Explicit Retrofitting of Distributional Word Vectors

Goran Glavaš
Data & Web Science Group
University of Mannheim

Ivan Vulić
Language Technology Lab
University of Cambridge

ACL, Melbourne
July 16, 2018

Distributional hypothesis

„You shall know the meaning of the word by the company it keeps"

„Words that occur in similar contexts tend to have similar meanings"

Harris, 1954

Cars, Drivers, Vehicles, and Wheels

- Words co-occur in text due to
  - Paradigmatic relations (e.g., synonymy, hypernymy), but also due to
  - Syntagmatic relations (e.g., selectional preferences)

- Distributional vectors conflate all types of association
  - *driver* and *car* are not paradigmatically related
    - Not synonyms, not antonyms, not hypernyms, not co-hyponyms, etc.
  - But both words will co-occur frequently with
    - *driving*, *accident*, *wheel*, *vehicle*, *road*, *trip*, *race*, etc.

Vector specialization using external resources

- **Key idea**: refine vectors using external resources
- Specializing vectors for **semantic similarity**

1. **Joint specialization models**
   - Integrate external constraints into the learning objective
   - E.g., Yu & Dredze, '14; Kiela et al., '15; Osborne et al., '16; Nguyen et al., '17

2. **Retrofitting models**
   - Modify the pre-trained word embeddings using lexical constraints
   - E.g., Faruqui et al., '15; Wieting et al., '15; Mrkšić et al., '16; Mrkšić et al., '17

Vector specialization using external resources

- **Joint specialization models**
  - (+) Specialize the entire vocabulary (of the corpus)
  - (–) Tailored for a specific embedding model

- **Retrofitting models**
  - (–) Specialize only the vectors of words found in external constraints
  - (+) Applicable to any pre-trained embedding space
  - (+) Much better performance than joint models (Mrkšić et al., 2016)

This work

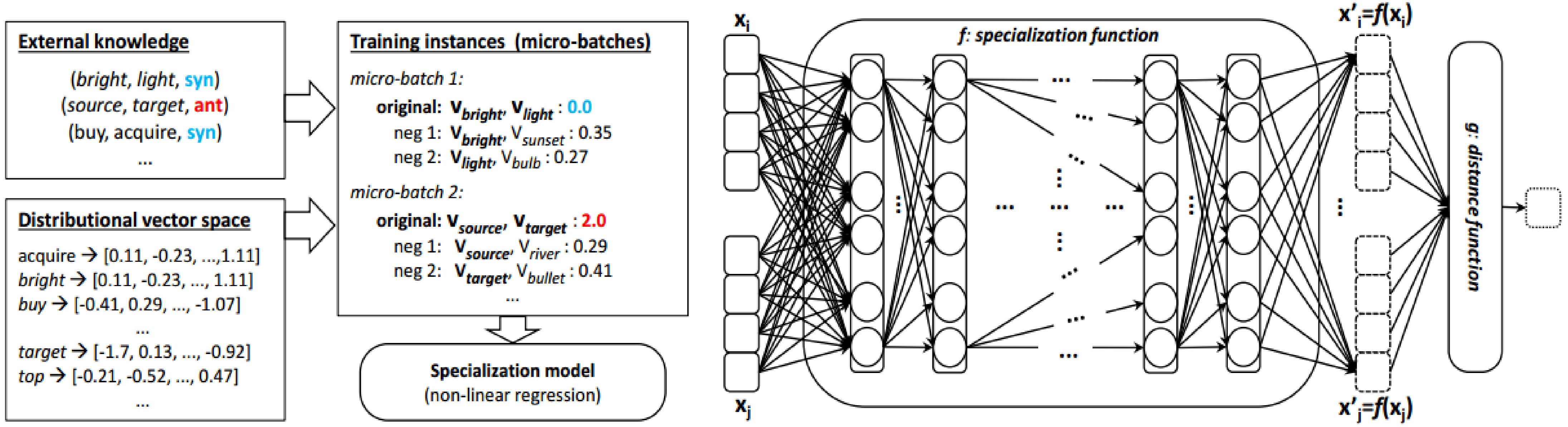
- **Best of both worlds**
  - Performance and flexibility of retrofitting models, while
  - Specializing entire embedding spaces (vectors of all words)

- **Simple idea**
  - Learn an explicit retrofitting/specialization function
  - Using external lexical constraints as training examples

Explicit Retrofitting Model

Explicit retrofitting

- Constraints (synonyms and antonyms) used as training examples for learning the **explicit specialization function**
  - Non-linear: **Deep Feed-Forward Network (DFFN)**



Constraints to training instances

- Specialization function: $\mathbf{x}' = f(\mathbf{x})$
- Distance function: $g(\mathbf{x}_1, \mathbf{x}_2)$
- Assumptions

1. (w_i, w_j, syn) – embeddings as close as possible after specialization

$$g(\mathbf{x}_i', \mathbf{x}_j') = g_{min}$$

2. (w_i, w_j, ant) – embeddings as far as possible after specialization

$$g(\mathbf{x}_i', \mathbf{x}_j') = g_{max}$$

3. (w_i, w_j) – the non-costraint words stay at the same distance

$$g(\mathbf{x}_i', \mathbf{x}_j') = g(\mathbf{x}_i, \mathbf{x}_j)$$

Constraints to training instances

- **Micro-batches** – each constraint (w_i, w_j, r) paired with
  - K pairs $\{(w_i, w_m{}^k)\}_k$ – $w_m{}^k$ most similar to w_i in distributional space
  - K pairs $\{(w_j, w_n{}^k)\}_k$ – $w_n{}^k$ most similar to w_j in distributional space
  - Total: 2K+1 word pairs

$$
\begin{aligned}
M(w_i, w_j, r) = \{(\mathbf{x}_i, \mathbf{x}_j, g_r)\} \;\cup \\
\{(\mathbf{x}_i, \mathbf{x}_m^k, g(\mathbf{x}_i, \mathbf{x}_m^k))\}_{k=1}^{K} \;\cup \\
\{(\mathbf{x}_j, \mathbf{x}_n^k, g(\mathbf{x}_j, \mathbf{x}_n^k))\}_{k=1}^{K}
\end{aligned}
$$

Loss function

- **Contrastive Objective (CNT)**

$$J_{CNT} = \sum_{M_s \in S} \sum_{i=2}^{2K+1} \Big(\underbrace{(g^i - \overset{=0}{g_{min}})}_{\text{„Gold” diff.}} - \underbrace{(g'^i - g'^1)}_{\text{Predicted diff.}} \Big)^2$$

$$+ \sum_{M_a \in A} \sum_{i=2}^{2K+1} \Big(\overbrace{(\underset{=2}{g_{max}} - g^i)}^{\text{„Gold” diff.}} - \overbrace{(g'^1 - g'^i)}^{\text{Predicted diff.}} \Big)^2$$

- **Regularization**

$$J_{REG} = \sum_{i=1}^{N} g(\mathbf{x}_1^i, f(\mathbf{x}_1^i)) + g(\mathbf{x}_2^i, f(\mathbf{x}_2^i))$$

Evaluation

Model Configuration

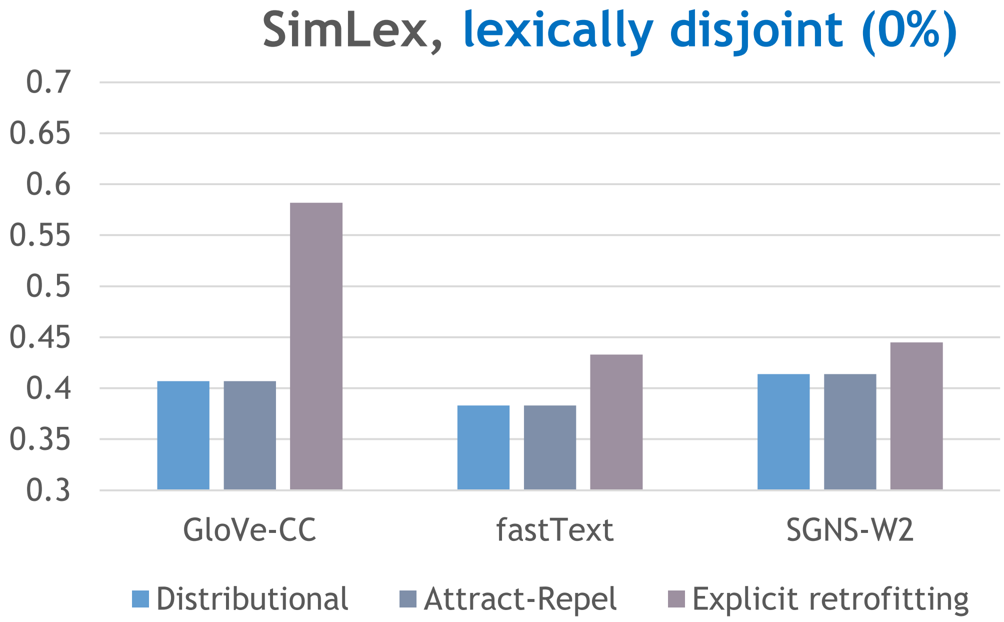
- Distance function g: **cosine distance**
- DFFN activation function: **hyperbolic tangent**

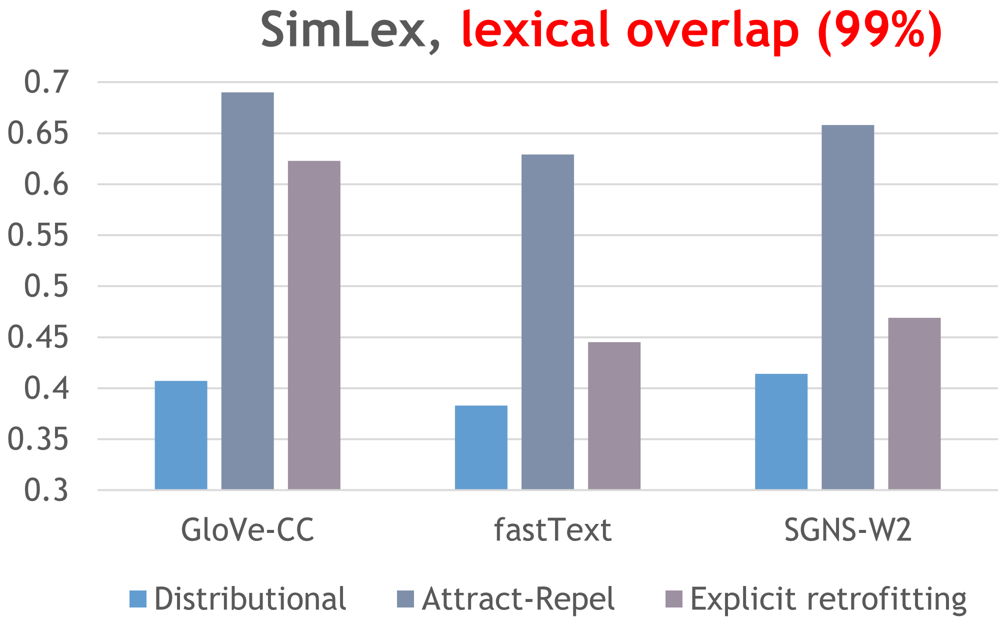
- Constraints from previous work (Zhang et al, ’14; Ono et al., ‘15)
  - 1M synonymy constraints
  - 380K antonymy constraints
  - But **only 57K unique words** in these constraints!

- **10%** of **micro-batches** used for model validation
  - H (hidden layers) = 5, d_h (layer size) = 1000, $\lambda = 0.3$
  - K = 4 (micro-batch size = 9), batches of 100 micro-batches
  - ADAM optimization (Kingma & Ba, 2015)

Intrinsic Evaluation

- SimLex-999 (Hill et al., 2014), SimVerb-3500 (Gerz et al., 2016)
- Important aspect: percentage of test words covered by constraints
- Comparison with Attract-Repel (Mrkšić et al., 2017)





Intrinsic Evaluation

- Intrinsic evaluation depicts two extreme settings
- **Lexical overlap** setting
  - Synonymy and antonymy constraints contain **99%** of SL and SV words
  - Performance is an optimistic estimate or true performance
- **Lexically disjoint** setting
  - Constraints contain **0%** of SL and SV words
  - Performance is a pessimistic estimate of true performance

- Realistic setting: **downstream tasks**
  - Coverage of test set words by constraints between **0%** and **100%**

Donwstream tasks: DST & LS

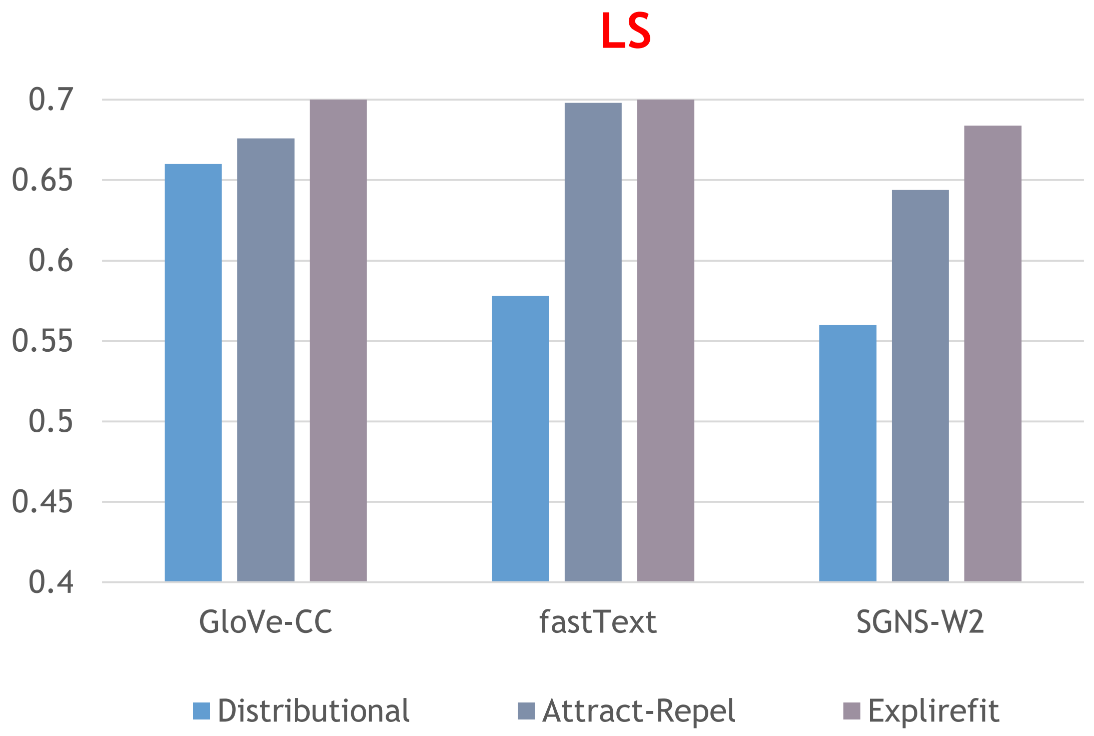
- **Dialog state tracking** (DST) – first component of a dialog system
  - Neural Belief Tracker (NBT) (Mrkšić et al., '17)
  - Makes inferences purely based on an embedding space
  - **57%** of words in NBT test set (Wen et al., '17) covered by specialization constraints

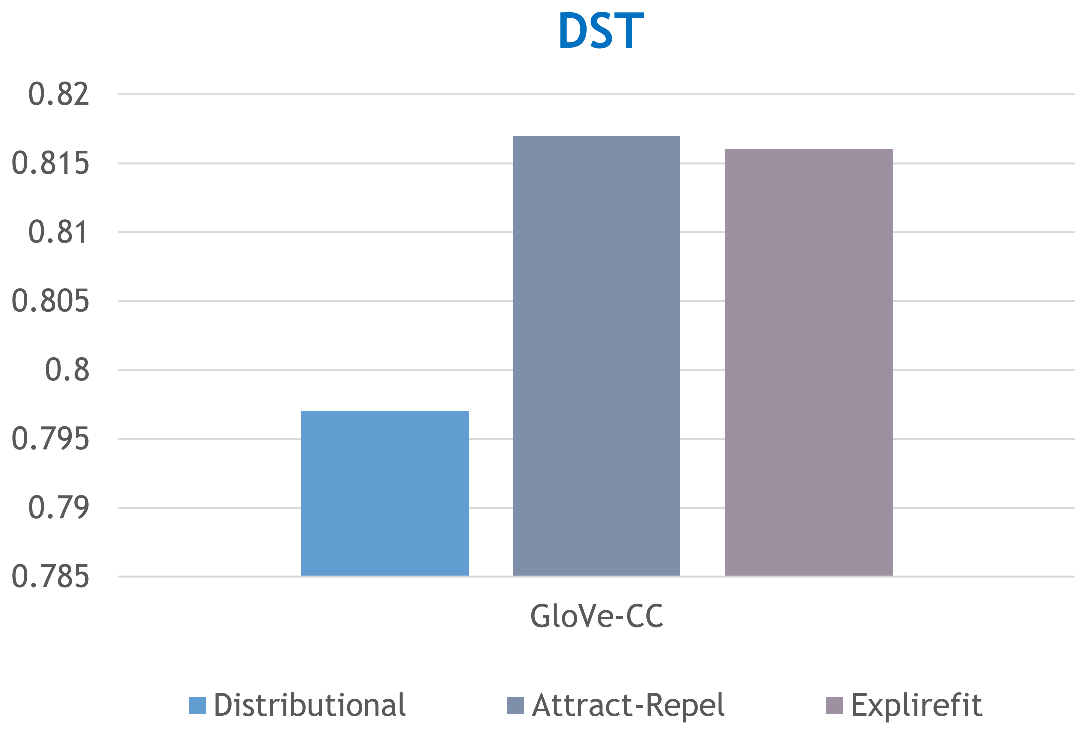
- **Lexical simplification** (LS) – complex words to simpler synonyms
  - Light-LS (Glavaš & Štajner, '15) – decisions purely based on an embedding space
  - **59%** of LS dataset words (Horn et al., 14) found in specialization constraints

- **Crucial to distinguish similarity from relatedness**
  - DST: „cheap pub in the east" vs. „expensive restaurant in the west"
  - LS: „Ferrari's **pilot** Sebastian Vettel won the race.", "driver" vs. "airplane"

Downstream tasks – Evaluation

- Lexical simplification (LS) and Dialog state tracking (DST)





Cross-lingual specialization transfer

Language transfer

- Lexico-semantic resources such as WordNet needed to collect synonymy and antonymy constraints
- **Idea:** use **shared bilingual embedding spaces** to transfer the specialization to another language

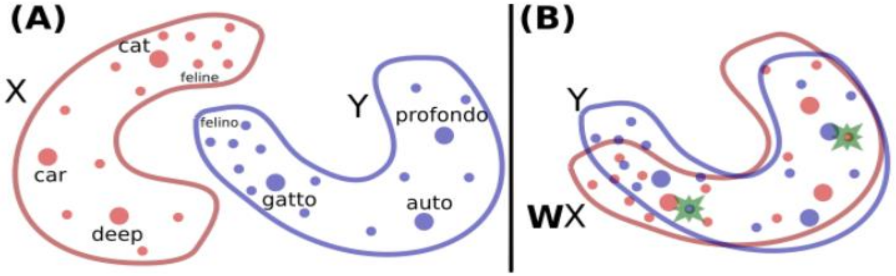
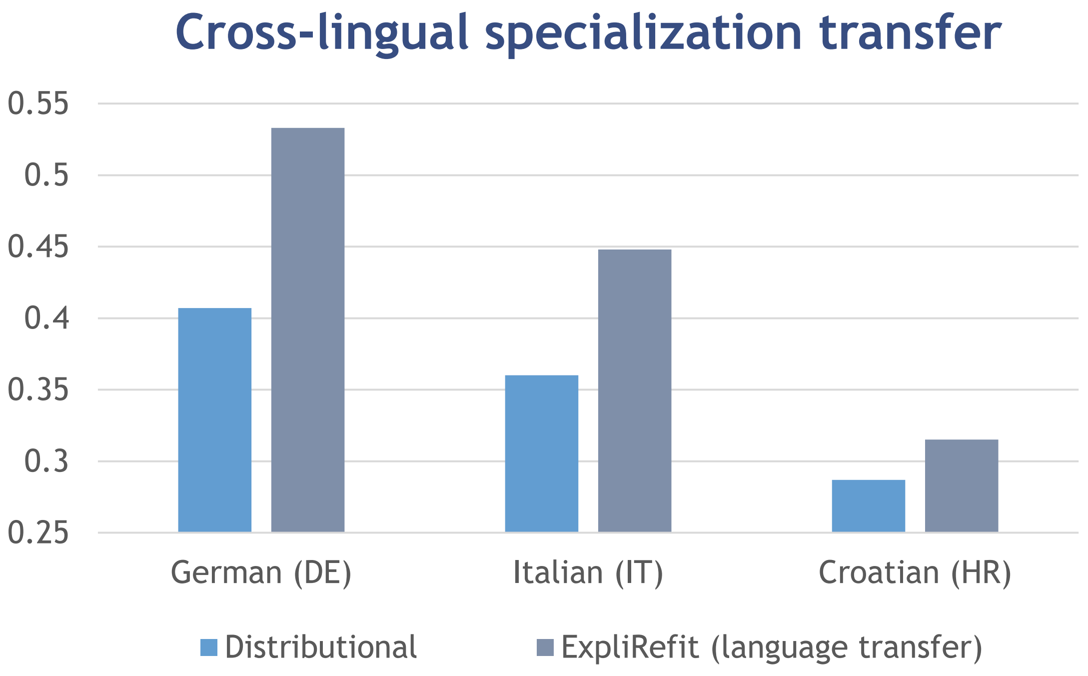


*Image taken from Lample et al., ICLR 2018

- Most models learn a (simple) linear mapping
  - Using word alignments (Mikolov et al., 2013; **Smith et al., 2017**)
  - Without word alignments (Lample et al., 2018; Artetxe et al., 2018)

Cross-lingual transfer – results

- Transfer to three languages: DE, IT, and HR
  - Different levels of proximity to English
  - Variants of SimLex-999 exist for each of these three languages



Conclusion

- **Retrofitting models** specialize (i.e., fine-tune) distributional vectors for semantic similarity
  - Shortcoming: specialize **only** vectors of words seen in external constraints
- **Explicit retrofitting**
  - Learning the specialization function using constrains as training examples
  - Able to specialize distributional vectors of all words
  - Good intrinsic (SL, SV) and downstream (DST, LS) performance
- **Cross-lingual specialization transfer** possible for languages without lexico-semantic resources

Thank you for attention!

- **Code & data**
  - https://github.com/codogogo/explirefit
- **Contact**
  - goran@informatik.uni-mannheim.de
  - iv250@hermes.cam.ac.uk