Incorporating Both Distributional and Relational Semantics in Word Representations

Daniel Fried[1] Kevin Duh[2]

[1]University of Arizona [2]Nara Institute of Science & Technology

dfried@cs.arizona.edu kevinduh@is.naist.jp

Integrating Representation Methods

- **Distributional semantics** defines a word by its context. Similar context distributions $\implies$ similar meaning:

  I closed the **door**.
  I closed the **window**.

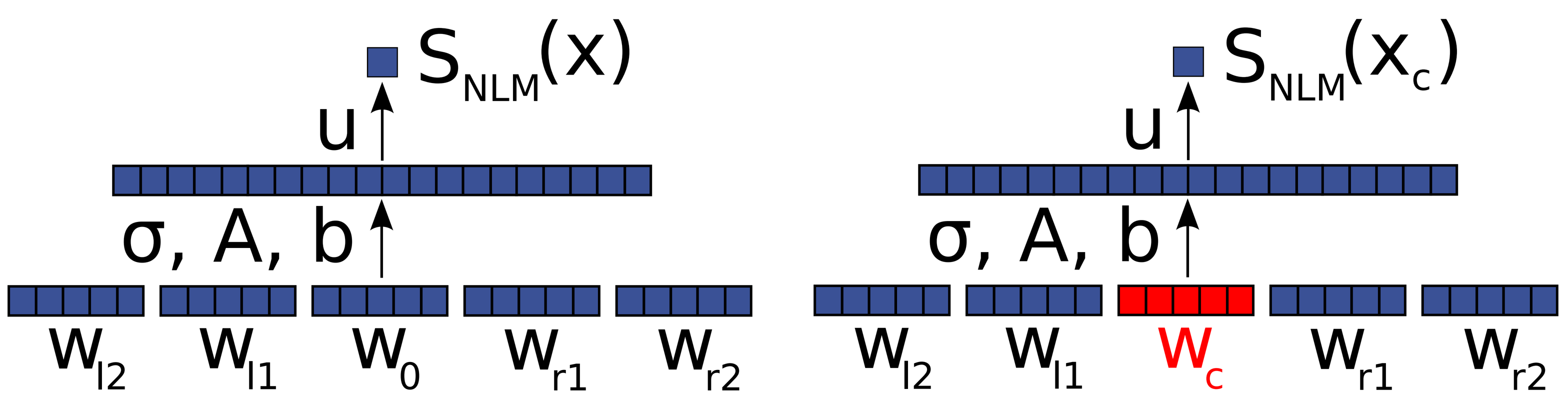
- **Relational semantics** defines a word by its relationship to other words. Encoded in knowledge bases like WordNet:

  door IS-PART-OF wall
  door HAS-A lock

- Can we learn more effective vector representations for words by modeling both types of semantics?

Distributional Objective

- **Neural Language Model** adapted from Collobert et al. (2011)



- Score a window of 5 words, $w_{l2}, w_{l1}, w_0, w_{r1}, w_{r2}$, by concatenating the word vectors, $\mathbf{x} = [\mathbf{w}_{l2}\ \mathbf{w}_{l1}\ \mathbf{w}_0\ \mathbf{w}_{r1}\ \mathbf{w}_{r2}]$, and passing through a NN with one hidden layer:

$$S_{NLM}(\mathbf{x}) = \mathbf{u}^\top(\sigma(\mathbf{Ax} + \mathbf{b}))$$

- Generate negative examples by corrupting the middle word, randomly: $\mathbf{x}_c = [\mathbf{w}_{l2}\ \mathbf{w}_{l1}\ \mathbf{w}_c\ \mathbf{w}_{r1}\ \mathbf{w}_{r2}]$. Train with hinge loss:

$$L_{NLM}(\mathbf{x}, \mathbf{x}_c) = \max(0, 1 - S_{NLM}(\mathbf{x}) + S_{NLM}(\mathbf{x}_c))$$

Relational Objectives

- **Graph Distance (GD)**: vector cosine similarity should be a function of words' normalized graph distance in WordNet:

$$L_{GD}(\mathbf{w}) = \sum_{i,j} \left(\frac{\mathbf{w}_i \cdot \mathbf{w}_j}{||\mathbf{w}_i||_2 ||\mathbf{w}_j||_2} - [a \times WordSim(i,j) + b] \right)^2$$

- Two existing approaches that model WordNet relationships (w_l:**door**, R:**IS-PART-OF**, w_j:**wall**) as vector operations (trained using negative sampling and hinge loss, as in L_{NLM})
  - **TransE** (Bordes et al., 2013): Translation-vector $\mathbf{t}_R$ for R:

  $$S_{TransE}(w_i, R, w_j) = -||\mathbf{w}_i + \mathbf{t}_R - \mathbf{w}_j||_2$$

  - **Neural Tensor Network** (NTN), Socher et al. (2013): NN with a tensor-based input layer ($\mathcal{T}_R, \mathbf{V}_R, \mathbf{b}_R$) for R:

  $$S_{NTN}(w_l, R, w_r) = \mathbf{v}^\top \sigma\left(\mathbf{w}_l^\top \mathcal{T}_R \mathbf{w}_r + \mathbf{V}_R[\mathbf{w}_l\ \mathbf{w}_r] + \mathbf{b}_R\right)$$

Joint Objective Optimization

- Want to learn embeddings $\mathbf{w}$ and parameters θ, ϕ that minimize the joint distributional and relational loss:

$$\arg\min_{\mathbf{w}} L_{dist}(\mathbf{w}, \theta) + L_{rel}(\mathbf{w}, \phi)$$

- **Alternating Direction Method of Multipliers** (ADMM): Keep separate word embeddings for each objective ($\mathbf{w}$ and $\mathbf{v}$), constrained by a Lagrangian penalty ($\mathbf{y}$ is a penalty vector for each embedding vector, ρ is a hyperparam):

$$\arg\min_{\mathbf{w},\mathbf{v}} L_{dist}(\mathbf{w}, \theta) + L_{rel}(\mathbf{v}, \phi) + L_P(\mathbf{w}, \mathbf{v})$$

$$L_P(\mathbf{w}, \mathbf{v}) = \sum_i \left(\mathbf{y}_i^\top(\mathbf{w}_i - \mathbf{v}_i)\right) + \frac{\rho}{2}\left(\sum_{i \in I} (\mathbf{w}_i - \mathbf{v}_i)^\top(\mathbf{w}_i - \mathbf{v}_i)\right)$$

- Modular, separable optimization:
  1. $\mathbf{w} := \arg\min_{\mathbf{w}} L_{dist}(\mathbf{w}, \theta) + L_P(\mathbf{w}, \mathbf{v})$
  2. $\mathbf{v} := \arg\min_{\mathbf{v}} L_{rel}(\mathbf{v}, \phi) + L_P(\mathbf{w}, \mathbf{v})$
  3. $\mathbf{y}_i := y_i + \rho(\mathbf{w}_i - \mathbf{v}_i)$ **and loop**

Experiments

Training

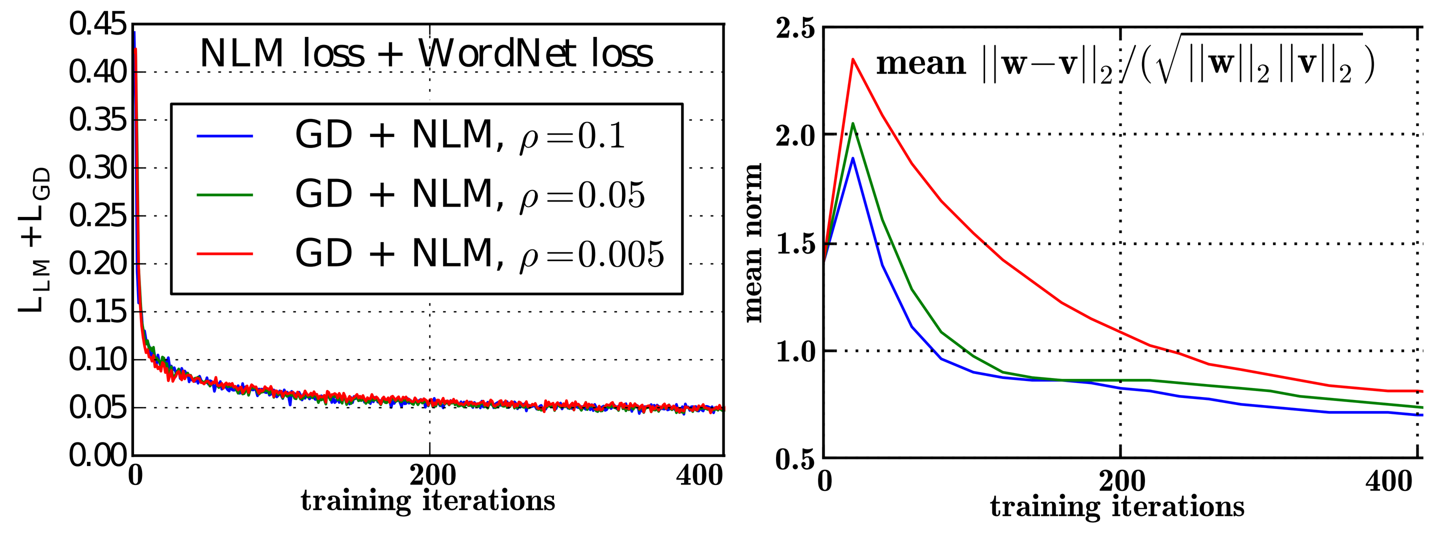
- Distributional: Google Books corpus of 180M 5-grams
- Relational: train using WordNet IS-A graph for GD and Socher et al.'s WordNet relations for NTN & TransE
- Stochastic gradient descent over n-grams and relations

Evaluation Tasks

- Knowledge base completion: determine whether a relationship $\mathbf{w}_i, R, \mathbf{w}_j$ is part of WordNet (Socher et al.)
- Analogy: score word pairs by degree they match an example relationship: *attack* is to *defend* as *buy* is to *sell*
- Dependency parsing: use clustered embeddings in a dependency parser on an out-of-domain parsing task

Results

- Despite non-convex loss functions, ADMM converges:



Joint loss (left) and embedding residual differences (right)

- Plain NLM is best on the analogy task. Small increase on KB task when adding NLM to NTN & TransE. Best parsing performance comes from GD+NLM joint objective:

	NLM	GD	GD+NLM	TransE	TransE+NLM	NTN	NTN+NLM
Analogy	**42**	41	41	37	38	36	41
KB	-	-	-	82.9	**83.1**	81.0	81.2
Parsing	76.0	75.9	**76.2**	75.9	76.0	75.9	76.1

This work is supported by a Microsoft Research CORE Grant and JSPS KAKENHI Grant Number 26730121. D.F. was supported by the Flinn Scholarship during the course of this work.