

- Average each node's transition distribution with its k nearest neighbors (weighted by transition probabilities):
- $k = 2$:



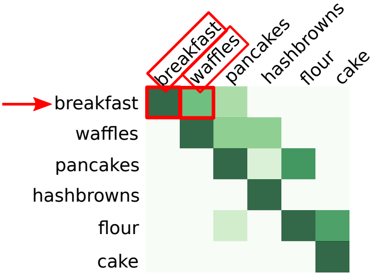
Cautious graph traversal

- Average each node's transition distribution with its k nearest neighbors (weighted by transition probabilities):
- $k = 2$:



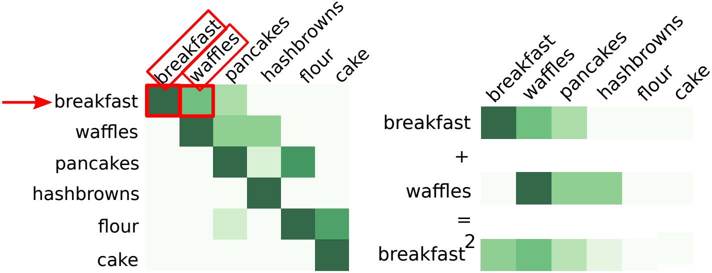
Cautious graph traversal

- Average each node's transition distribution with its k nearest neighbors (weighted by transition probabilities):
- $k = 2$:



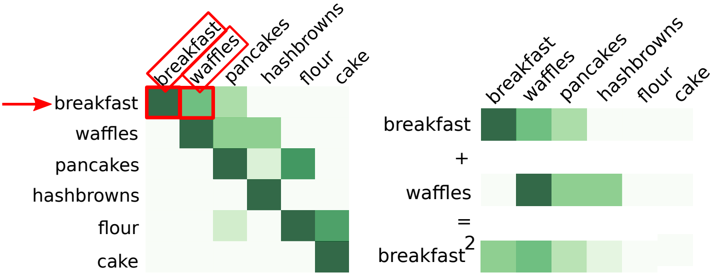
Cautious graph traversal

- Average each node's transition distribution with its k nearest neighbors (weighted by transition probabilities):
- $k = 2$:



Cautious graph traversal

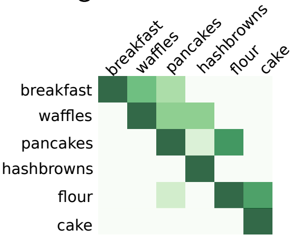
- Average each node's transition distribution with its k nearest neighbors (weighted by transition probabilities):
- $k = 2$:



- Produces a new set of *second-order* vectors. Can be iterated.
- Like a PageRank iteration, but only nearest neighbors.

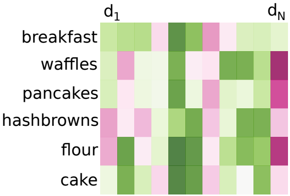
Inputs to the higher-order method

Term alignment distributions



- Nearest-neighbors encoded in each vector as conditional probabilities

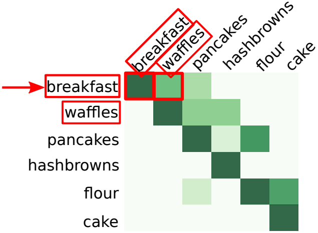
Neural network embeddings



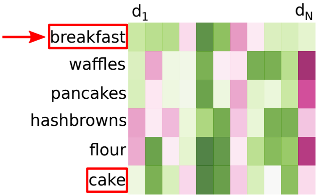
- Nearest-neighbors given by cosine similarity between vectors

Inputs to the higher-order method

Term alignment distributions

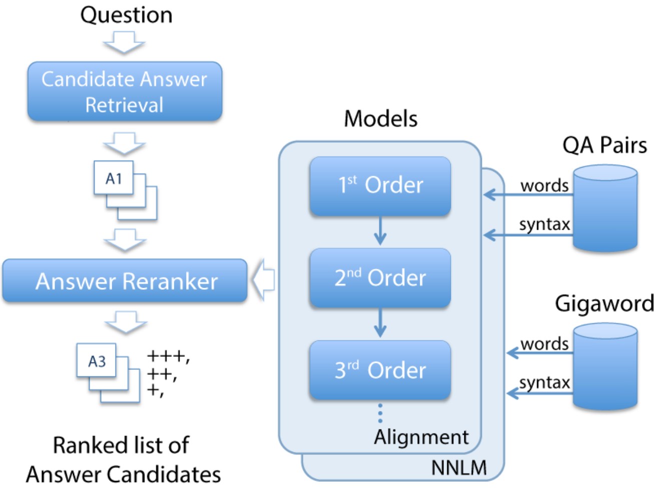


Neural network embeddings



- Nearest-neighbors encoded in each vector as conditional probabilities
- Nearest-neighbors given by cosine similarity between vectors

Reranking model architecture



Experiments

QA dataset:

- Yahoo! Answers Community Question Answering Corpus
- 10,000 "How" QA pairs (5k train, 2.5k dev, 2.5k test)
- Minimum 4 answers per question (average 9)

Experiments

QA dataset:

- Yahoo! Answers Community Question Answering Corpus
- 10,000 "How" QA pairs (5k train, 2.5k dev, 2.5k test)
- Minimum 4 answers per question (average 9)

Lexical association data:

- Alignment models: separate set of 100k Yahoo! QA pairs
  IBM Model 1, GIZA++
- Embedding models: Annotated Gigaword
  skip-gram with hierarchical sampling

Experiments

QA dataset:

- Yahoo! Answers Community Question Answering Corpus
- 10,000 “How” QA pairs (5k train, 2.5k dev, 2.5k test)
- Minimum 4 answers per question (average 9)

Lexical association data:

- Alignment models: separate set of 100k Yahoo! QA pairs
  IBM Model 1, GIZA++
- Embedding models: Annotated Gigaword
  skip-gram with hierarchical sampling

Higher-order Models:

- Use $k = 20$ nearest neighbors (tuned on dev set, stable values)

Results: higher-order models

- Higher-order helps for sparse training data: dependency embeddings and both types of alignment
- Does *not* help for word embeddings

IR baseline: 19.6% Precision at 1 (P@1)

Word Alignment

Models	P@1
Order 1	27.3
Order 1-2	29.0*
Order 1-3	**30.5***
Order 1-4	29.6*

Word Embeddings

Models	P@1
Order 1	**30.7**
Order 1-2	29.6
Order 1-3	30.2
Order 1-4	30.4

Dependency Alignment

Models	P@1
Order 1	25.89
Order 1-2	28.81*
Order 1-3	**29.41***

Dependency Embeddings

Models	P@1
Order 1	30.85
Order 1-2	31.69*
Order 1-3	**31.89***

*: significant ($p < 0.05$) increase over Order 1

Results: combining representations

- Aligment models complement embedding models
- Syntactic dependencies complement words

IR baseline: 19.6% Precision at 1 (P@1)

Word Align. + Emb.

Models	P@1
Order 1	30.85
Order 1-2	31.85*
Order 1-3	**32.09***
Order 1-4	31.69

Dependency Align. + Emb.

Models	P@1
Order 1	31.49
Order 1-2	**32.85***
Order 1-3	32.77*

Word + Dependency: Align. + Emb.

Models	P@1
Order 1	31.85
Order 1-2	$32.89^{\dagger}$
Order 1-3	$\mathbf{33.01}^{\dagger}$

*: significant ($p < 0.05$) increase over Order 1

†: nearly significant ($0.05 < p < 0.10$) increase over Order 1

Comparison to PageRank

- Add small teleportation probabilities to word alignment matrix
- Do power iteration (multiply matrix by itself)

Comparison to PageRank

- Add small teleportation probabilities to word alignment matrix
- Do power iteration (multiply matrix by itself)

Word Alignment

Models	P@1	Memory	Time
Order 1	27.3	75MB	–
Order 1-2	29.0*	1.8GB	33 sec
Order 1-3	**30.5***	9.7GB	4.5 min
Order 1-4	29.6*	19GB	8.6 min

PageRank

Models	P@1	Memory	Time
Order 1	27.1	41GB	–
Order 1-2	**31.01***	41GB	45.6 hrs
Order 1-3	29.89*	41GB	45.6 hrs

Ablation experiments

IR baseline: 19.6% Precision at 1 (P@1)

1st Order Word Alignment

Features	P@1	Δ P@1
all features	27.33	–
– $P(Question \mid Answer)$	25.69	-6%
– max JSD	27.33	0%
– min JSD	23.57	-14%
– average JSD	25.41	-7%
– composite JSD	27.17	-1%

1st Order Word Embeddings

Features	P@1	Δ P@1
all features	30.69	–
– max cosine sim.	29.65	-3%
– min cosine sim.	29.69	-3%
– average cosine sim.	26.49	-14%
– composite cosine sim.	27.01	-12%

Conclusions

- Conservative graph-based lexical inference
- Simple implementation, comparable performance to PageRank but large memory and time savings
- Toward robust, approximate inference for QA

Conclusions

- Conservative graph-based lexical inference
- Simple implementation, comparable performance to PageRank but large memory and time savings
- Toward robust, approximate inference for QA

Thanks!