Improving Word and Sense Embedding with Hierarchical Semantic Relations

Yow-Ting Shiue, Department of Computer Science and Information Engineering, National Taiwan University

Wei-Yun Ma, Institute of Information Science, Academia Sinica

Slides: http://goo.gl/LWzZWZ

Outline

I. Introduction

II. Methods

  A. Sense Vectors

  B. Word Vectors

III. Experimental Results and Analysis

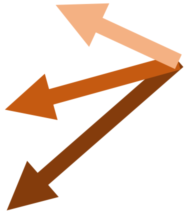
  Intrinsic Evaluation

  Extrinsic Evaluation

IV. Conclusion

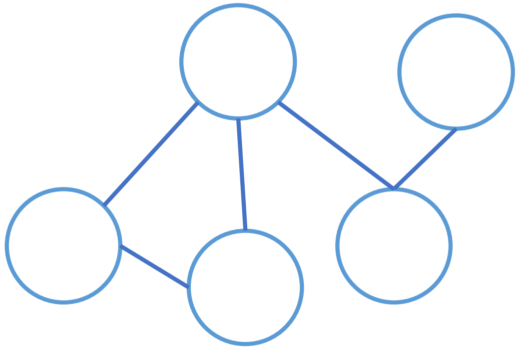
I. Introduction

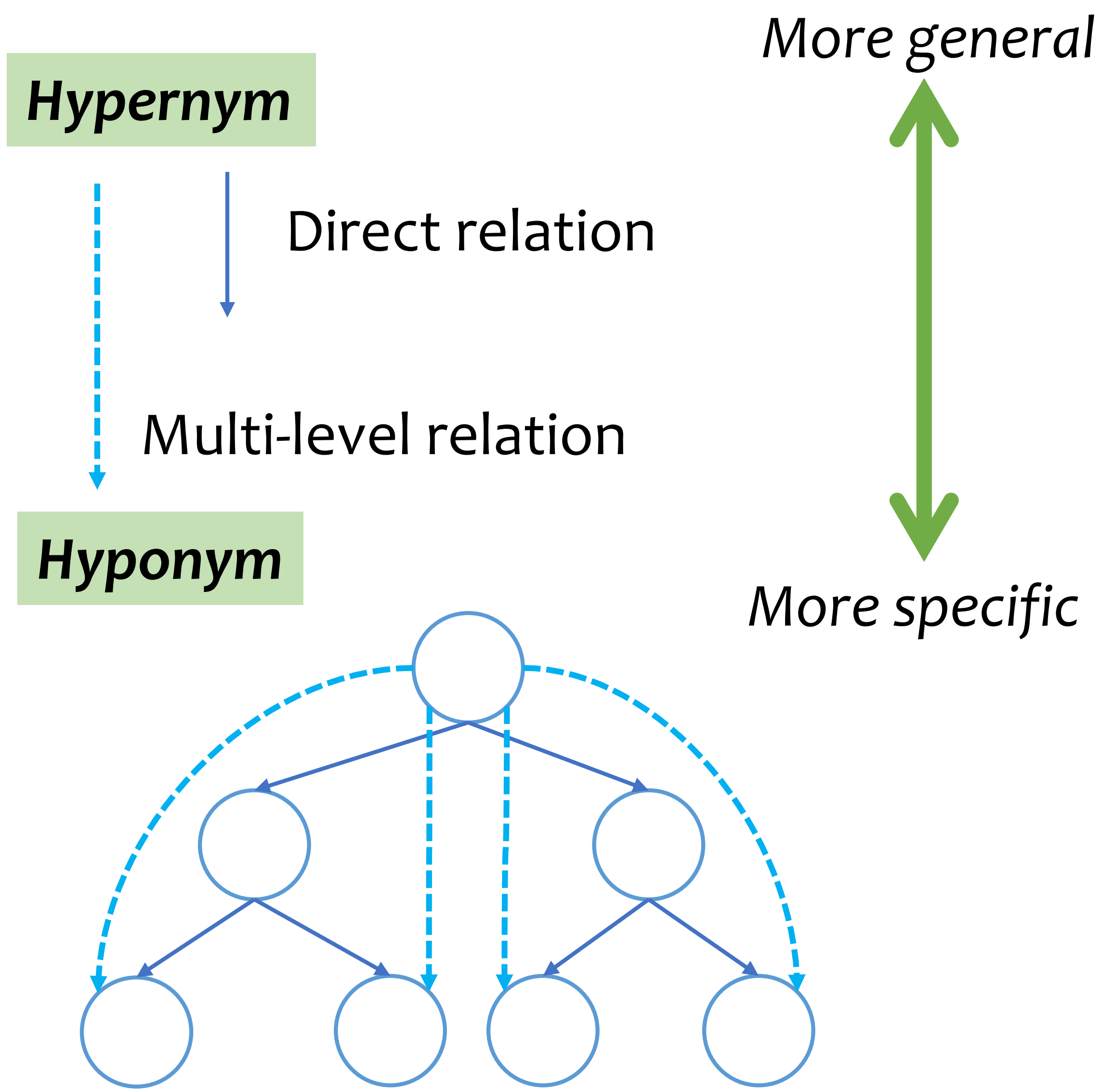
Distributed Representations



+

Lexical Resource





WordNet: hierarchical resource

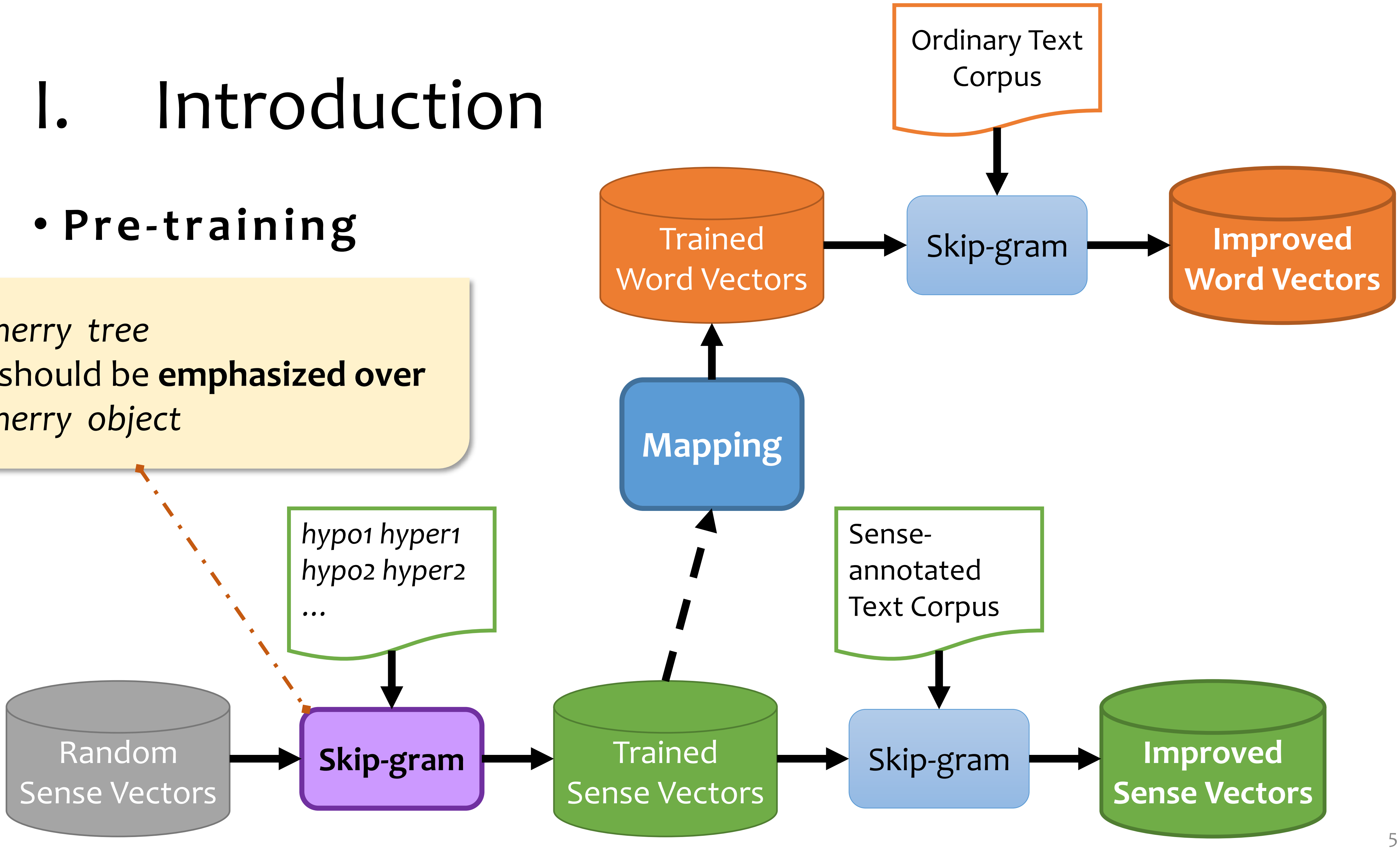
I. Introduction

Distributional **Word** Embeddings	Word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), …
+ Lexical Resource	• JointRCM (Yu and Dredze, 2014): synonym • Retrofitting (Faruqui et al., 2015): synonym + **single-level** hypernym/hyponym • Semantic word embeddings (Liu et al., 2015): synonym + antonym + hyponym/hypernym • Constrained optimization problem • Strength of constraints does not vary with # levels
Distributional **Sense** Embeddings + Lexical Resource	• AutoExtend (Rothe and Schutze, 2015): Learn synset embeddings from trained word embeddings • SensEmbed (Iacobacci et al., 2015): Utilize relations in BabelNet when computing word similarities, **not updating** vectors

I. Introduction

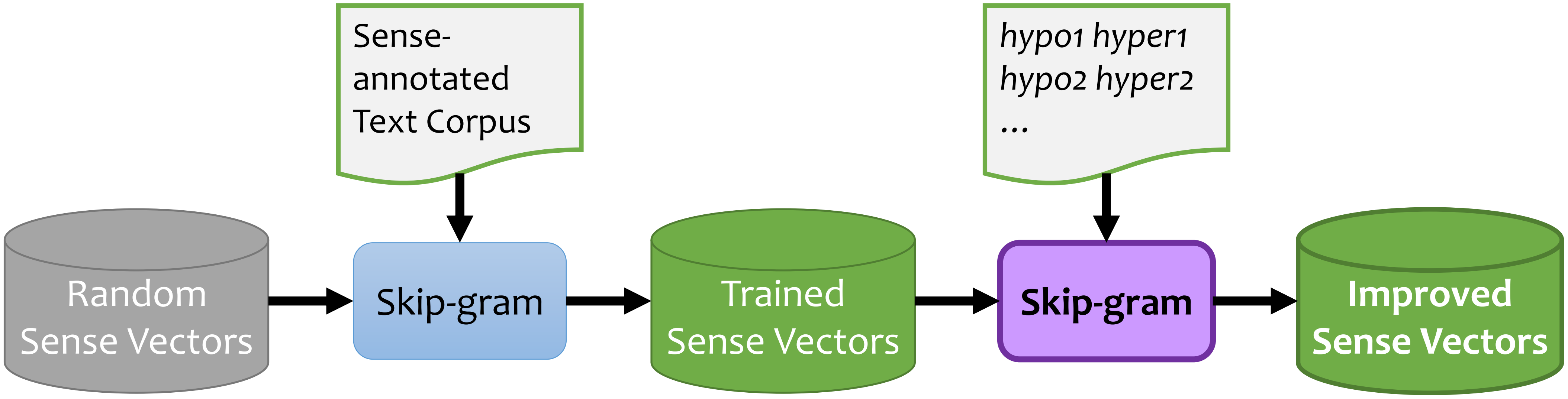
- **Pre-training**

cherry tree
should be **emphasized over**
cherry object

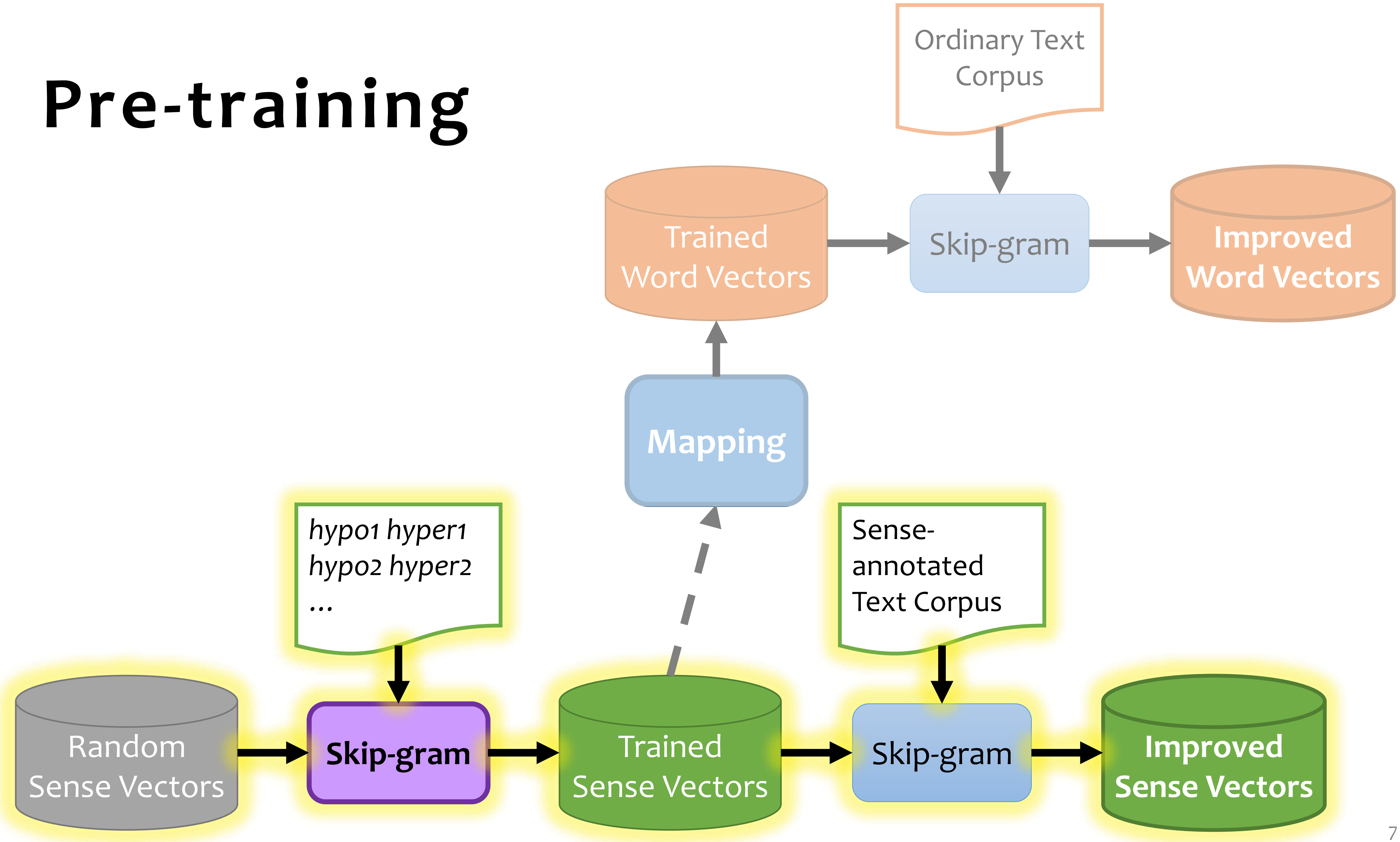


I. Introduction

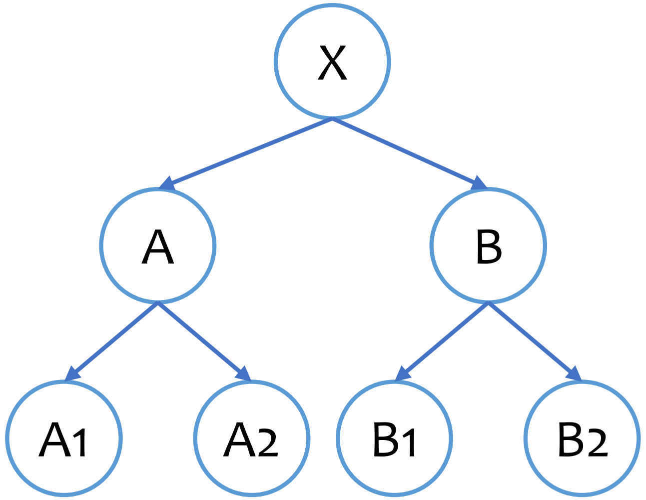
- **Post-processing**
  - Reverse the order of training on sense-annotated corpus and hyponym-hypernym relations
  - Relations are sense level → cannot perform post-processing after expending sense vectors to word vectors



Pre-training



II. Methods – Sense Vectors



- Hierarchical relation: hyponym-hypernym
  - Reflects an organized hierarchy of concepts
  - Direct: (X, A), (X, B), (A, A1), …
  - Multi-level: (X, A1), (X, A2), …
  - 766,158 (direct & multilevel) hierarchical relations from Word-Net 3.0
- Handling the **Distance** Factor
  - The **closer** two senses are, the **larger impact** they should have on each other during training.

$$weight(s_1, s_2) = \max_{i,j} d(s_i, s_j) - d(s_1, s_2) + 1$$

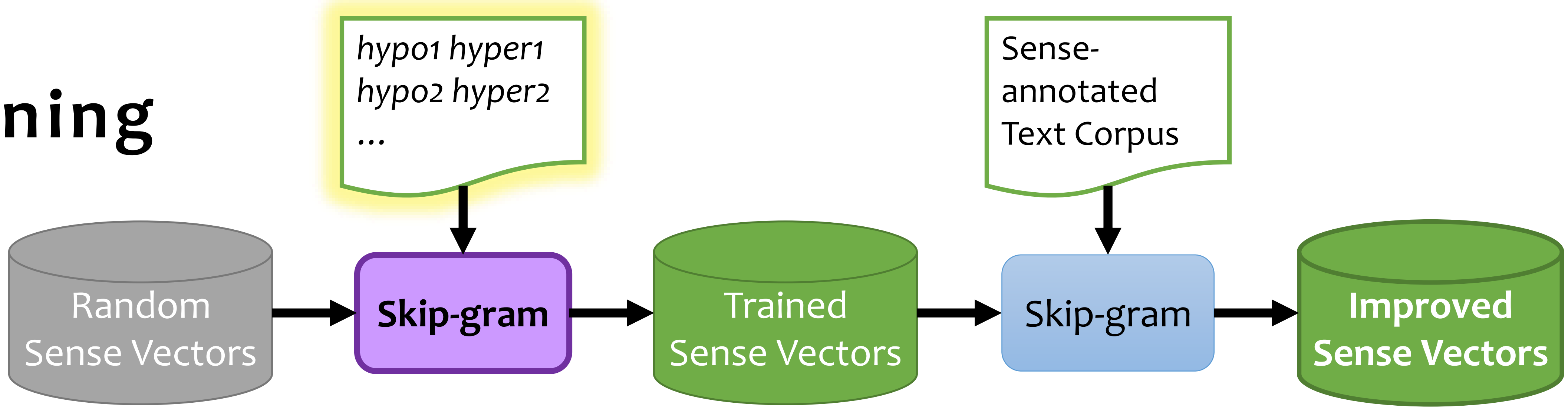
    - $d(s_i, s_j)$: distance / shortest path length of hyponym-hypernym senses s_i and s_j

II. Methods – **Sense** Vectors

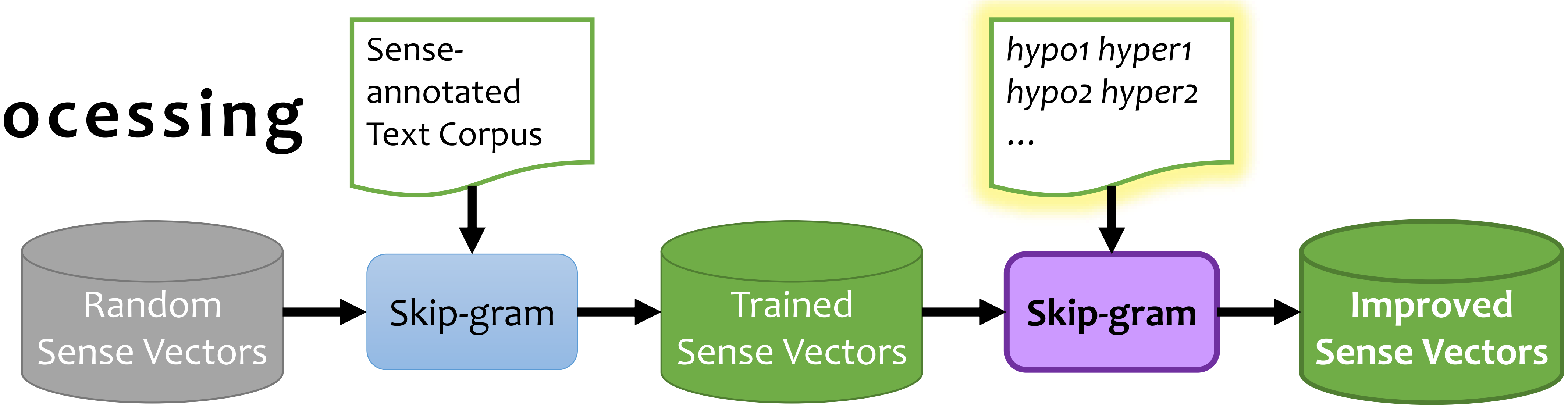
$$weight(s_1, s_2) = \max_{i,j} d(s_i, s_j) - d(s_1, s_2) + 1$$

- Incorporating weight factor
  - ***wn_cnt***: let relation pair (s_1, s_2) **occur** $weight(s_1, s_2)$ times in training file
  - ***wn_dis***: **multiply** the gradient by $weight(s_1, s_2)$ when using vector of s_1 to update that of s_2, or vice versa
- For comparison
  - ***wn_all***: no any weighting on relations
  - ***wn_dir***: only use direct relations
- Continue to train the pre-processed sense vectors with a sense-disambiguated corpus

Pre-training



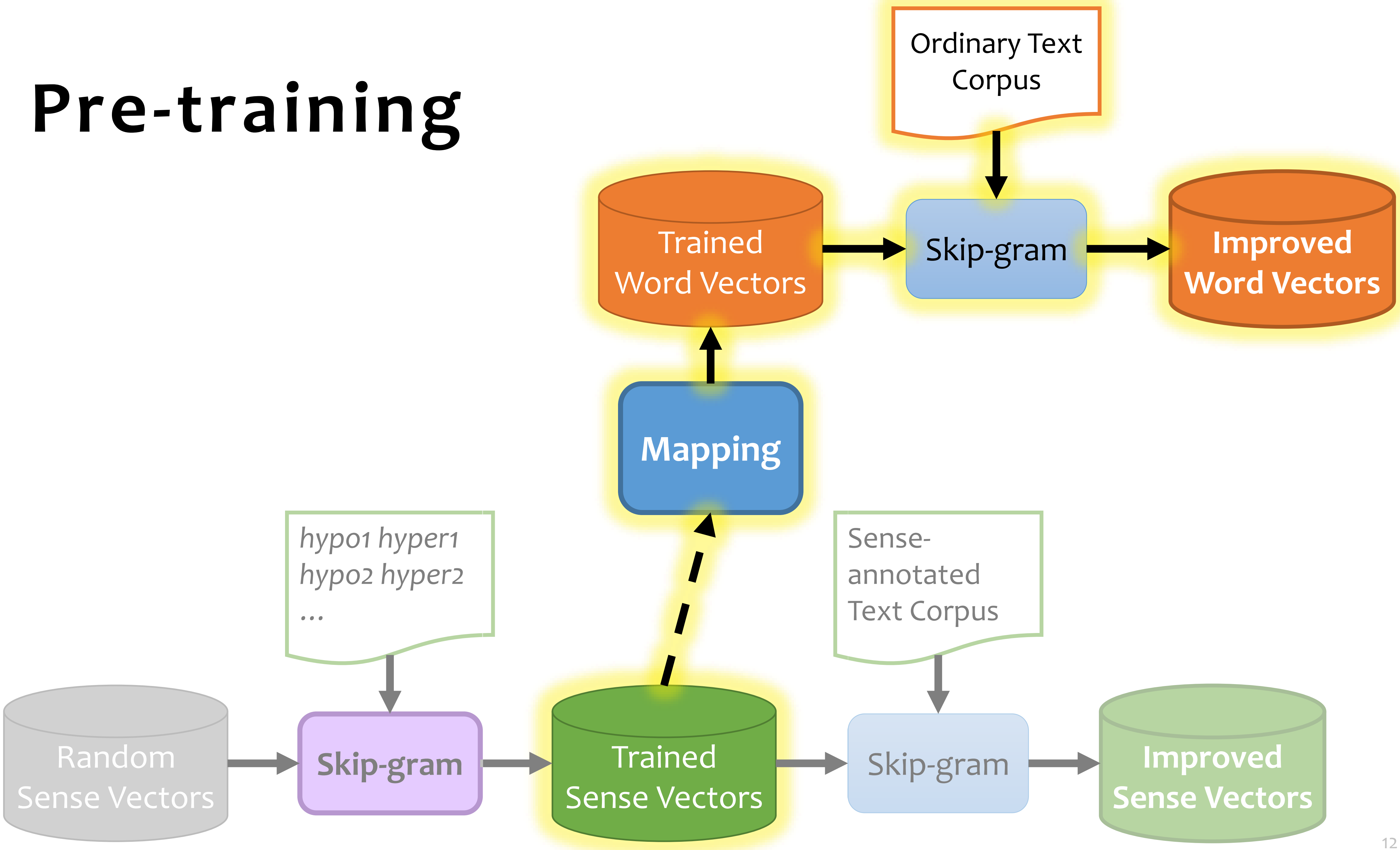
Post-processing



II. Methods – **Sense** Vectors

- Difference of characteristics between semantic relation data and corpus data
  - Relation: fairly **accurate** relationship, rather **sparse**
  - Corpus: very **dense**, describe **a variety of possible connections** among words
- Pre-training on the hierarchical relations
  - Build a framework of **core concepts** into the model
    → learn better in subsequent corpus training phase
- Post-processing: for comparison
- Save both target (input) & context (output) vectors

Pre-training



II. Methods – Word Vectors

- Despite benefits of moving from word to sense vectors, performing WSD may not be practical in all applications.
- **Mapping** pre-trained sense vectors to word vectors
  - ***First sense (FS)***: $vec(w) = vec(s_{w,1})$
    - $s_{w,1}$: first sense (predominant sense) of word w
  - ***Weighted senses (WS)***:

$$vec(w) = \frac{\sum_{i=1}^{n} freq(w, s_{w,i})\, vec(s_{w,i})}{\sum_{i=1}^{n} freq(w, s_{w,i})}$$

    - $freq(w, s_{w,i})$: # times word w is associated with sense $s_{w,i}$ in a disambiguated corpus
- Advantage: no need to expand a single **sense-level** relation (s_1, s_2) into $size(s_1) * size(s_2)$ **word-level** ones

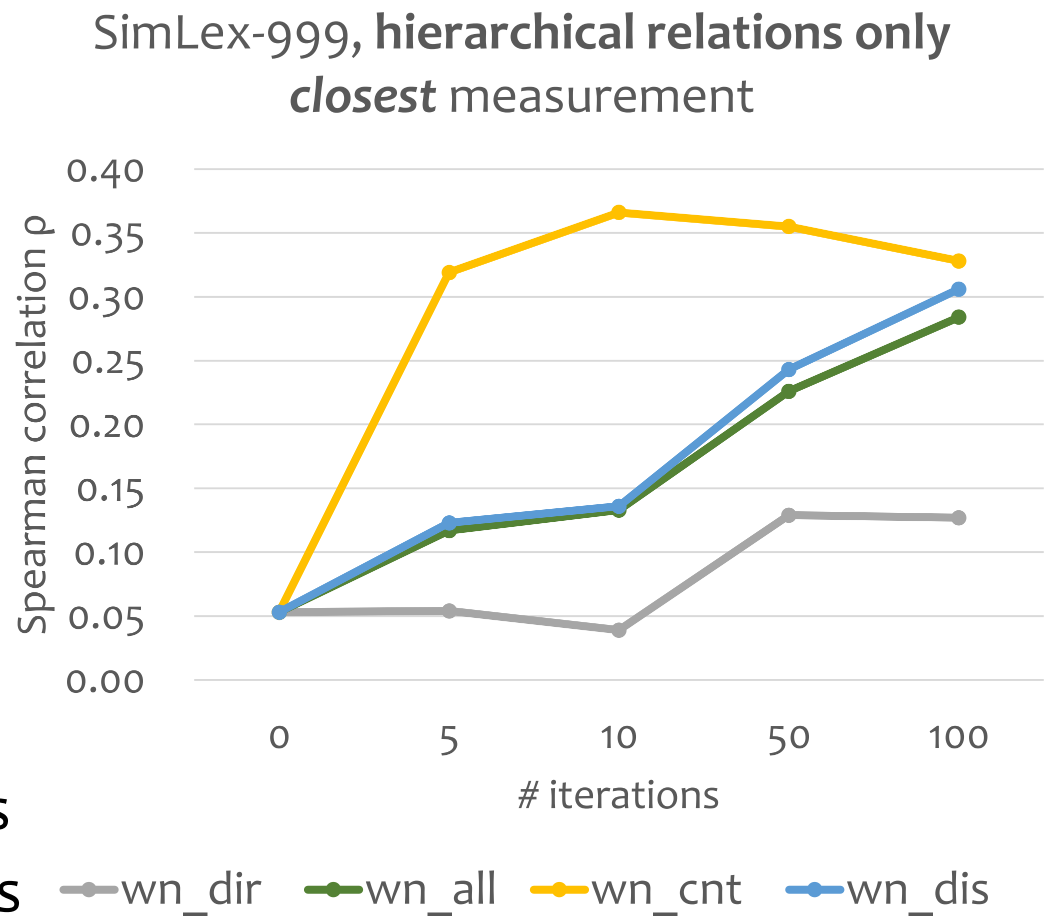
III. Experimental Results & Analysis

- **Intrinsic evaluation**: word similarity
- **Extrinsic evaluation**: sentiment analysis / dependency parsing

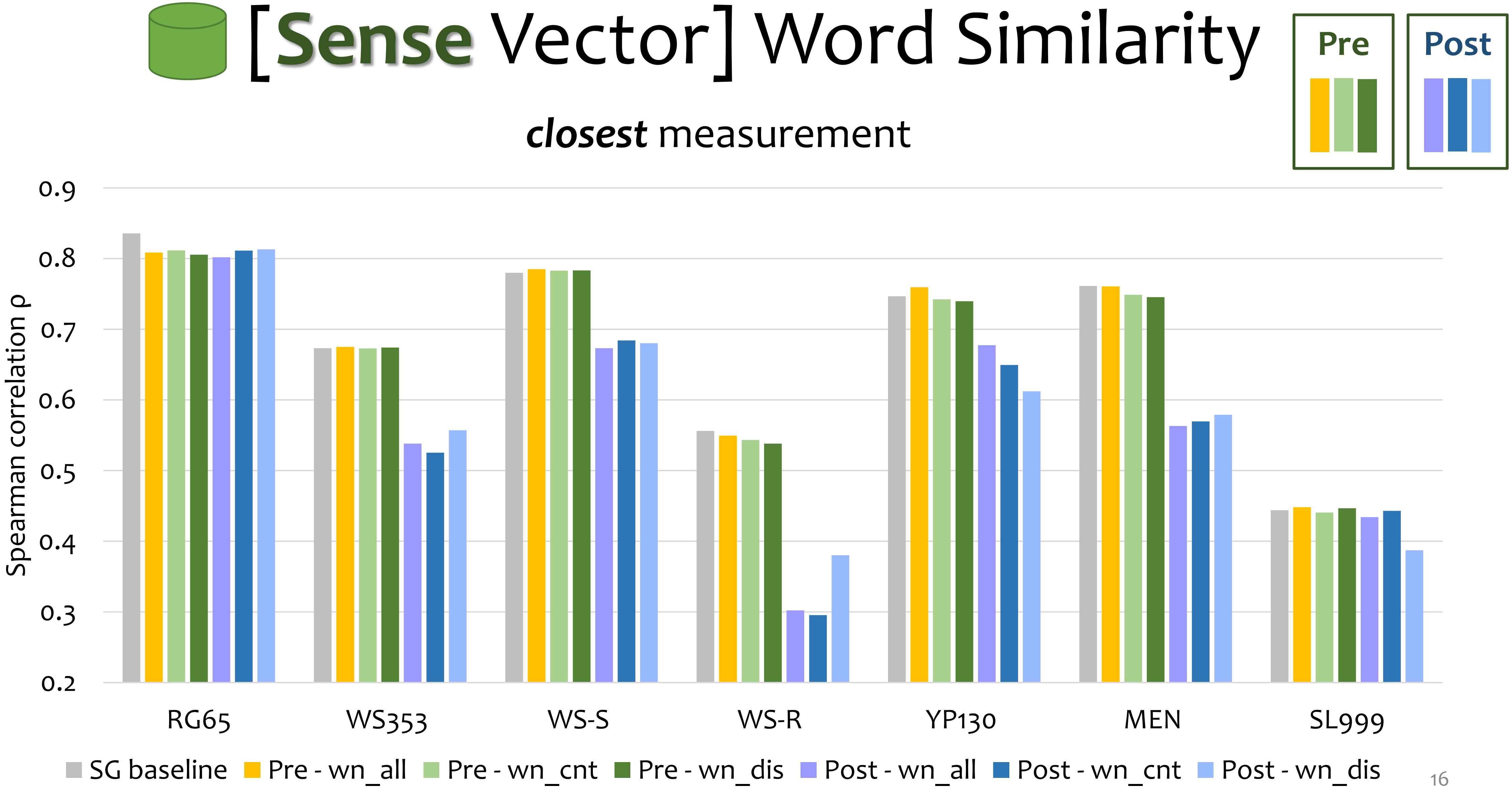
- Skip-gram (SG) parameters:
  dim.: 400 window size: 5 #(negative samples): 15
- Validation
  - Use SimLex-999 to validate for other datasets / WS-353 for SimLex-999
- Corpus: Dec-2015 dump of English Wikipedia
  - Sense-annotated corpus: obtained with Adapted Lesk

III. Experimental Results & Analysis
[**Sense** Vector] Word Similarity

- Use sense vectors to compute word similarities
  - ***closest***: similarity of the closest pair of senses
  - ***weighted***: sum of vectors of all possible senses weighted by frequency
- Rationale of using hierarchical relations
  - Increasing performance with more iterations → word similarity information can be learned from hierarchical relations
  - Multi-level relations > only direct relations

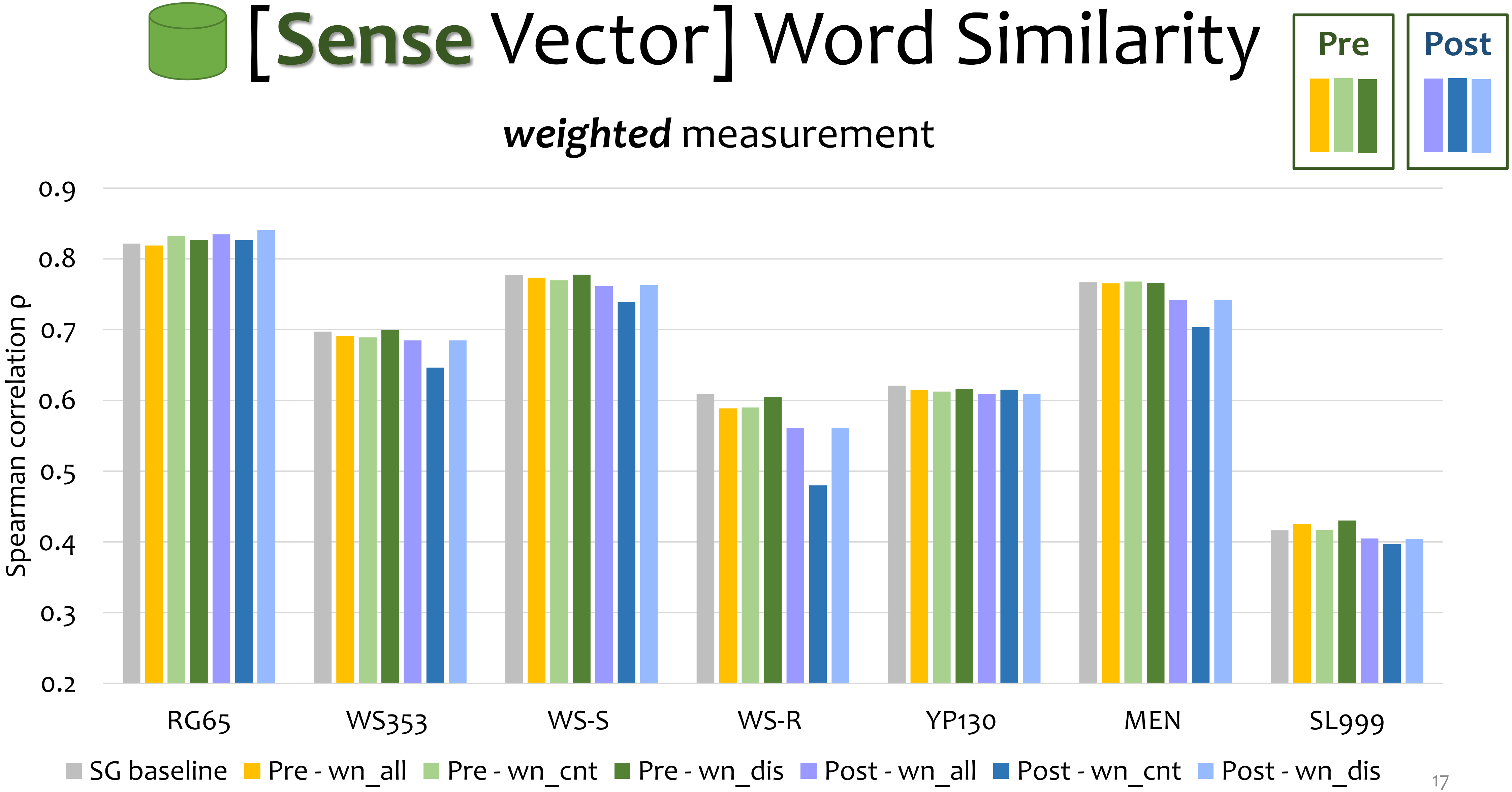


III. Experimental Results & Analysis
[Sense Vector] Word Similarity



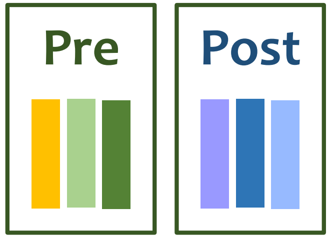
III. Experimental Results & Analysis

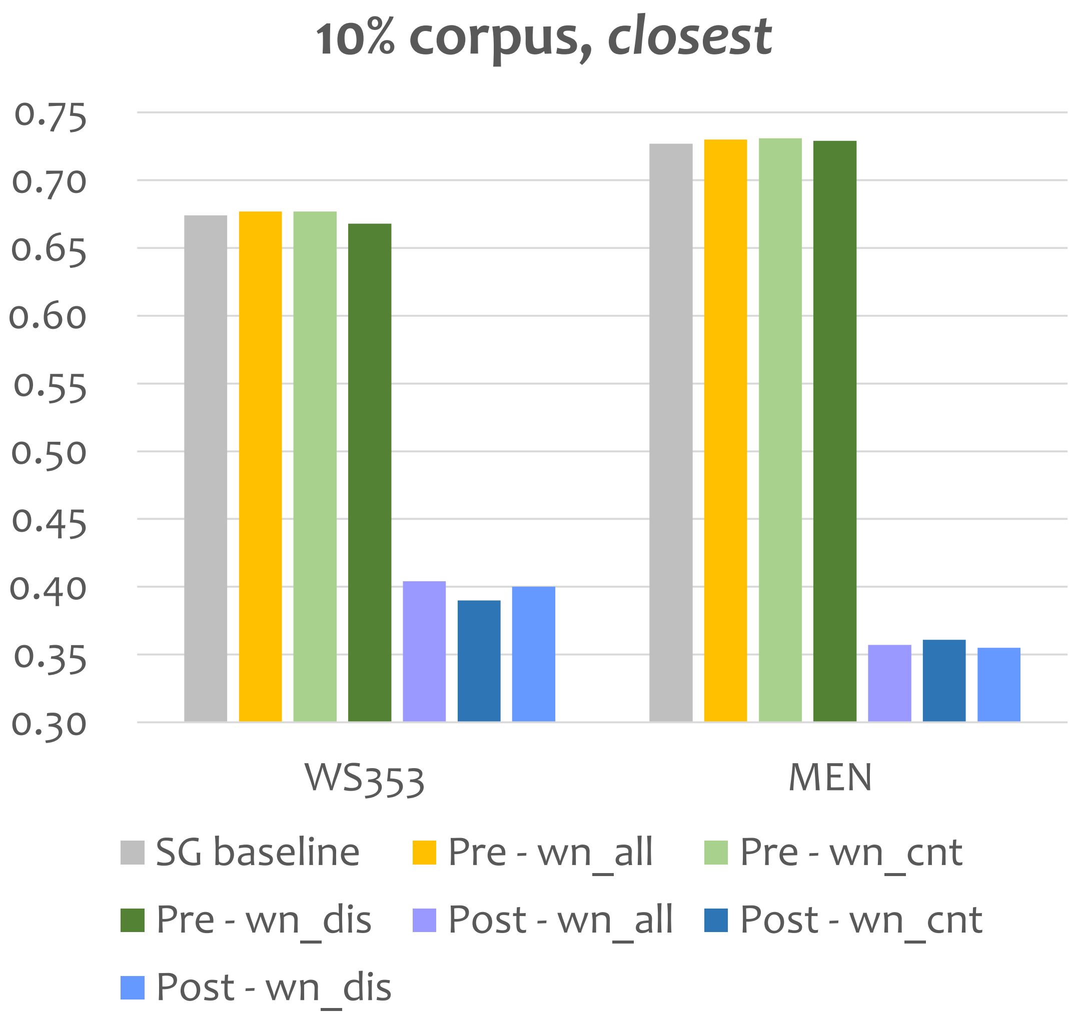
[Sense Vector] Word Similarity

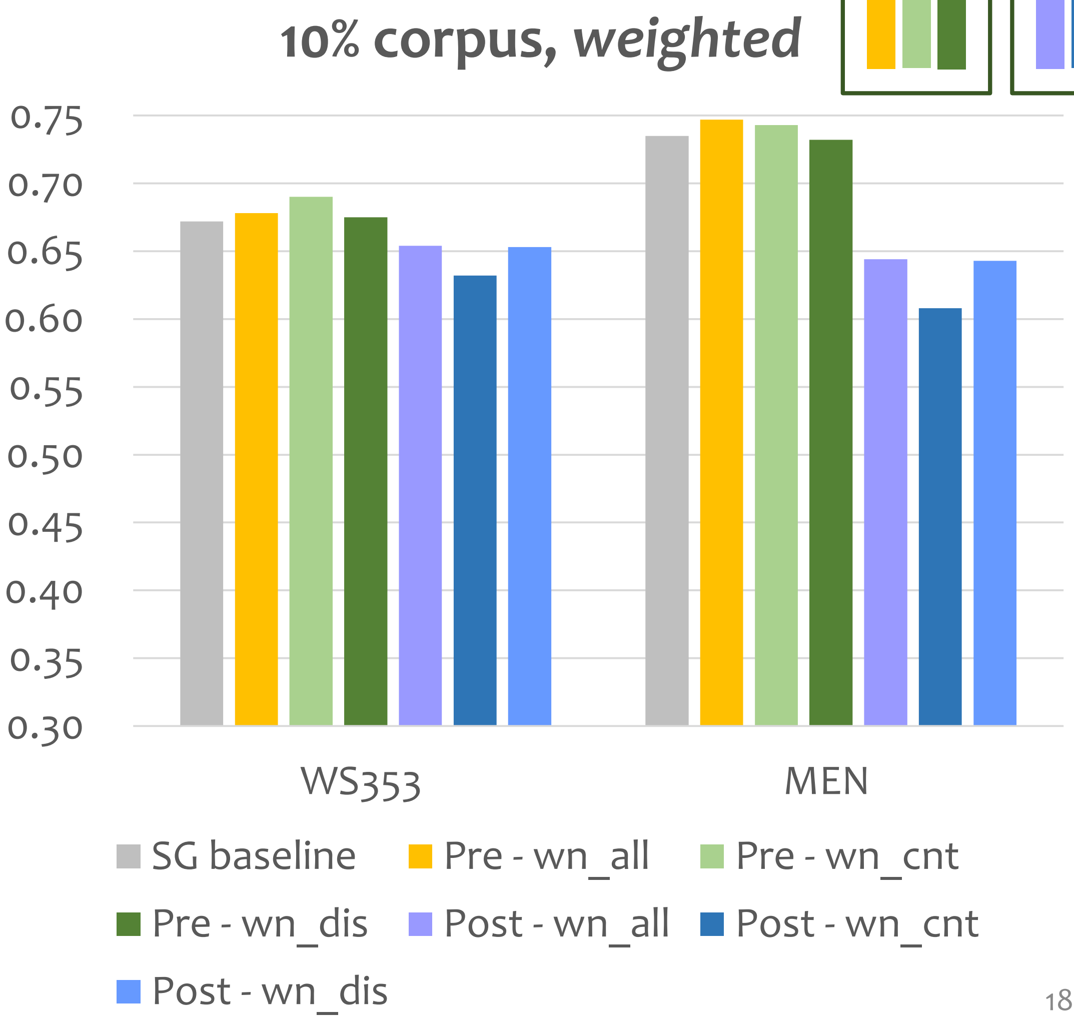


III. Experimental Results & Analysis

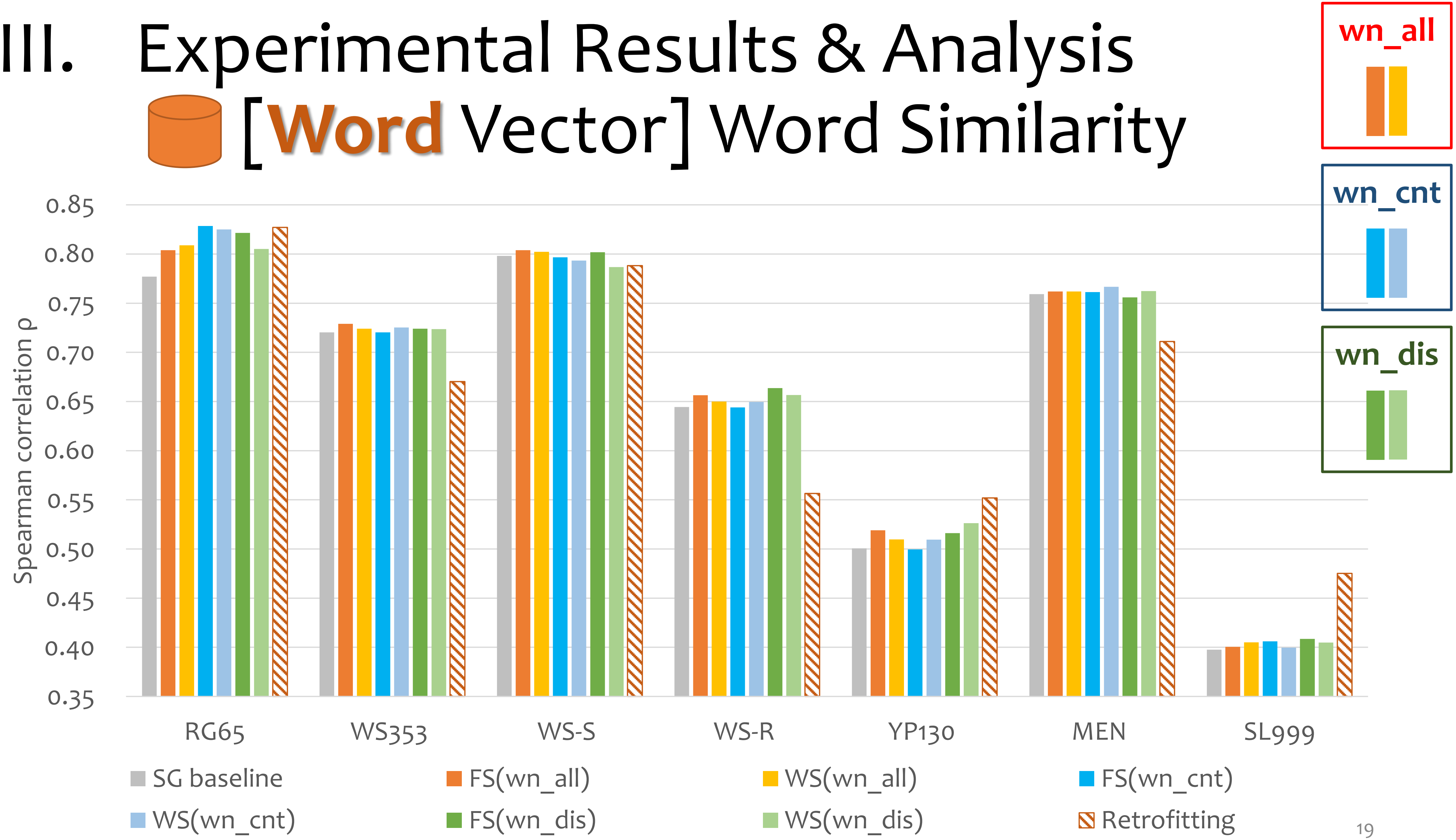
[Sense Vector] Word Similarity





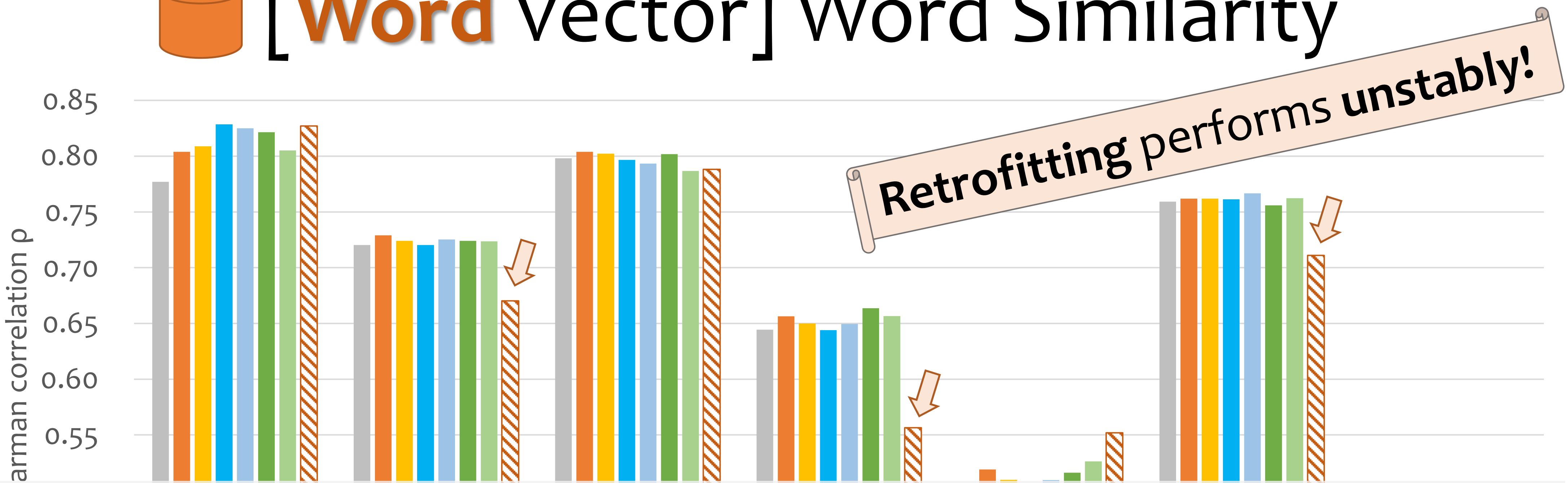


III. Experimental Results & Analysis
[Word Vector] Word Similarity



III. Experimental Results & Analysis

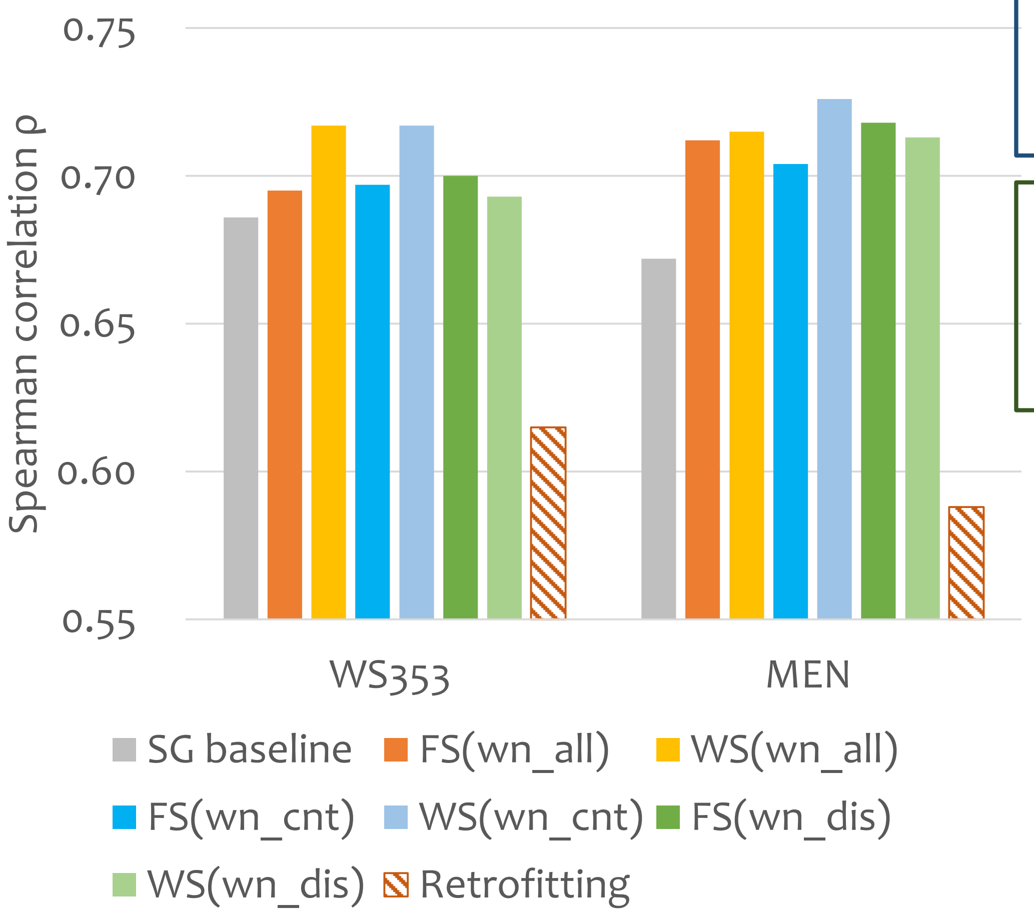
[**Word** Vector] Word Similarity

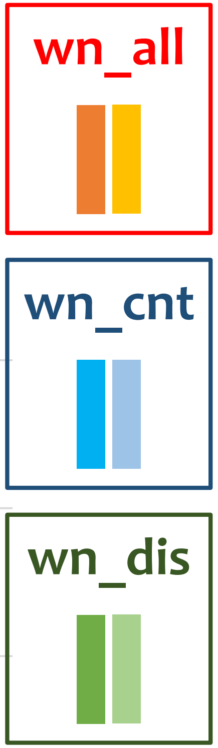


- **Closely-related** concepts could be **far apart** in WordNet
  - [WS-REL] sim(country, citizen) > sim(computer, news)
  - dis(country, citizen) = 8 , dis(computer, news) = 10
  - Baseline: X Retrofitting: X ***FS***(***wn_dis***): O
- **Multi-level > single-level**

III. Experimental Results & Analysis
[Word Vector] Word Similarity

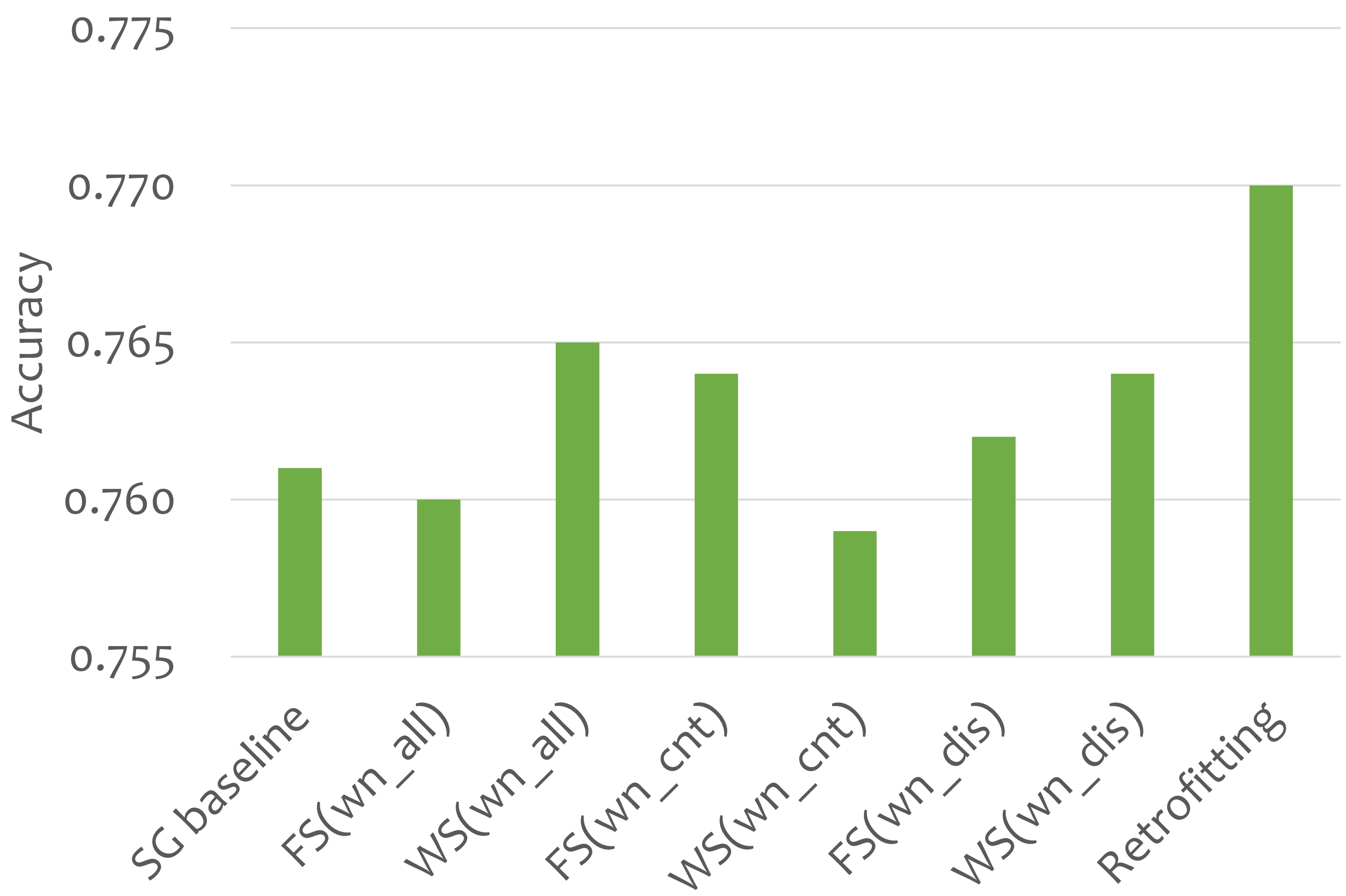
- 10% Corpus
  - More obvious improvement over baseline
  - Our method can be applied to **domain-specific** tasks
    e.g. biomedical domain
    - **Existing ontologies** of terms
    - **Less available text**
  - Retrofitting does not help when corpus is small, but **harms the performance a lot**





III. Experimental Results & Analysis
[Word Vector] Sentiment Analysis

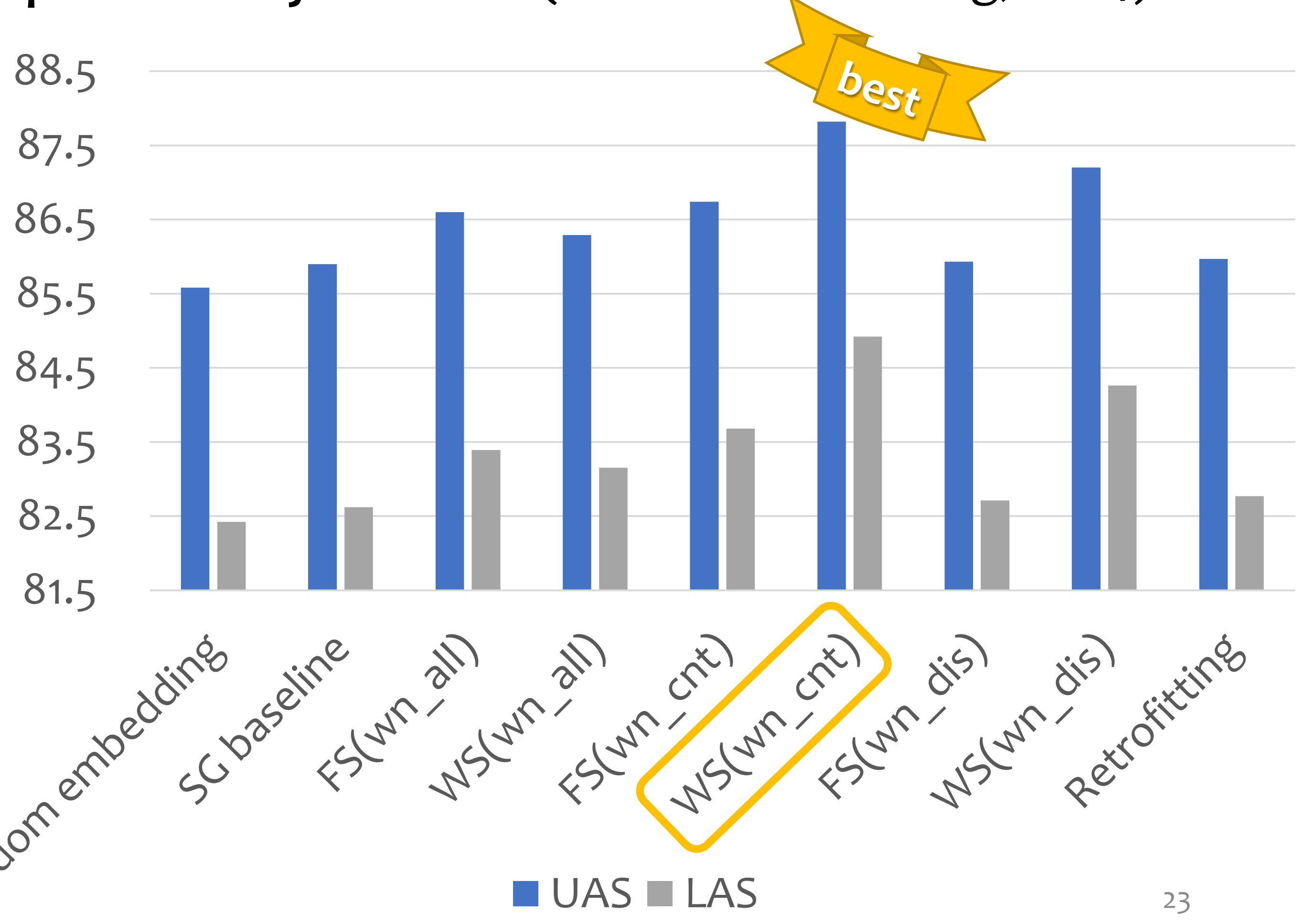
- Movie review dataset (Socher et al., 2013)
- Binary classification (positive/negative)
- Feature: sum of embedding of words
- Classifier: logistic regression



Slides: http://goo.gl/LWzZWZ

III. Experimental Results & Analysis
[Word Vector] Dependency Parsing

- Stanford Neural Network Dependency Parser (Chen and Manning, 2014)
- Evaluate on test set of PTB



- **Ontological knowledge** can serve as a clue for determining **whether there is a dependency** between two words
  - Retrofitting only includes limited knowledge

IV. Conclusion

- Simple but effective methods of utilizing hierarchical semantic relations to improve **sense & word vectors**
  - Model the **importance** of a relation according to its **distance**
  - Consistent improvement on intrinsic and extrinsic evaluations
    → directly applicable to existing **applications**
- Especially useful when **corpus is small**
  - **Pre-training** is more **reliable** than post-processing in such cases
- Future work: encode **directional** information
  - *cherry* is a kind of *tree* so it should **inherit** the properties of *tree*
  - But some properties of *cherry* **might not apply** to all kinds of *tree*

Slides: http://goo.gl/LWzZWZ

Thank you!

Yow-Ting Shiue orina1123@gmail.com

Wei-Yun Ma ma@iis.sinica.edu.tw

Slides: http://goo.gl/LWzZWZ