Paper in review -

Dependency-Based Word Embeddings

Omer Levy and Yoav Goldberg
Bar-Ilan University

By –
Rishav Hada

rishavhada@gmail.com

Word Representation

- Words can not be represented as discrete and distinct symbols.
  - It is insufficient for many tasks and suffers from poor generalization.
  - Ex – Pizza and Hamburger.
- Therefore, we seek a representation that captures semantic and syntactic similarities between words.
- Common paradigm for acquiring such representation – **Distributional Hypothesis** – Words in similar context have similar meaning
  - Word clustering based on context.
  - High dimensional sparse vectors – each entry is a measure of the association between the word and a particular context.
  - **Word/Neural embeddings** - Most recent, represent words as dense vectors derived from various training methods inspired from neural network language modelling.
- State-of-the-art word embedding is the **Skip-gram with negative sampling**.

The Skip-Gram Model

- Skip-gram model can capture two semantics for a single word. Two vector representations for Orange: Color and Fruit.
- Other methods like CBOW takes the average of the context of a word. Will place Orange in between a cluster for color and fruit.
- In Skip-gram model,
  - Each word w ∈ W, is associated with a vector v_w
  - Each context c ∈ C, is associated with a vector v_c
  - We seek vector representation for both v_w and v_c such that the dot-product associated with "good" word-context pair is maximized.
  - Same word has 2 different embeddings (as "word", as "context")

    $V_w(\text{Amsterdam}) \neq V_c(\text{Amsterdam})$
  - $P(w, c) = \log \sigma(Vw.Vc) + \sum_{i=1}^{k} \log \sigma(-Vw.Vc(i))$
  - k negative samples are taken for each true (w,c) pair, where, k is a hyperparameter.
  - Instead of changing all of the weights of thousands of observations each time, using only K of them increases computational efficiency.

Embedding with Arbitrary Contexts (Motivation)

- In Skip-gram model, contexts for a word w are the words surrounding it. The context vocabulary C is thus identical to the word vocabulary W. However, the model can be generalized to take arbitrary contexts.
- **Linear Bag-of-Words(BOW) Contexts** –
  - Uses a window of size K around the word
  - 2K contexts are produced – K before and after the target word
  - Ex – Australian scientist discovers star with telescope
    - If K=2, context for word *discovers* – *Australian, scientist, star, with.*
    - Missed important contexts like – *telescope.*
    - Included accidental context like – *Australian.*
    - Unmarked contexts – *discovers* is a context for both *stars and scientist.* This will result in both of them being neighbors in embedded space.
- K=5 will capture broad topical content.
- Smaller window size capture more focused information about the target word.

Dependency-Based Contexts –

- Derive contexts based on syntactic relations the word participates in.
- After parsing each sentence, context is derived as - for a target word w with modifiers $m_1,\dots,m_k$ and a head h, we consider the contexts $(m_1, lbl_1),\dots,(m_k, lbl_k),(h, lbl_h^{-1})$
- Where, lbl is the type of dependency relation between the head and the modifier (e.g. nsubj, dobj, prepwith, amod) and lbl^{-1} is used to mark the inverse-relation.
- Relations that include a preposition are "collapsed" prior to context extraction, by directly connecting the head and the object of the preposition, and subsuming the preposition itself into the dependency label.
- Example –

WORD	CONTEXTS
australian	scientist/amod^{-1}
scientist	australian/amod, discovers/nsubj^{-1}
discovers	scientist/nsubj, star/dobj, telescope/prep_with
star	discovers/dobj^{-1}
telescope	discovers/prep with^{-1}

- Syntactic dependencies are both more inclusive and more focused
- Captures relations that are far apart. (*telescope* for *discovers*)
- No fixed window size.
- Filters out coincidental contexts (*Australian* is not a context for *discovers*).
- Contexts are marked – *Stars* are objects of *discovers* and *scientists* are subjects.
- Therefore we expect syntactic contexts to yield more functional similarity and less topical similarity.

Experimental Setup

- Experimentation with 3 training conditions –
  - BOW5(window size = 5), BOW2(window size = 2), and DEPS(dependency based syntactic contexts).
- Word2vec was modified to support arbitrary contexts
- Negative sampling parameter(how many negative contexts to sample for every correct one) set to 15.
- Embeddings trained on English Wikipedia.
- For DEPS, the corpus was tagged with parts-of-speech using the Stanford tagger(Toutanova et al., 2003) and parsed into labeled Stanford dependencies (de Marneffe and Man-ning, 2008) using an implementation of the parser described in (Goldberg and Nivre, 2012).
- All tokens were converted to lowercase.
- Words and contexts that appeared less than 100 times were filtered.
- Resulted in - vocabulary of about 175,000 words, with over 900,000 distinct syntactic contexts.
- Results are reported on 300 dimension embeddings.

Qualitative Evaluation

- Bag-of-words contexts induce **topical** similarities
  - Contexts reflect the domain aspect.
  - Words that associate with w.
  - Generates meronyms
- Dependency contexts induce **functional** similarities
  - Share the same semantic type
  - Words that behave like w
  - Cohyponyms

Target Word	Bag of Words (k=5)	Dependencies
Hogwarts (Harry Potter's school)	Dumbledore hallows half-blood Malfoy Snape	Sunnydale Collinwood Calarts Greendale Millfield
	Related to Harry Potter	Schools

Qualitative Evaluation

- Bag-of-words contexts induce **topical** similarities
  - Contexts reflect the domain aspect.
  - Words that associate with w.
  - Generates meronyms
- Dependency contexts induce **functional** similarities
  - Share the same semantic type
  - Words that behave like w
  - Cohyponyms

Target Word	Bag of Words (k=5)	Dependencies
Turing (computer scientist)	nondeterministic non-deterministic computability deterministic finite-state	Pauling Hotelling Heting Lessing Hamming
	Related to computability	**Scientists**

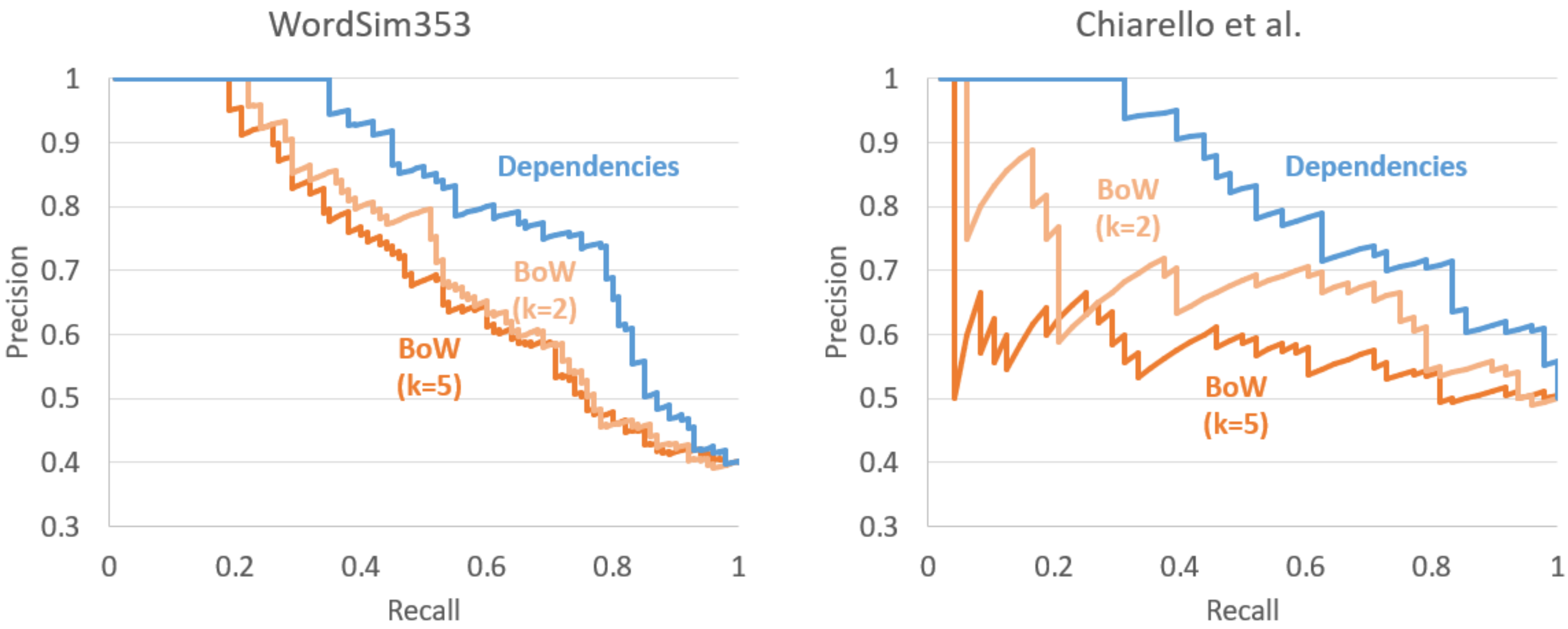
Qualitative Evaluation

- Bag-of-words contexts induce **topical** similarities
  - Contexts reflect the domain aspect.
  - Words that associate with w.
  - Generates meronyms
- Dependency contexts induce **functional** similarities
  - Share the same semantic type
  - Words that behave like w
  - Cohyponyms

Target Word	Bag of Words (k=5)	Dependencies
dancing (dance gerund)	singing dance dances dancers tap-dancing	singing rapping breakdancing miming busking
	Related to dance	**Gerunds**

Quantitative Evaluation

- WordSim353 dataset used (Finkelstein et al., 2002;Agirre et al., 2009).
  - Dataset contains pairs of similar words that reflect either relatedness(topical similarity) or similarity(functional similarity) relations.
- Embeddings used in a retrieval/ranking setup, where the task is to rank the similar pairs in the dataset above the related ones.
- Recall-precision curve is drawn that describes the embedding's affinity towards one subset ("similarity") over another ("relatedness").
- The experiment was repeated with a different dataset (Chiarello et al., 1990) that was used by Turney (2012) to distinguish between domain and functional similarities. The results show a similar trend.


WordSim353
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Precision
0
0.2
0.4
0.6
0.8
1
Recall
Dependencies
BoW
(k=2)
BoW
(k=5)
Chiarello et al.
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Precision
0
0.2
0.4
0.6
0.8
1
Recall
Dependencies
BoW
(k=2)
BoW
(k=5)

Model Introspection (Analyzing Embeddings)

- Neural words embeddings are considered opaque and uninterpretable.
- Skip-gram allows for a non-trivial amount of introspection.
- The DEPS model is queried for the contexts that are most activated by (have the highest dot product with) a given target word.
- By doing so, it can be seen what the model learned to be a good discriminative context for the word.
- 5 most activated contexts are listed.

Target Word	Dependencies
Hogwarts	students/prep_at^{-1} educated/prep_at^{-1} student/prep_at^{-1} stay/prep_at^{-1} learned/prep_at^{-1}

Model Introspection (Analyzing Embeddings)

- Neural words embeddings are considered opaque and uninterpretable.
- Skip-gram allows for a non-trivial amount of introspection.
- The DEPS model is queried for the contexts that are most activated by (have the highest dot product with) a given target word.
- By doing so, it can be seen what the model learned to be a good discriminative context for the word.
- 5 most activated contexts are listed.

Target Word	Dependencies
Turing	$machine/nn^{-1}$ $test/nn^{-1}$ $theorem/poss^{-1}$ $machines/nn^{-1}$ $tests/nn^{-1}$

Model Introspection (Analyzing Embeddings)

- Neural words embeddings are considered opaque and uninterpretable.
- Skip-gram allows for a non-trivial amount of introspection.
- The DEPS model is queried for the contexts that are most activated by (have the highest dot product with) a given target word.
- By doing so, it can be seen what the model learned to be a good discriminative context for the word.
- 5 most activated contexts are listed.

Target Word	Dependencies
dancing	dancing/conj dancing/$conj^{-1}$ singing/$conj^{-1}$ singing/conj ballroom/nn

Observations

- The most discriminative contexts in these cases are not associated with subjects or objects of verbs.
- They are rather associated with conjunctions, appositions, noun-compounds and adjectival modifiers.
- The collapsed preposition relation is very useful (ex - for capturing the school aspect of Hogwarts).
- The presence of many conjunction contexts, such as superman/conj for batman and singing/conj for dancing, may explain the functional similarity observed; conjunctions in natural language tend to enforce their conjuncts to share the same semantic types and inflections.

My Opinion

- Peek into the embeddings from DEPS was insightful.
- Should look into words with multi contexts. Ex – Apple, Orange.
- How does the model perform in comparison to DCBOW or LSTMS where word order matters and other advanced neural/embedding models.
- How does the model perform for certain applications like classification?
- Dependency-based word embeddings excel at predicting brain activation patterns. (Samira Abnar, 2018)
- Limitation - has only explored only English-tailored Stanford dependency scheme.
- Are Universal Dependencies, which are less tailored to English, actually better or worse than the English-specific labels and graphs? (Sean MacAvaney, 2018)
- Comparison of cross-lingual embeddings using this model

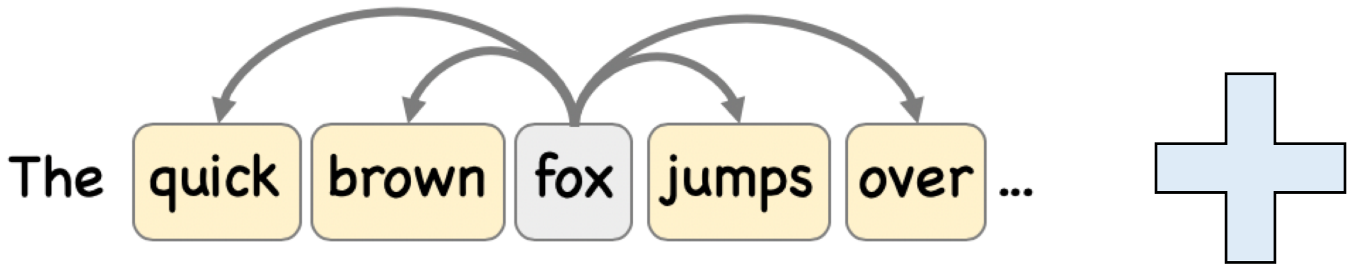
Conclusion

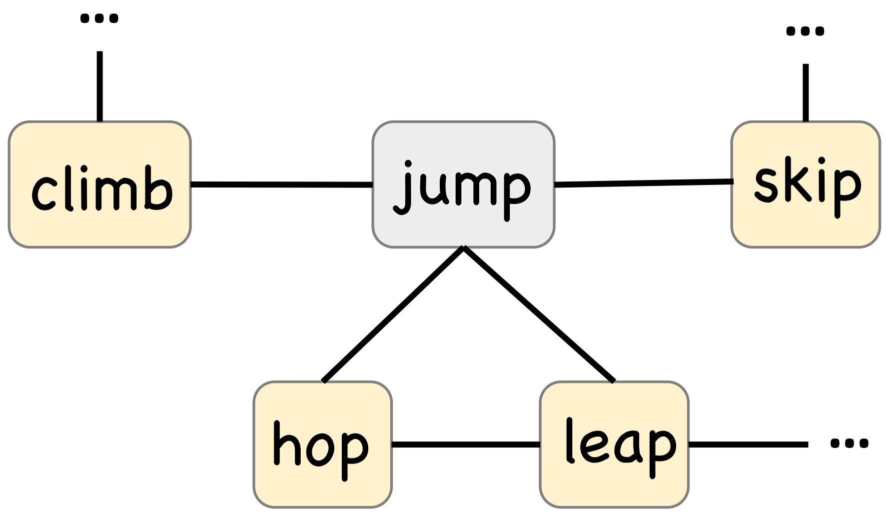
- Generalized Skip-Gram with Negative Sampling to arbitrary contexts.
- Different contexts induce different similarities.
- Suggested a way to peek inside the black box of embeddings.
- Future work –
  - Insights from model introspection will help in development of better contexts.
  - Figuring out why the subject and object relations are absent and finding how their contribution can be increased.
  - Using the information to develop better task specific embedded representations.

Retrofitting Word Vectors to Semantic Lexicons

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, Noah A. Smith

Statistical Methods in Natural Language Semantics
Presentation by Phillip Lippe

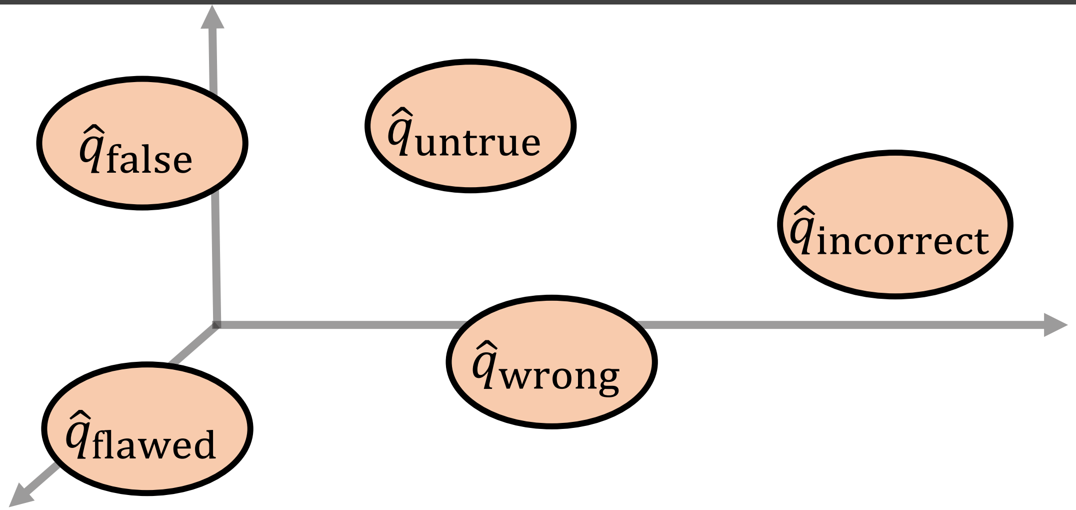




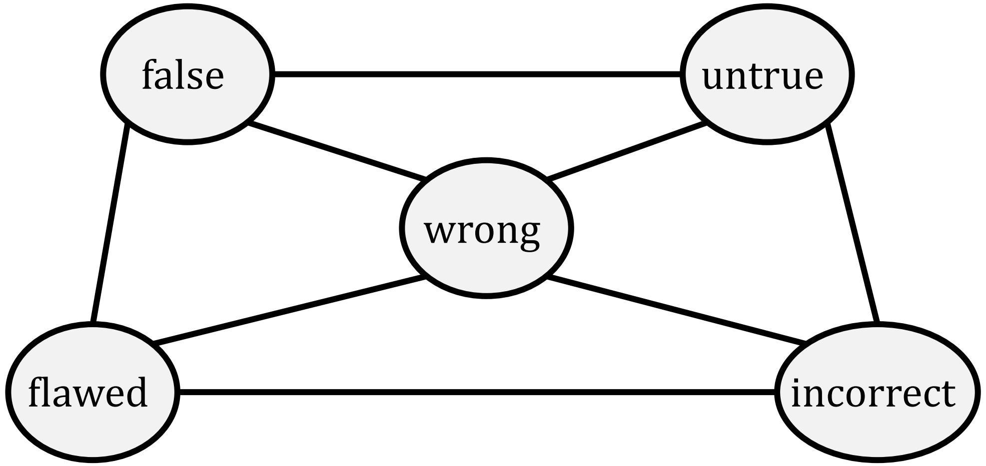
Motivation

Text corpus

Distributional vector model



Semantic Lexicon
WordNet synonyms



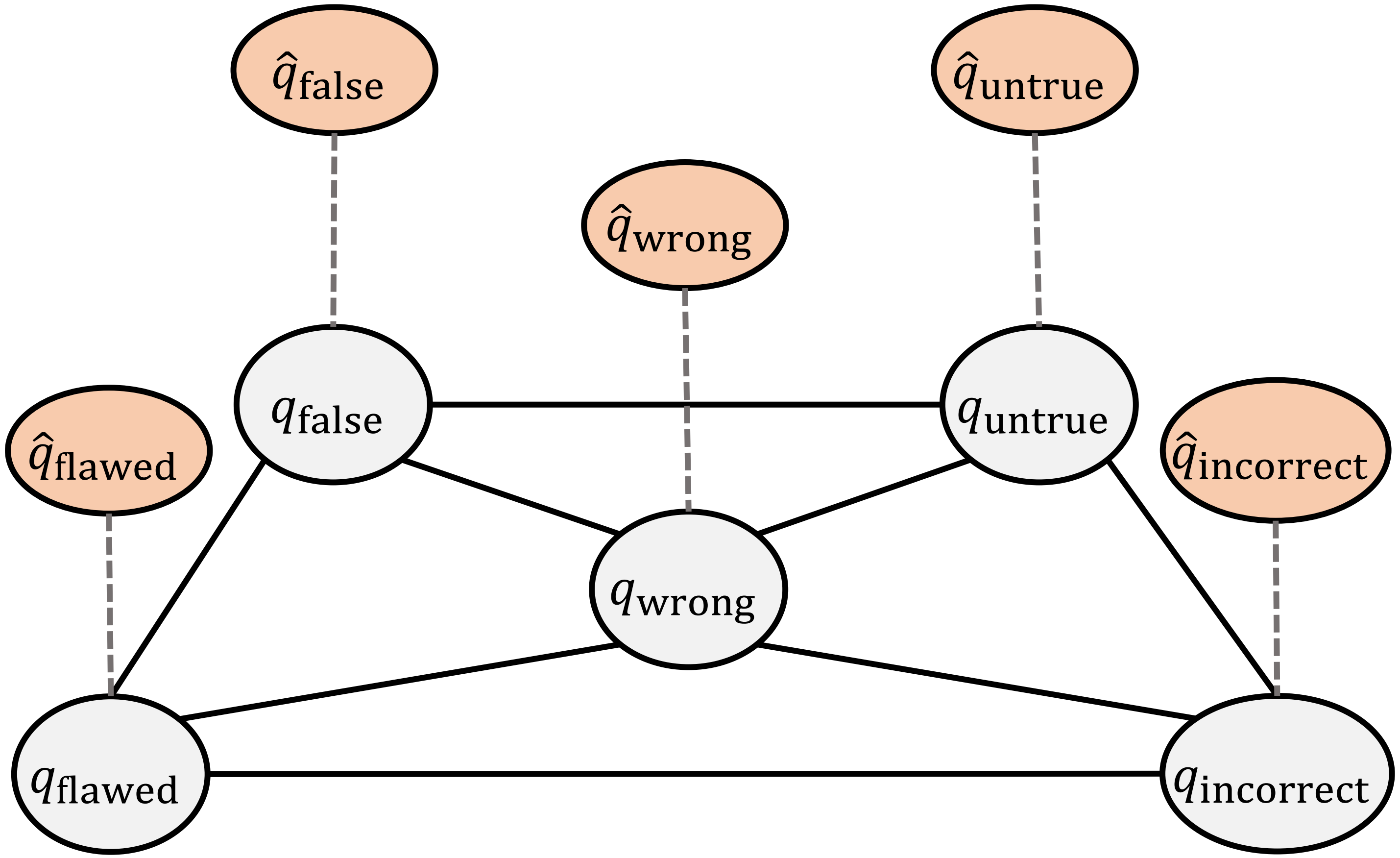
Motivation

- Why should be bother at all about semantic lexicons?
- Distributional vector models learn by maximizing probability of co-occurrences, not word relations
- Improve/estimate representations of infrequent/unseen words
- Examples
  - **Natural Language Inference**: pairwise synonym/antonym relation can already indicate label

Premise: *A lady standing in a wheat field*

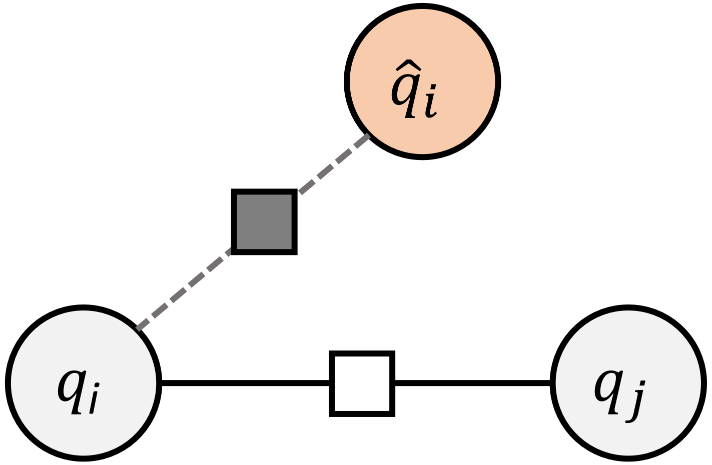
Hypothesis: *A person standing in a corn field.*

Motivation



$\hat{q}_{\text{false}}$
$\hat{q}_{\text{untrue}}$
$\hat{q}_{\text{wrong}}$
q_{false}
q_{untrue}
$\hat{q}_{\text{flawed}}$
$\hat{q}_{\text{incorrect}}$
q_{wrong}
q_{flawed}
$q_{\text{incorrect}}$

Retrofitting

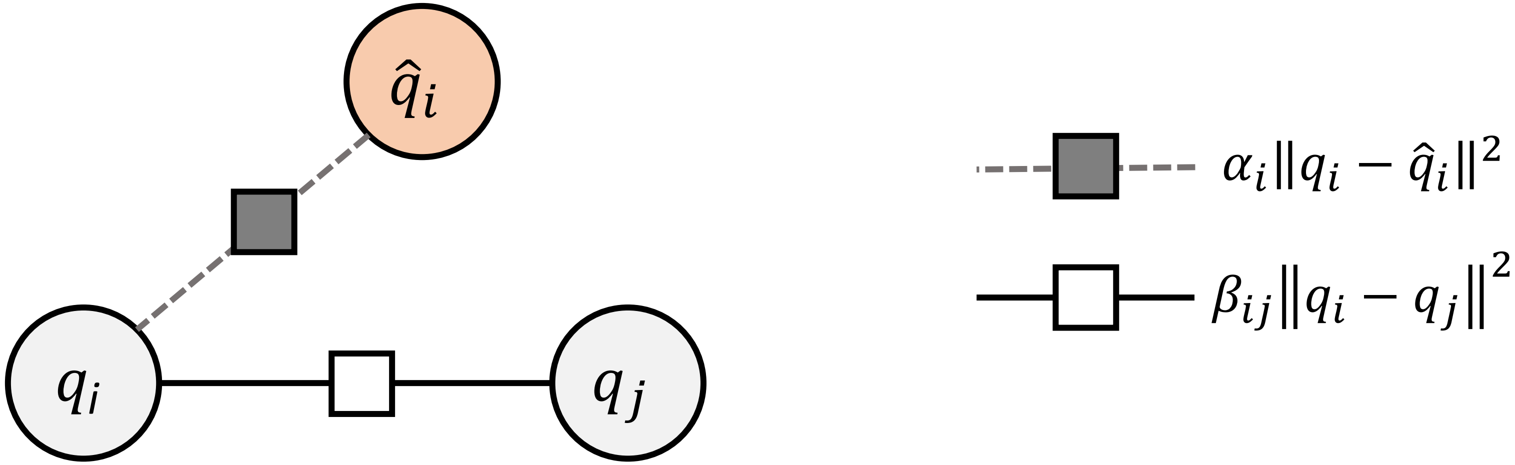


$\alpha_i \|q_i - \hat{q}_i\|^2$

$\beta_{ij} \|q_i - q_j\|^2$

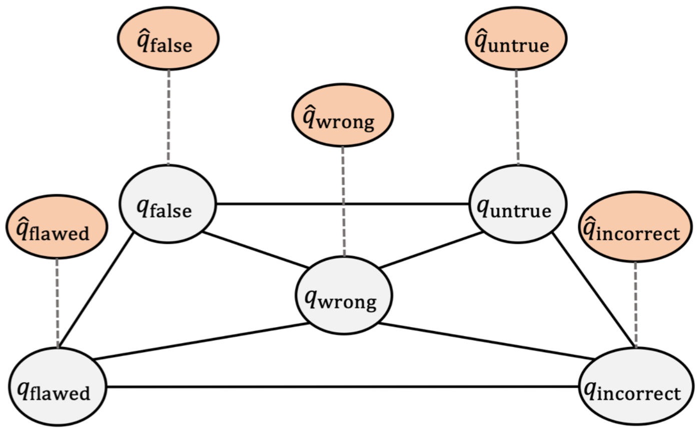
Objective function: $\Psi(Q) = \sum_{i=1}^{n} \Big[\quad + \sum_{(i,j) \in E} \quad \Big]$

Retrofitting



Objective function: $$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

Retrofitting



Objective function: $\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \| q_i - \hat{q}_i \|^2 + \sum_{(i,j)\in E} \beta_{ij} \left\| q_i - q_j \right\|^2 \right]$

Optimization: $\frac{\partial \Psi(Q)}{\partial q_i} = 0 \implies q_i = \frac{\alpha_i \hat{q}_i + \sum \beta_{ij} q_j}{\alpha_i + \sum \beta_{ij}}$

- Optimization by iteratively updating vectors till convergence
- Hyperparameters α_i and β_{ij} balance the influence of neighbors and distributional representation

Related Work

- Previous work focused on using semantic lexicons as prior information during training
- Adjusting objective function of distributional vector model by for example:

$$p(Q) \propto \exp\left(-\gamma \sum_{i=1}^{n} \sum_{j:(i,j)\in E} \beta_{ij} \left\| q_i - q_j \right\|^2\right)$$

- Retrofitting has two important advantages
  - Post-processing step ⇒ no need to re-train representations, done in seconds
  - Modular approach ⇒ applicable to any vector space model

Semantic Lexicons

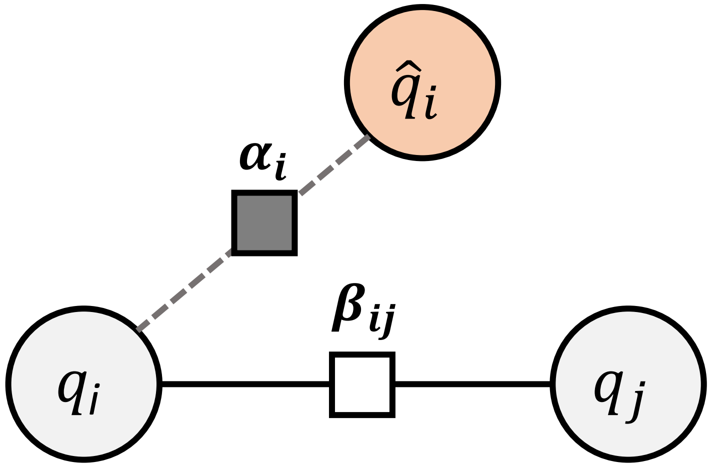
- Paraphrase Database **PPDB**
  - Two words that are translated to the same word in a different language, are synonyms
  - Example: *incorrect* and *wrong*
  - 100k words, 375k edges

- **WordNet**
  - Groups English words into sets of synonyms (*synsets*)
  - Contains additional relations such as hypernyms and hyponyms
  - 150k words, 300k synonym edges (WN_{syn}), 935k edges overall (WN_{all})

Semantic Lexicons

- **FrameNet**
  - Containing information about lexical and predicate-argument semantics
  - Example: frame *Cause_change_of_position_on_a_scale* contains 26 words including *push, raise* and *growth*
  - 10k words, 420k edges

- Hyperparameters $\alpha_i = 1$ and $\beta_{ij} = degree(i)^{-1}$ for every lexicon



Experiments

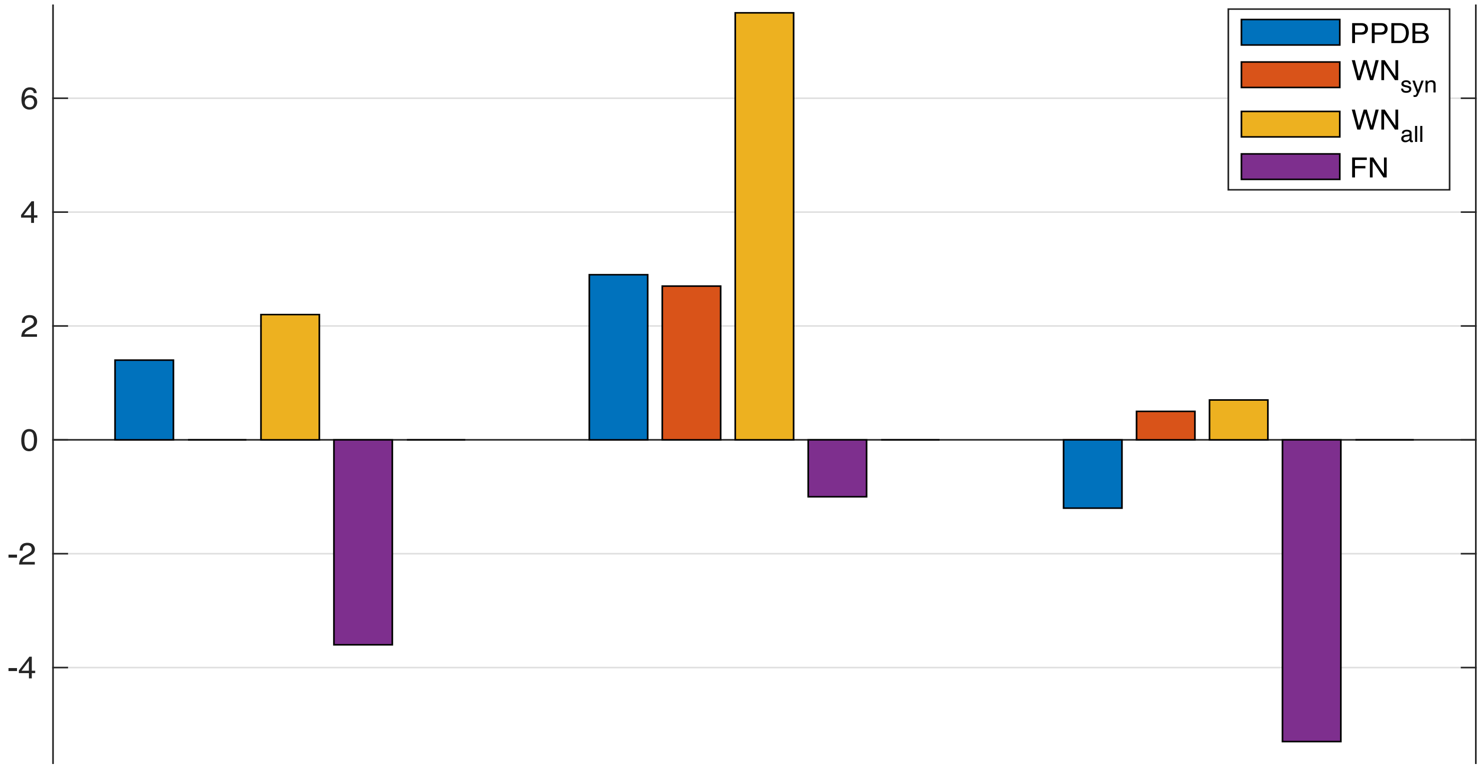
- Word Similarity
- Syntactic relations
- Synonym selection
- Sentiment Analysis
- Multilingual Evaluation
- Vector length dependency

Word Similarity

- Comparing similarity between words in vector space to human intuition
- Datasets
  - **WS-353**: 353 words pairs with human annotated similarity ratings
  - **RG-65**: 65 pairs of nouns for which the similarity of meaning is rated on a scale of 0 to 4
  - **MEN**: 3,000 frequent word pairs, ranked by humans which word pairs are most similar
- Metric
  - Compute *Spearman's rank correlation coefficient* to measure the difference in rankings

Word Similarity

Improvements for Glove word embeddings

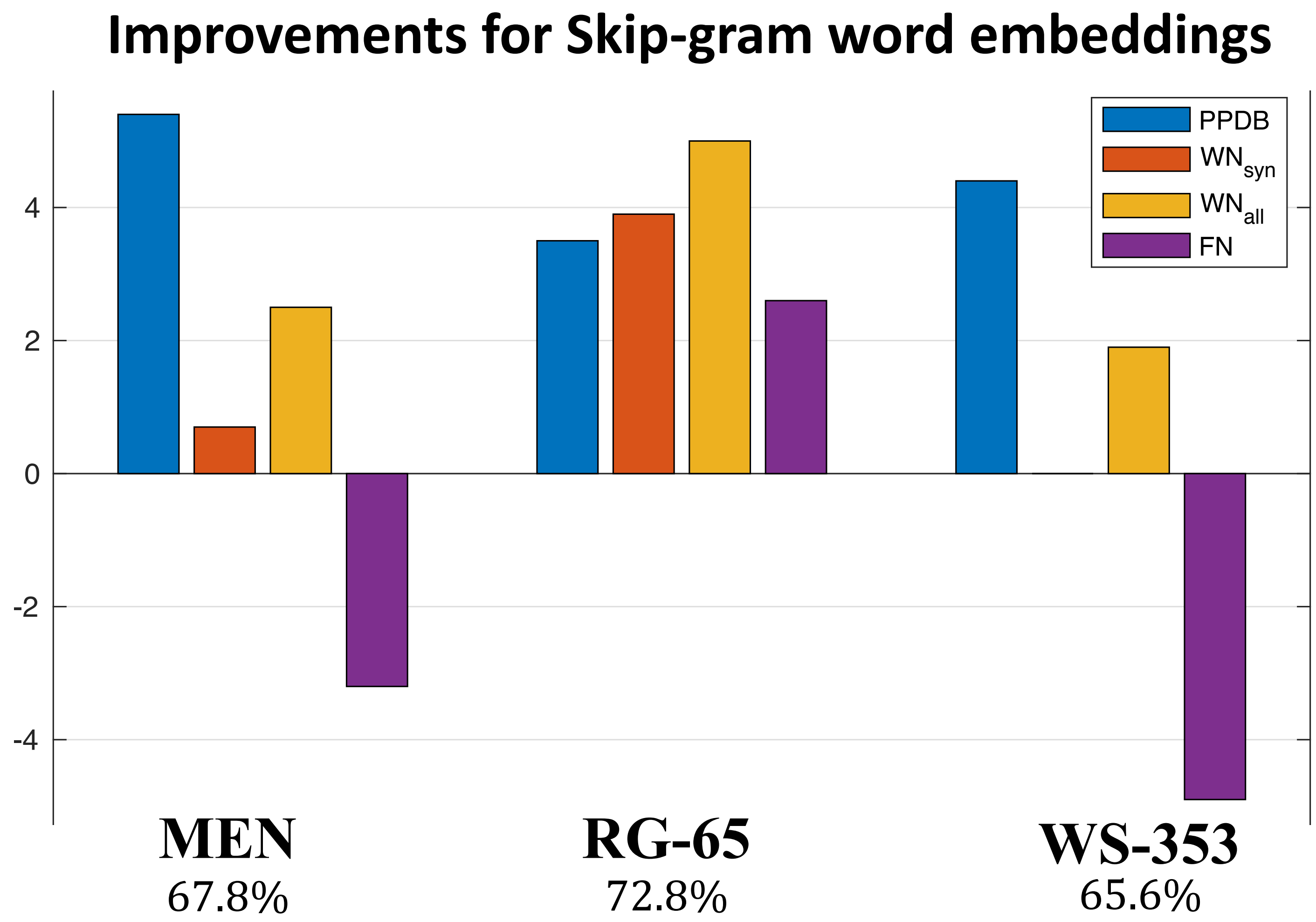


MEN 73.7%

RG-65 76.7%

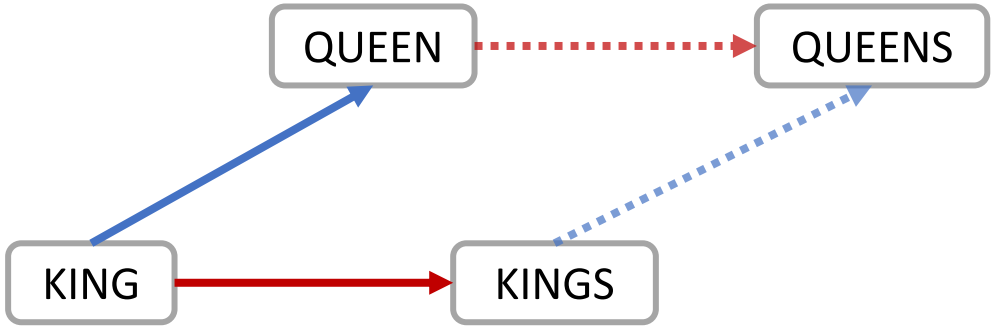
WS-353 60.5%

Word Similarity



Syntactic Relations

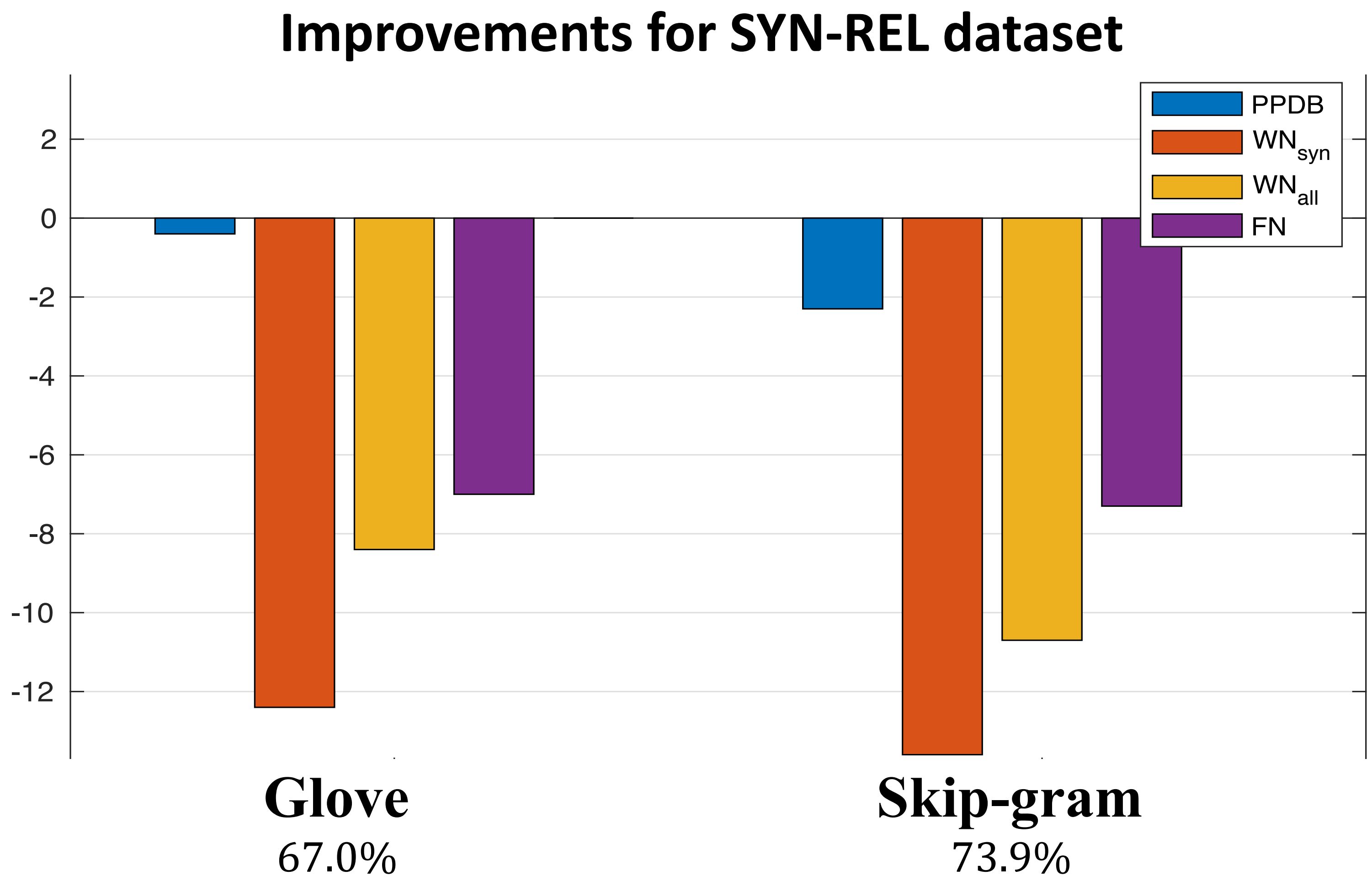
- Testing the encoding of syntactic relations in the representations
- Dataset
  - Contains pairs of tuples of word relations that follow a common syntactic relation



$$q_{\text{QUEEN}} - q_{\text{KING}} + q_{\text{KINGS}} = ?$$

  - 9 different kinds of relation for 10k pairs
- Metric
  - Accuracy of finding the right word *d*

Syntactic Relations



Conclusion

- Retrofitting is a simple method to combine distributional vector models with semantic lexicons
- Post-processing step, can be applied to any distributional vector space model
- Focuses rather on semantical information than syntactical
- Improvements can be up to or better than approaches that incorporate lexicons during training
- Performance highly depends on lexicon and task. Best lexicon across tasks was PPDB
    - But: word vectors can easily be adopted for specific task by Retrofitting

Post-processing vs. Prior

- Post-processing is efficient and fast, **but** might not be optimal

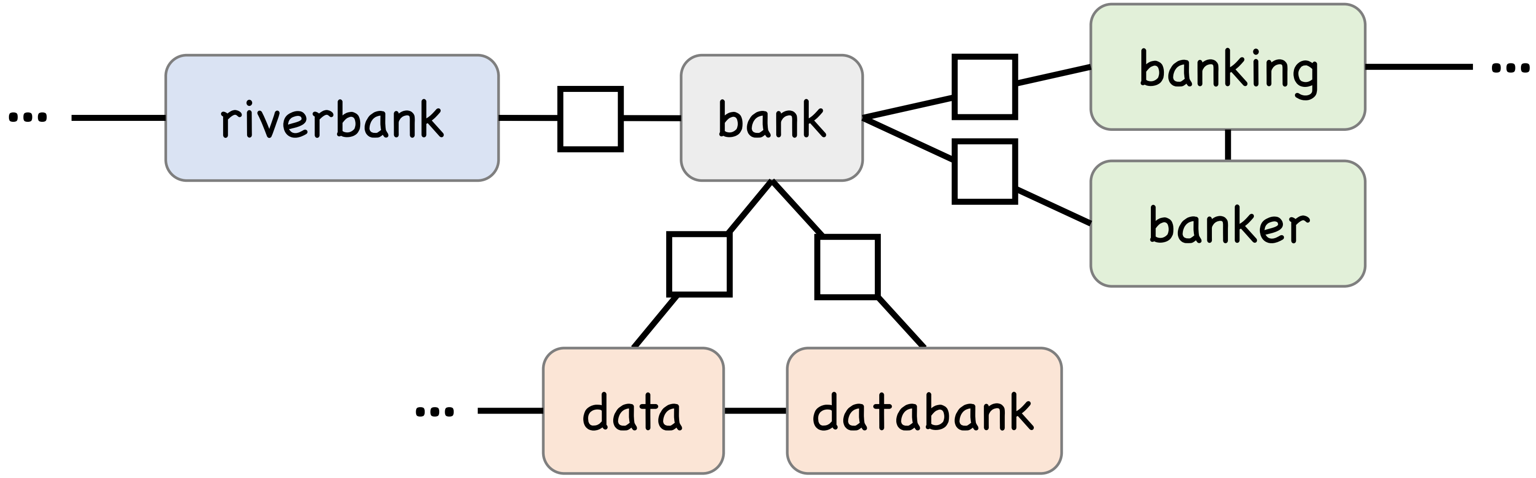
Any costs incurred due to incorrect *settings will be borne by you.*

Applying the wrong *hair care can lead to extensive damage.*

- The representation of a word is influenced by relations of context words
- Hard to integrate in post-processing method

Multiple Meanings

- Words with multiple meanings have synonyms specifically for a certain sense



- Retrofitted vector is a weighted average between meanings (based on number of synonyms)

Similarity measurement

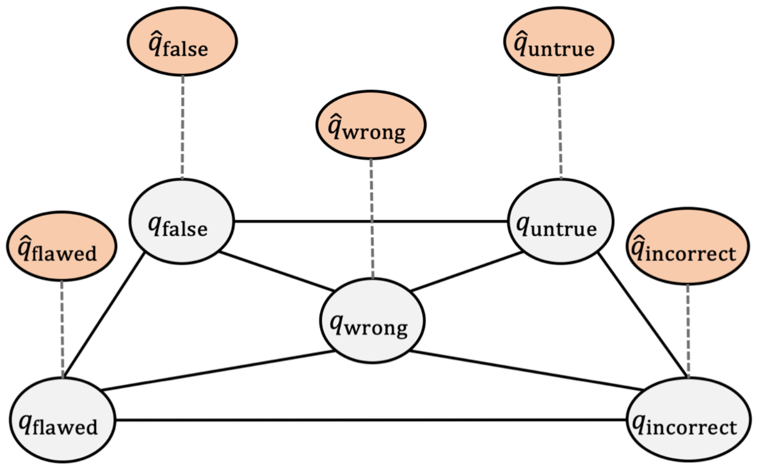
- Similarity between two vectors is measured by Euclidean distance
    - Semantic lexicons contain more than pure synonyms
    - How to deal with other relations correctly (antonyms, hypernyms, …)?

Future work

- Counterfitting (Mrkšić et. al, 2016): extend Retrofitting by pushing antonyms as far away as possible
- ATTRACT-REPEL (Mrkšić et. al, 2017): learn similarities from mono- and cross-lingual relations
- Explicit retrofitting (Glavaš and Vulić, 2018):
  - Learn mapping function as neural network to retrofit vectors for relations (synonyms, antonyms,…)
- Extrofitting (Jo and Choi, 2018):
  - Expanding word vectors by additional dimensions encoding semantic knowledge
  - Reduce vector space to original dimensions by Linear Discriminant Analysis

Thank you for your attention!

Questions?



Specializing Word Embeddings for Similarity or Relatedness

Authors : Douwe Kiela, Felix Hill and Stephen Clark

Presenter: Sohi Sudhir

Introduction and a quick recap

Distributional Hypothesis : Words occurring in similar contexts have similar meanings.

Word Embeddings: Vector representations of words

Why are word embeddings so famous?

They are ***'general purpose'***

But, not all neural embeddings are born equal! (Hill et al., 2014).

Similarity vs Relatedness

“Genuine” similarity vs “Associative” similarity

SIMILARITY	RELATEDNESS
Car - Automobile	Car - Petrol
Cat - Animal	Cat - Dog
Chair - Seat	Table - Chair

- Embeddings are both similar and related but not perfect at either (due to distributional hypothesis).
- In NLP, semantic spaces are evaluated on how well both the aspects are captured.
- However, they are both **mutually incompatible**.

Similarity vs Relatedness

<u>Machine Translation</u>

'Cat' is related to 'dog'. Does this mean the translation of cat is 'cien'?

SIMILAR WORDS ARE MORE IMPORTANT

<u>Document Classification</u>

Knowing dog and cat are associated is more informative than knowing canine is a synonym of dog.

RELATED WORDS ARE MORE IMPORTANT

The idea : Specialise word embeddings

How to specialize?

- **Nudge** the embeddings in a particular direction by learning from **task related additional semantic sources**.
  - MyThes thesaurus which contains ***synonyms*** for almost 80,000 words.
  - USF(University of South Florida) free association norms which ***contain scores for free association*** of over 10,000 concept words.
- **Specializing for similarity** : Train from both a corpus and MyThes
- **Specializing for relatedness**: Train from both a corpus and USF free norms
- Raw text taken from a dump of English Wikipedia plus newswire text (8 billion words)

Specialise Word Embeddings

Methods

- Joint Learning
- Retrofitting (Faruqui et al)
- Skip - gram retrofitting

Evaluation

- **Intrinsic Evaluation**:
  - SimLex-999: Similarity
  - MEN : Relatedness
- **Extrinsic(downstream) Evaluation**:
  - TOEFL Synonym Test
  - Document Classification (based on Reuters Corpus Volume1)

Methods : Joint Learning

- **Joint Learning**: Training multiple sub-tasks together

Training objective of a standard skipgram

$w_1 \ldots w_T$ Sequence of training words

c context size

U_w and v_w: context and target vector representations for word w

$$\frac{1}{T}\sum_{t=1}^{T} J_\theta(w_t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c} \log p(w_{t+j}|w_t)$$

where $p(w_{t+j}|w_t)$ is obtained via the softmax:

$$p(w_{t+j}|w_t) = \frac{\exp^{u_{w_{t+j}}^\top v_{w_t}}}{\sum_{w'} \exp^{u_{w'}^\top v_{w_t}}}$$

2 conditions

Sampling condition

W^a is uniformly sampled from a set of additional contexts A_{wt}.

$$\frac{1}{T}\sum_{t=1}^{T}\left(J_\theta(w_t) + [w^a \sim \mathcal{U}_{A_{w_t}}] \log p(w^a|w_t)\right)$$

All condition

The set of additional contexts A_{wt} contains the relevant contexts for a word w_t.

$$\frac{1}{T}\sum_{t=1}^{T}\left(J_\theta(w_t) + \sum_{w^a \in A_{w_t}} \log p(w^a|w_t)\right)$$

Methods : Retrofitting

- **Retrofitting** is a post-processing step which can be used on pre-trained word vectors obtained using ***any*** vector training model.
- Original paper's (Faruqui et al. (2015)) approach : Graph-Based retrofitting
- **Skip-gram Retrofitting:**
  - 1st stage: Train a standard skip-gram model
  - 2nd stage: Learn from additional contexts
- All embeddings have 300 dimensions

First phase: standard skip-gram

$$\frac{1}{T}\sum_{t=1}^{T} J_{\theta}^{1}(w_t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c} \log p(w_{t+j}|w_t)$$

Second phase: additional context skip-gram

$$\frac{1}{T}\sum_{t=1}^{T} J_{\theta}^{2}(w_t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{w^a\in A_{w_t}} \log p(w^a|w_t)$$

Results : Intrinsic Evaluation

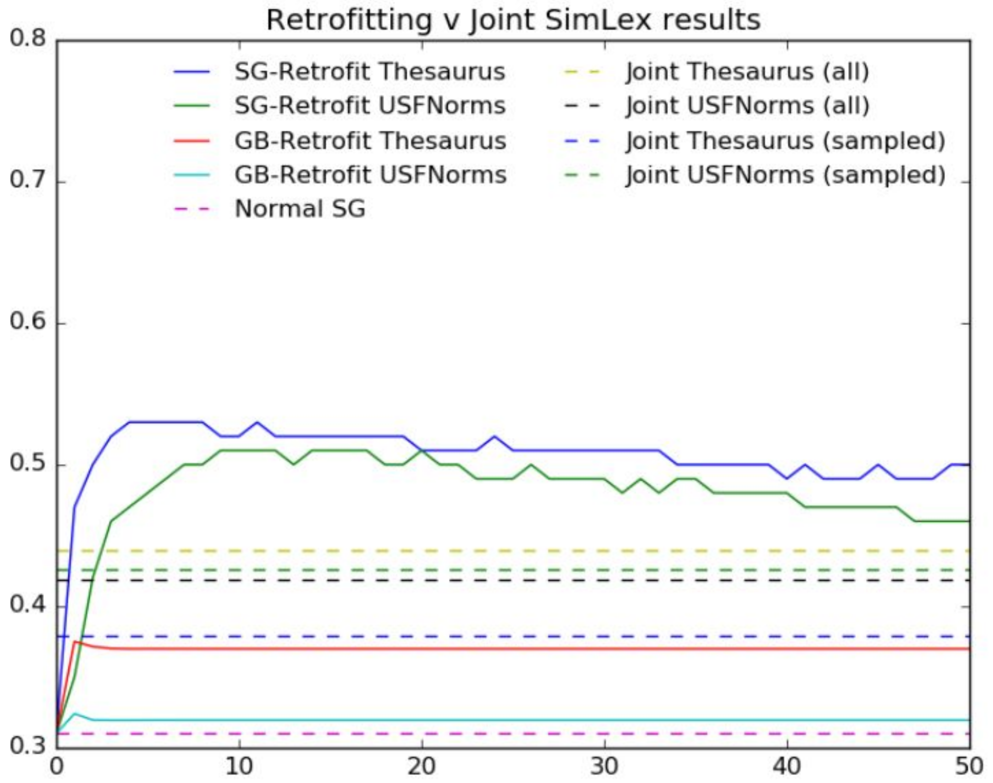
Results on 1 iteration

Method	SimLex-999	MEN
Skip-gram	0.31	0.68
Fit-Norms	0.08	0.14
Fit-Thesaurus	0.26	0.14
Joint-Norms-Sampled	0.43	**0.72**
Joint-Norms-All	0.42	0.67
Joint-Thesaurus-Sampled	0.38	0.69
Joint-Thesaurus-All	0.44	0.60
GB-Retrofit-Norms	0.32	0.71
GB-Retrofit-Thesaurus	0.38	0.68
SG-Retrofit-Norms	0.35	0.71
SG-Retrofit-Thesaurus	**0.47**	0.69

Interesting observations

- SG-Retrofit-Thesaurus works best on SimLex
- Joint-Norms-Sampled works best on MEN
- Sampling a single free associate works best for relatedness.
- Presenting all additional contexts (all synonyms) works best for similarity.

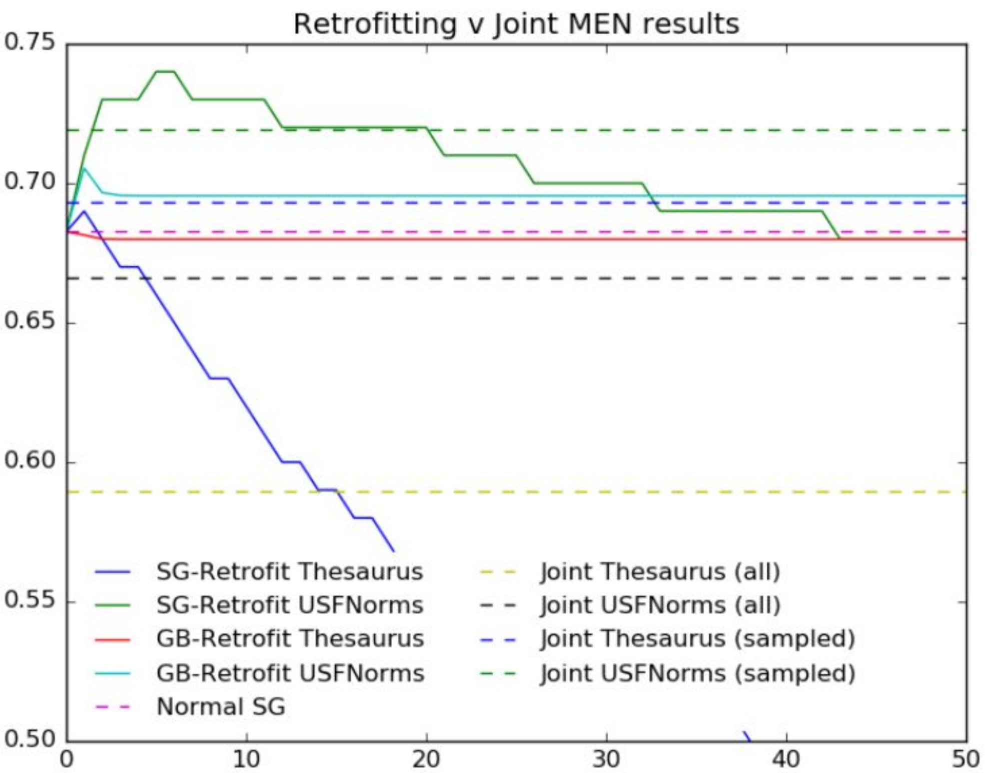
Results on multiple iterations



Interesting observations

- SG-Retrofit Thesaurus works the best.
- Too many iterations lead to overfitting.
- Highest performance at 5 iterations (the then, current state of art)

Results on multiple iterations

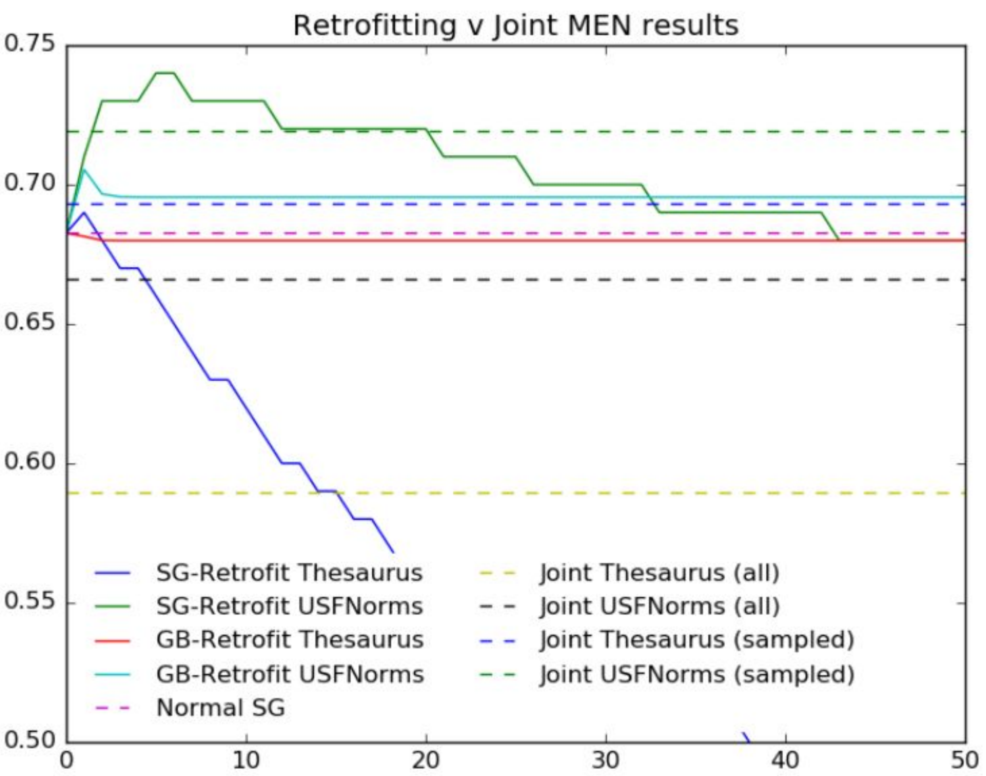


Interesting observations

- Overall effect not very clear.
- Joint learning performs better (after the 2-10 iteration margin).
- Performance of similarity goes down(SG-Retrofit Thesaurus).

- Too many iterations lead to overfitting.

Results on multiple iterations



Interesting observations

- Overall effect not very clear.

(after

Embeddings gettings dragged away from relatedness and towards similarity!

s

down(SG-Retrofit Thesaurus).

- Too many iterations lead to overfitting.

Results : Downstream Tasks

TOEFL SYNONYM TASK

Method	TOEFL		Doc
Skip-gram	77.50		83.96
Joint-Norms-Sampled	78.75		84.46
Joint-Norms-All	66.25		**84.82**
Joint-Thesaurus-Sampled	81.25		83.90
Joint-Thesaurus-All	80.00		83.56
GB-Retrofit-Norms	80.00		80.58
GB-Retrofit-Thesaurus	83.75		80.24
SG-Retrofit-Norms	80.00		84.56
SG-Retrofit-Thesaurus	**88.75**		84.55

Interesting observations

- SG-Retrofit-Thesaurus works best on the TOEFL test (also did on SimLex).
- It clearly outperforms standard skipgram model.

Document Classification task

Method	TOEFL		Doc
Skip-gram	77.50		83.96
Joint-Norms-Sampled	78.75		84.46
Joint-Norms-All	66.25		**84.82**
Joint-Thesaurus-Sampled	81.25		83.90
Joint-Thesaurus-All	80.00		83.56
GB-Retrofit-Norms	80.00		80.58
GB-Retrofit-Thesaurus	83.75		80.24
SG-Retrofit-Norms	80.00		84.56
SG-Retrofit-Thesaurus	**88.75**		84.55

Interesting observations

- Joint-Norms-All is the best performing model.
- Relatedness-specialized embeddings perform better on this task than similarity embeddings.
- It clearly outperforms standard skipgram model.

Observations

- Joint learning works better with relatedness (additional free associates).

- Skip-gram retrofitting works better with similarity (additional thesaurus information).

WHY?

Curriculum Learning (Bengio et al. (2009))

- Thesaurus has synonyms : uncommon words (less frequency, more advanced).
- USF Norms mostly has common words (high frequency, less advanced)

- Advanced words can be detrimental to the model.
    - Retrofitting
    - Thesaurus
- Less advanced words can be learned together
    - Joint Learning
    - USF Norms

Method	TOEFL		Doc
Skip-gram	77.50		83.96
Joint-Norms-Sampled	78.75		84.46
Joint-Norms-All	66.25		**84.82**
Joint-Thesaurus-Sampled	81.25		83.90
Joint-Thesaurus-All	80.00		83.56
GB-Retrofit-Norms	80.00		80.58
GB-Retrofit-Thesaurus	83.75		80.24
SG-Retrofit-Norms	80.00		84.56
SG-Retrofit-Thesaurus	**88.75**		84.55

But aren’t the differences too small?

Method	TOEFL		Doc
Skip-gram	77.50		83.96
Joint-Norms-Sampled	78.75		84.46
Joint-Norms-All	66.25		**84.82**
Joint-Thesaurus-Sampled	81.25		83.90
Joint-Thesaurus-All	80.00		83.56
GB-Retrofit-Norms	80.00		[illegible]
GB-Retrofit-Thesaurus	83.75		[illegible]
SG-Retrofit-Norms	80.00		[illegible]
SG-Retrofit-Thesaurus	**88.75**		[illegible]

Every percentage point is worth more than 100 documents. The dataset has more than 10,000 documents.

But aren’t the differences too small

Personal Views/Observations

- Observation : Similarity works well for relatedness but it does not work the other way round.
- Pro: The difference carries on to downstream NLP tasks is a major strength.
- Con: It would be better to have a common embedding rather than different embeddings for different tasks. (Maybe concatenate similarity and relatedness?)
- Con: Dependent on a semantic source (reliable? available?)

Personal Views/Observations

- Con: Joint learning can be expensive as it requires adapting to the underlying model.
- Con (as discussed) : No statistical test to prove conclusions
- To think about: The method of document-level representation is taken by the sum of all embeddings. Does it really capture the true representation of the document?
- To think about: Why does SG-Retrofitting thesaurus work worse than GB-Retrofitting thesaurus ?

Conclusion

- Specialized embeddings outperform standard embeddings by a large margin on intrinsic similarity and relatedness evaluations.
- Difference in how embeddings are specialized carries to downstream NLP tasks.
- Performance could be improved even further by going over several iterations of the semantic resource (In retrofitting)
- **Future work:**
  - Making embeddings general purpose (concatenation?)
  - Making learning independent of semantic source

Thank you! Questions?