Incorporating Relational Knowledge into Word Representations using Subspace Regularization

Jun Araki (Carnegie Mellon University)

joint work with Abhishek Kumar (IBM Research)

ACL 2016

Distributed word representations

- Low-dimensional dense word vectors learned from unstructured text
  - Based on distributional hypothesis (Harris, 1954)
  - Capture semantic and syntactic regularities of words, encoding word relations
    - e.g., $v(king) - v(man) + v(woman) \approx v(queen)$
  - Publicly available, well-developed software: word2vec and GloVe
  - Successfully applied to various NLP tasks

Underlying motivation

- Two variants of the word2vec algorithm by Mikolov et al. (2013)
  - Skip-gram maximizes

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_{t-c}^{t+c}|w_t) = \frac{1}{T}\sum_{t=1}^{T}\log\frac{\exp(\mathbf{v}_t'^{\top}\mathbf{w}_t)}{\sum_{w\in V}\exp(\mathbf{v}'^{\top}\mathbf{w}_t)}$$

  - Continuous bag-of-words (CBOW) maximizes

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-c}^{t+c}) = \frac{1}{T}\sum_{t=1}^{T}\log\frac{\exp(\mathbf{w}_t'^{\top}\mathbf{v}_t)}{\sum_{w\in V}\exp(\mathbf{w}'^{\top}\mathbf{v}_t)}$$

Underlying motivation

- Two variants of the word2vec algorithm by Mikolov et al. (2013)
  - Skip-gram maximizes

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_{t-c}^{t+c}|w_t) = \frac{1}{T}\sum_{t=1}^{T}\log\frac{\exp(\mathbf{v}_t'^{\top}\mathbf{w}_t)}{\sum_{w\in V}\exp(\mathbf{v}'^{\top}\mathbf{w}_t)}$$

  - Continuous bag-of-words (CBOW) maximizes

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-c}^{t+c}) = \frac{1}{T}\sum_{t=1}^{T}\log\frac{\exp(\mathbf{w}_t'^{\top}\mathbf{v}_t)}{\sum_{w\in V}\exp(\mathbf{w}'^{\top}\mathbf{v}_t)}$$

- They rely on co-occurrence statistics only

- **Motivation**: combining word representation learning with lexical knowledge

Prior work (1): Grouping similar words

- Lexical knowledge: {(w_i, r, w_j)}
  - Words w_i and w_j are connected by relation type r

Prior work (1): Grouping similar words

- Lexical knowledge: {(w_i, r, w_j)}
  - Words w_i and w_j are connected by relation type r
- Treats w_i and w_j as generic similar words
  - (Yu and Dredze, 2014; Faruqui et al., 2015; Liu et al., 2015)
  - Regularization effect: $\mathbf{w}_i \approx \mathbf{w}_j \ \forall(w_i, r, w_j)$
  - Based on a (over-)generalized notion of word similarity
  - Ignores relation types

Prior work (1): Grouping similar words

- Lexical knowledge: {(w_i, r, w_j)}
  - Words w_i and w_j are connected by relation type r
- Treats w_i and w_j as generic similar words
  - (Yu and Dredze, 2014; Faruqui et al., 2015; Liu et al., 2015)
  - Regularization effect: $\mathbf{w}_i \approx \mathbf{w}_j \ \forall(w_i, r, w_j)$
  - Based on a (over-)generalized notion of word similarity
  - Ignores relation types

- Limitations
  - Places an implicit restriction on relation types
    - E.g., synonyms and paraphrases

Prior work (2): Constant translation model

- CTM models each relation type *r* by a **relation vector *r***
  - (Bordes et al., 2013; Xu et al., 2014; Fried and Duh, 2014)
  - Regularization effect: $\mathbf{w}_i + \mathbf{r} \approx \mathbf{w}_j \ \forall(w_i, r, w_j)$
  - Assumes that w_i can be translated into w_j by a simple sum with a single relation vector

Prior work (2): Constant translation model

- CTM models each relation type *r* by a **relation vector *r***
  - (Bordes et al., 2013; Xu et al., 2014; Fried and Duh, 2014)
  - Regularization effect: $\mathbf{w}_i + \mathbf{r} \approx \mathbf{w}_j \ \forall(w_i, r, w_j)$
  - Assumes that w_i can be translated into w_j by a simple sum with a single relation vector

- Limitations
  - The assumption can be very restrictive when word representations are learned from co-occurrence instances
  - Not suitable for modeling:
    - symmetric relations (e.g., antonymy)
    - transitive relations (e.g., hypernymy)

Subspace-regularized word embeddings

- We model each relation type by a **low-rank subspace**
  - This relaxes the constant translation assumption
  - Suitable for both symmetric and transitive relations
- Formalization
  - Relational knowledge: $R_k = \{(w_i, r_k, w_j)\}\ \forall 1 \leq k \leq m$
  - Difference vector: $\mathbf{d}_{ij} = (\mathbf{w}_j - \mathbf{w}_i) \in \mathbb{R}^d$
  - Construct a matrix $\mathbf{D}_k \in \mathbb{R}^{d \times |R_k|}$ stacking difference vectors

$$\mathbf{D}_k = [\cdots \mathbf{d}_{ij} \cdots]\ \forall \{(i,j) : (w_i, r_k, w_j) \in R_k\}$$

- **Assumption**: $\mathbf{D}_k$ is approximately of low rank p

$$\mathbf{D}_k \approx \mathbf{U}_k \mathbf{A}_k^\top \text{ where } \mathbf{U}_k \in \mathbb{R}^{d \times p} \text{ and } p \ll d$$

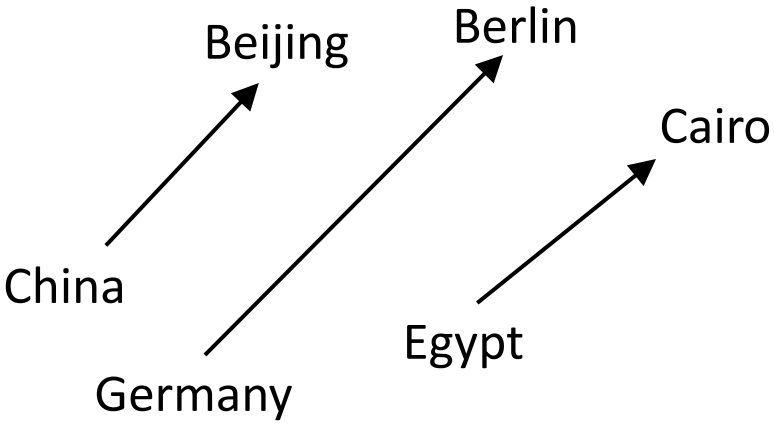
Rank-1 subspace regularization

- p = 1 ➔ $\mathbf{D}_k \approx \mathbf{u}_k \boldsymbol{\alpha}_k^T$ where $\mathbf{u}_k \in \mathbb{R}^d$ and $\boldsymbol{\alpha}_k \in \mathbb{R}^{|R_k|}$
  - All difference vectors for the same relation type are collinear
- Minimizes a joint objective:

$$-\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-c}^{t+c}) + \frac{\lambda}{2\,|R|}\sum_{k=1}^{m}\left\|\mathbf{D}_k - \mathbf{u}_k\boldsymbol{\alpha}_k^\top\right\|_F^2$$

$$\text{s.t. } \boldsymbol{\alpha}_k \geq 0 \,\forall \text{ asymmetric } r_k, \|\mathbf{u}_k\|_2 = 1, |(\boldsymbol{\alpha}_k)_l| \leq c.$$

- Example: relation “capital-of”
  - Our method: $v(Berlin) - v(Germany) \approx \alpha\{v(Beijin) - v(China)\}$
  - CTM: $v(Berlin) - v(Germany) \approx v(Beijin) - v(China)$



Optimization for word vectors

- We use parallel asynchronous SGD with negative sampling
  - Each thread works on a predefined segment of the text corpus by:
    - sampling a target word and its local context window, and
    - updating the parameters stored in a shared memory
  - Puts our regularizer on input embeddings
- Gradient updates by regularization

$$\mathbf{w}_i \longleftarrow \mathbf{w}_i - \eta \frac{\lambda}{|R|} \left[\sum_{j:(w_i,r_k,w_j)\in R} \left(\mathbf{w}_i - \mathbf{w}_j + \mathbf{u}_k \alpha_{k_{ij}}\right) + \sum_{j:(w_j,r_k,w_i)\in R} \left(\mathbf{w}_i - \mathbf{w}_j - \mathbf{u}_k \alpha_{k_{ji}}\right) \right]$$

Optimization for relation parameters

- Optimizes $\mathbf{u}_k$ and $\boldsymbol{\alpha}_k$ by solving the batch optimization problem

$$\min_{\mathbf{u}_k, \boldsymbol{\alpha}_k} \left\| \mathbf{D}_k - \mathbf{u}_k \boldsymbol{\alpha}_k^\top \right\|_F^2, \text{ s.t. } \|\mathbf{u}_k\|_2 = 1, |\boldsymbol{\alpha}_k| \leq c$$

  - Launches a thread that keeps solving the problem
  - Alternates between two least-squares sub-problems for $\mathbf{u}_k$ and $\boldsymbol{\alpha}_k$
  - Uses projected gradient descent with an asynchronous batch update

Data sets

- Text corpus
  - English Wikipedia: ~4.8M articles and ~2B tokens
- Relational knowledge data
  - **WordRep** (Gao et al., 2014)
    - 44,584 triplets (w_i, r, w_j) of 25 relation types from WordNet etc.
  - **Google word analogy** (Mikolov et al., 2013)
    - 19,544 quadruplets of a:b::c:d from 550 triplets (w_i, r, w_j)
- Relations used for our training
  - Split the WordRep triplets randomly to <train>:<test> = 4:1
  - Remove from <train> triplets containing words in Google analogy data

Results (1): Knowledge-base completion

- Task:
  - Complete (x, r, y) by predicting y^* for the missing word y given x and r
- Inference by RELSUB
  - y^* = the word closest to the rank-1 subspace $\boldsymbol{x} + s\boldsymbol{r}$ where $|s| \leq c$
- Inference by RELCONST
  - y^* = the word closest to $\boldsymbol{x} + \boldsymbol{r}$

Relation-type	RELCONST	RELSUB
capital-cities	48.15	**59.26**
currency	**58.33**	50.00
city-in-state	17.88	**18.94**
gender	44.44	**50.00**
similar-to	5.44	**7.26**
made-of	0	0
has-context	**10.00**	8.26
is-a	1.35	**1.83**
part-of	17.50	**19.00**
instance-of	8.40	**12.98**
derived-from	9.14	**10.27**
antonym	20.00	**20.62**
entails	0	**4.35**
causes	0	0
member-of	13.43	**26.87**
related-to	0	0
attribute	**11.76**	8.82
SEMANTIC	7.47	**8.44**
adjective-to-adverb	10.14	**47.83**
plural-verbs	61.25	**71.77**
plural-nouns	66.70	**71.89**
comparative	70.00	**75.00**
superlative	66.67	**77.78**
nationality	**85.71**	**85.71**
past-tense	42.20	**66.84**
present-participle	45.76	**47.62**
SYNTACTIC	54.88	**65.38**
TOTAL	24.61	**29.03**

Results (2): Word analogy

- Task:
  - Complete *a*:*b*::*c*:*d* by predicting *d** for the missing word *d* given *a*, *b* and *c*
- Inference by RELSUB and RELCONST
  - *d** = the word closest to ***c*** + ***b*** - ***a***

Relation-type	CBOW	RELCONST	RELSUB
SEMANTIC	68.37	69.85	**70.96**
SYNTACTIC	**66.69**	65.42	65.96
TOTAL	67.48	67.43	**68.22**

Conclusion and future work

- **Conclusion**
  - We present a novel approach for modeling relational knowledge based on rank-1 subspace regularization
  - We show the effectiveness of the approach on standard tasks
- **Future work**
  - Investigate the interplay between word frequencies and regularization strength
  - Study higher-rank subspace regularization
    - Formalization for word similarity
  - Evaluate our methods by other metrics including downstream tasks

Thank you very much.
Any questions?