Unsupervised Learning of Distributional Relation Vectors

(Modeling Semantic Relatedness using Global Relation Vectors)

Shoaib Jameel Zied Bouraoui Steven Schochaert

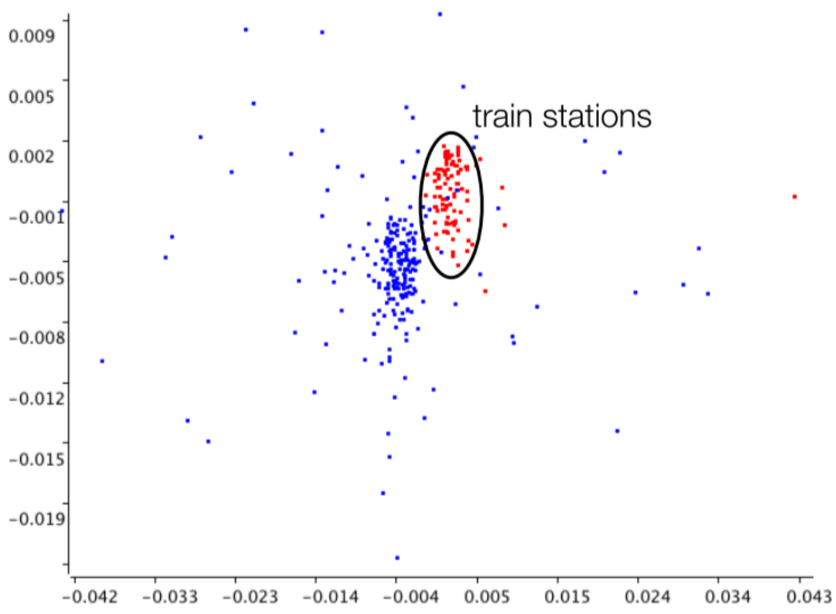
ACL 2018

Mingshi Cai

i@unoiou.com

Contents

Introduction

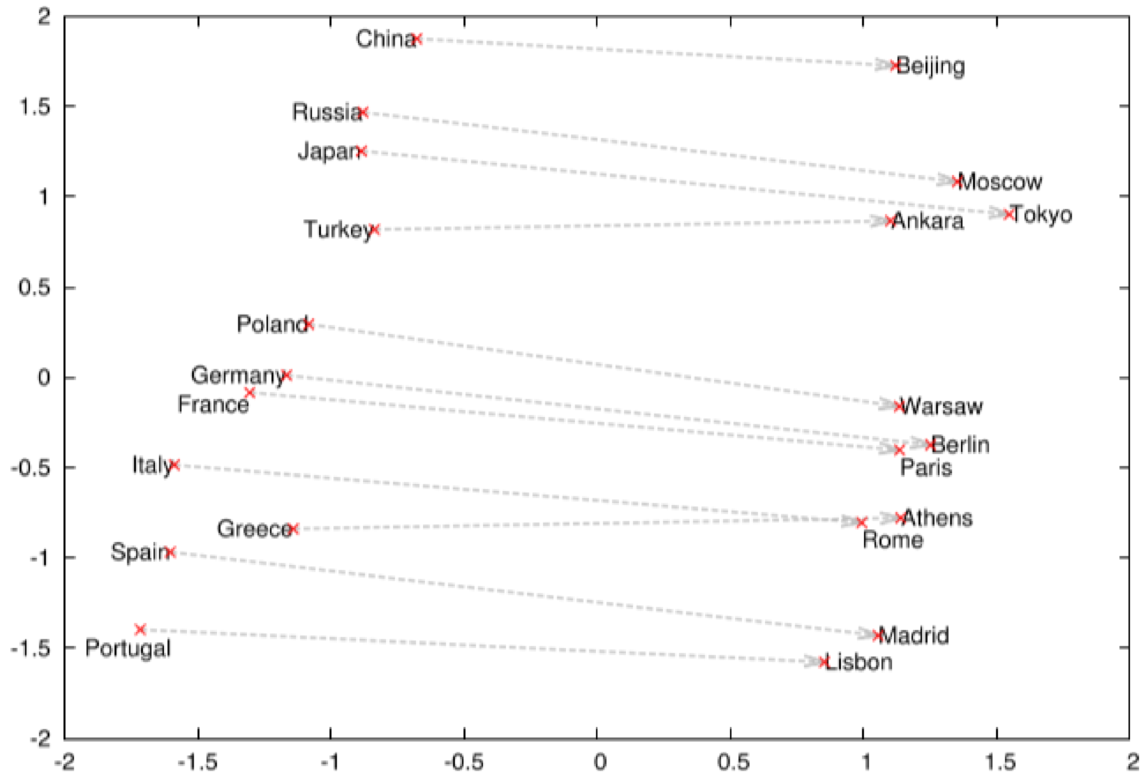


a is to **b** what **c** is to ?

$$\cos(w_b - w_a + w_c, w_d)$$

Induction with learned entity embeddings

Introduction

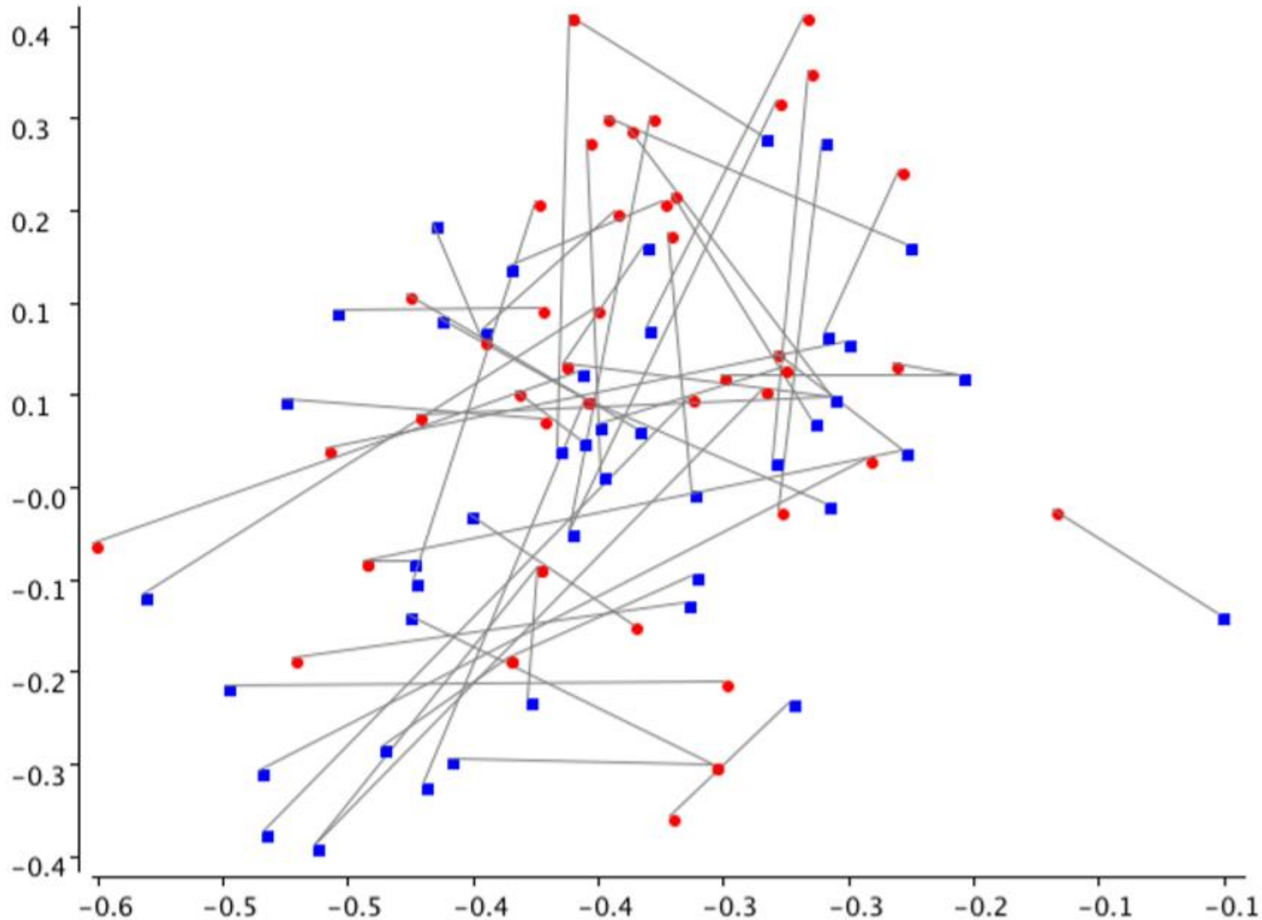


(Mikolov et al, 2013)

word pair	cos
(horse, horses)	0.84
(boy, girl)	0.79
(madrid, spain)	0.73
(london, england)	0.69
(spain, madrid)	0.68
(walk, walks)	0.65

What about relations?

Introduction



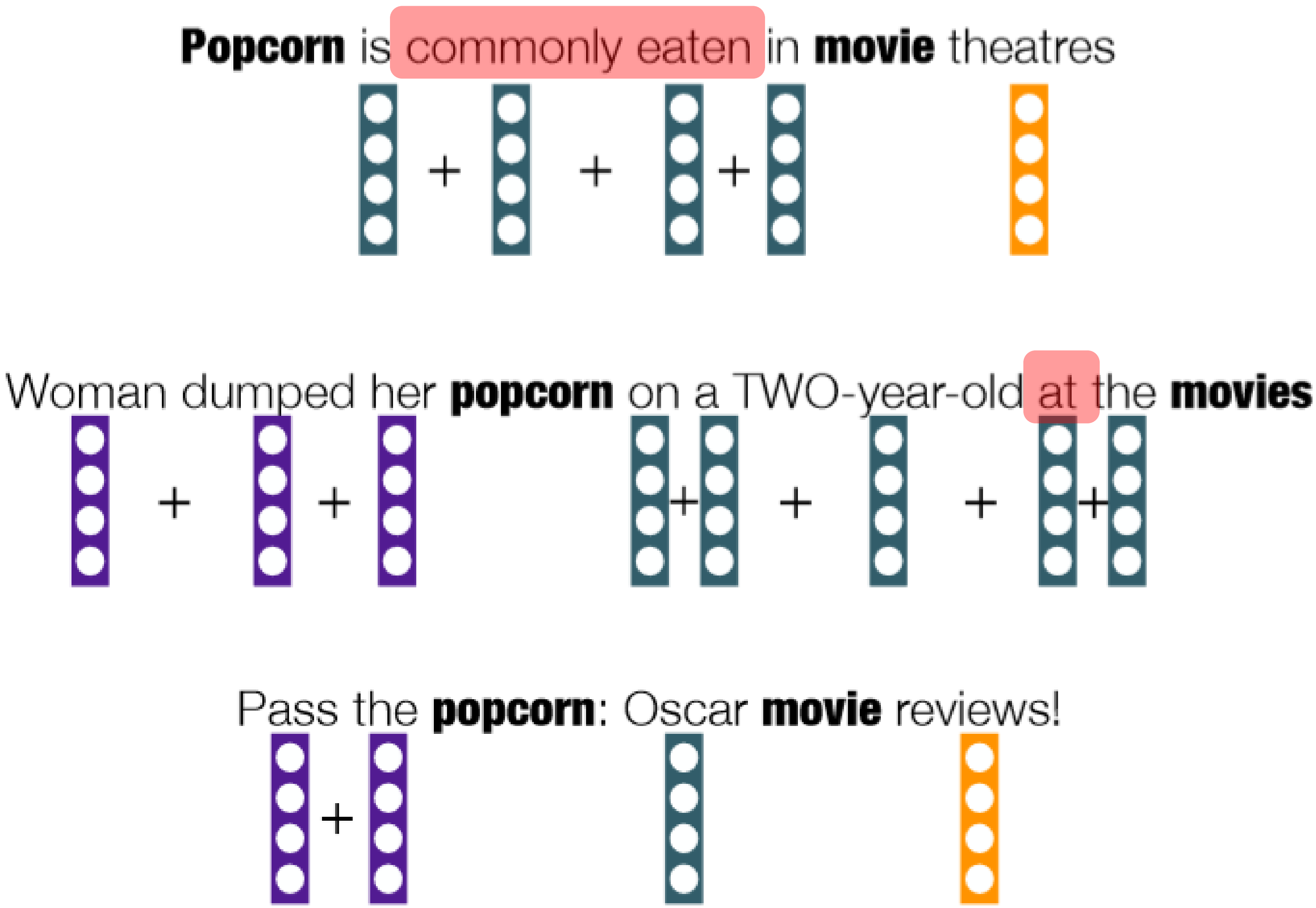
What about relations?

Problem formulation

- Given a pair of ***words (i, k),*** we want to learn a vector that represents their relationship.

- **Main strategy**: use the distribution of context words that appear in sentences which contains ***word i*** and ***word k***.

Standard approach: averaging word vectors



GloVe
word embedding model

$$\sum_{i,j}^{N} f(X_{i,j})(v_i^T v_j + b_i + b_j - log(X_{i,j}))^2$$

Co-occurred Matrix item

Word vectors

bias

$$PMI_X(i,j) = \log\left(\frac{P(i,j)}{P(i)P(j)}\right)$$

$$\sum_{i} \sum_{\substack{j \\ x_{ij} \neq 0}} f(x_{ij})(w_i \cdot \tilde{w}_j + b_i + \tilde{b_j} - PMI_X(i,j))^2$$

$$f(x) = \begin{cases} (x/xmax)^{0.75}, & \text{if } x < xmax \\ 1, & \text{if } x >= xmax \end{cases}$$

(Pennington et al., 2014)

Variant GloVe

$$\sum_{i} \sum_{\substack{j \\ x_{ij} \neq 0}} f(x_{ij})(w_i \cdot \tilde{w}_j + b_i + \tilde{b}_j - \boldsymbol{PMI}_X(i,j))^2$$

$$\sum_{i} \sum_{j \in J_i} \frac{1}{\sigma_j^2}(w_i \cdot \tilde{w}_j + \tilde{b}_j - \boldsymbol{PMI}_S(i,j))^2$$

Learning word vectors

$$\sum_{i} \sum_{j \in J_i} \frac{1}{\sigma_j^2} \Big(\overbrace{w_i}^{\text{“target vector” for word i}} \cdot \underbrace{\tilde{w}_j}_{\text{“context vector” for word j}} + \tilde{b}_j - PMI_S(i,j) \Big)^2$$

Learning word vectors

$$\sum_i \sum_{j \in J_i} \frac{1}{\sigma_j^2} (w_i \cdot \tilde{w}_j + \tilde{b}_j - \boxed{PMI_S(i,j)})^2$$

Smoothed estimation of pointwise mutual information

$$PMI_S(i,j) = \log \left(\frac{P(i,j)}{P(i)P(j)} \right)$$

$$P(i) = \frac{x_{i*} + \alpha}{x_{**} + n\alpha}$$

$$P(i,j) = \frac{x_{ij} + \alpha}{x_{**} + n^2\alpha}$$

Learning word vectors

bias term

$$\sum_i \sum_{j \in J_i} \boxed{\frac{1}{\sigma_j^2}} (w_i \cdot \tilde{w}_j + \boxed{\tilde{b_j}} - PMI_S(i,j))^2$$

Weighting the importance of context words

$$M = 2 \cdot |\{j \,:\, x_{ij} > 0\}|.$$

$$\sigma_j^2 = \frac{1}{|J_j^{-1}|} \sum_{i \in J_j^{-1}} (w_i \cdot \tilde{w}_j + \tilde{b_j} - PMI_S(i,j))^2$$

Learning word vectors

$$PMI_W(i,j) = w_i \cdot \tilde{w}_j + \tilde{b_j}$$

$$PMI_S(i,j) = \log\left(\frac{P(i,j)}{P(i)P(j)}\right)$$

$$PMI_W(i,j) \approx PMI_S(i,j)$$

Learning global relation vectors

- The main idea is r_{ik} will capture which context words ***j*** are most closely associated with the *word* pare ***(i, k).***

- We need statistics on ***(source word, context word, target word)*** triples.

Learning global relation vectors
Co-occurrence statistics for triples

$$y_{ijk} = \sum_{l=1}^{m} \sum_{p \in \mathcal{P}_i^l} \sum_{q \in \mathcal{P}_j^l} \sum_{r \in \mathcal{P}_k^l} weight(p, q, r)$$

$$\mathcal{P}_i^l \subseteq \{1, ..., n_l\}$$

$$weight(p, q, r) \; = \; \max(\tfrac{1}{q-p}, \tfrac{1}{r-q})$$

(p < q < r ***and*** *r –p <= W)*

Learning global relation vectors

Co-occurrence statistics for triples

$$SI^1(i,j,k) = \log\left(\frac{P(i,j)P(i,k)P(j,k)}{P(i)P(j)P(k)P(i,j,k)}\right)$$

$$SI^2(i,j,k) = \log\left(\frac{P(i,j,k)}{P(i)P(j)P(k)}\right)$$

$$SI^3(i,j,k) = \log\left(\frac{P(i,j,k)}{P(i,k)P(j)}\right)$$

$$SI^4(i,j,k) = \log\left(\frac{P(i,k|j)}{P(i|j)P(k|j)}\right)$$

$$PMI(i,j) + PMI(j,k) - SI^1(i,j,k) = SI^3(i,j,k)$$

$$SI^2(i,j,k) - PMI(i,j) - PMI(j,k) = SI^4(i,j,k)$$

Learning global relation vectors

Co-occurrence statistics for triples

$$\sum_{i} \sum_{j \in J_i} \frac{1}{\sigma_j^2} (w_i \cdot \tilde{w}_j + \tilde{b_j} - PMI_S(i,j))^2$$

$$\Downarrow$$

$$\sum_{j \in J_{i,k}} (r_{ik} \cdot \tilde{w}_j + \tilde{b_j} - SI(i,j,k))^2$$

Learning relation vectors

Overall representation of relationship between words i and k:

$$\boxed{w_i \oplus w_j} \oplus \boxed{r_{ik} \oplus s_{ik} \oplus t_{ik}} \oplus \boxed{r_{ki} \oplus s_{ki} \oplus t_{ki}}$$

word vector representations

relation vectors obtained from sentences where i appears before k

relation vectors obtained from sentences where k appears before i

Learning relation vectors

Overall representation of relationship between words i and k:

$$w_i \oplus w_j \oplus \boxed{r_{ik}} \oplus s_{ik} \oplus t_{ik} \oplus \boxed{r_{ki}} \oplus s_{ki} \oplus t_{ki}$$

relation vectors obtained from
context words that occur
between i and k

Learning relation vectors

Overall representation of relationship between words i and k:

$$w_i \oplus w_j \oplus r_{ik} \oplus \boxed{s_{ik}} \oplus t_{ik} \oplus r_{ki} \oplus \boxed{s_{ki}} \oplus t_{ki}$$

relation vectors obtained from context words that occur **before** i and k

Learning relation vectors

Overall representation of relationship between words i and k:

$$w_i \oplus w_j \oplus r_{ik} \oplus s_{ik} \oplus \boxed{t_{ik}} \oplus r_{ki} \oplus s_{ki} \oplus \boxed{t_{ki}}$$

relation vectors obtained from
context words that occur
after i and k

Evaluation relation induction

Table 1: Results for the relation induction task.

Google Analogy							
	Diff	Conc	Avg	R_{ik}^1	R_{ik}^2	R_{ik}^3	R_{ik}^4
Acc	90.0	89.0	89.9	90.0	**92.3**	90.9	90.4
Pre	81.6	78.7	80.8	79.9	87.1	83.2	81.1
Rec	82.6	83.9	83.9	86.0	84.8	84.8	85.5
F1	82.1	81.2	82.3	82.8	**85.9**	84.0	83.3

DiffVec							
	Diff	Conc	Avg	R_{ik}^1	R_{ik}^2	R_{ik}^3	R_{ik}^4
Acc	29.5	28.9	29.7	29.7	**31.3**	30.4	30.1
Pre	19.6	18.7	20.4	21.5	22.9	21.9	22.3
Rec	23.8	22.9	23.7	24.5	25.7	25.3	22.9
F1	21.5	20.6	21.9	22.4	**24.2**	23.5	22.6

Evaluation: relation induction

Table 2: Results for the relation induction task using alternative word embedding models.

	GloVe				SkipGram				CBOW			
	Google		DiffVec		Google		DiffVec		Google		DiffVec	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Diff	90.0	81.9	21.2	13.9	89.8	81.9	21.7	14.5	89.9	82.1	17.4	9.7
Conc	88.9	80.4	20.2	11.9	89.2	81.6	20.5	12.0	89.1	81.1	16.4	7.7
Avg	89.8	82.1	21.4	13.9	90.2	82.4	21.8	14.4	89.8	82.2	17.5	10.0
R^1_{ik}	89.7	81.7	20.9	12.5	89.4	81.2	21.1	12.3	89.8	81.9	17.2	9.2
R^2_{ik}	90.0	82.8	21.2	13.4	89.1	81.3	21.1	12.9	90.2	82.4	17.7	10.0
R^3_{ik}	90.0	82.3	20.0	11.2	89.5	81.1	20.5	12.3	89.5	81.1	17.2	9.6
R^4_{ik}	90.0	82.5	20.0	11.4	88.9	80.8	20.6	12.1	90.5	82.2	17.1	8.4

Evaluation: relation induction

Table 3: Relation induction without position weighting (left) and without the relation vectors s_{ik} and t_{ik} (right).

	Google		DiffVec	
	Acc	F1	Acc	F1
R_{ik}^1	89.7	82.4	30.2	22.2
R_{ik}^2	91.0	83.4	30.8	24.1
R_{ik}^3	90.4	83.2	30.1	22.3
R_{ik}^4	90.2	82.9	29.1	21.2

	Google		DiffVec	
	Acc	F1	Acc	F1
R_{ik}^1	90.0	82.5	29.9	22.3
R_{ik}^2	92.3	85.8	31.2	24.2
R_{ik}^3	90.5	83.2	30.2	23.0
R_{ik}^4	90.3	83.1	29.8	22.3

Evaluation: Measuring Degrees of Prototypicality

Table 4: Results for measuring degrees of prototypicality (Spearman $\rho \times 100$).

Diff	Conc	Avg	R^1_{ik}	R^2_{ik}	R^3_{ik}	R^4_{ik}
17.3	16.7	21.1	22.7	**23.9**	21.8	22.2

Evaluation: relation extraction

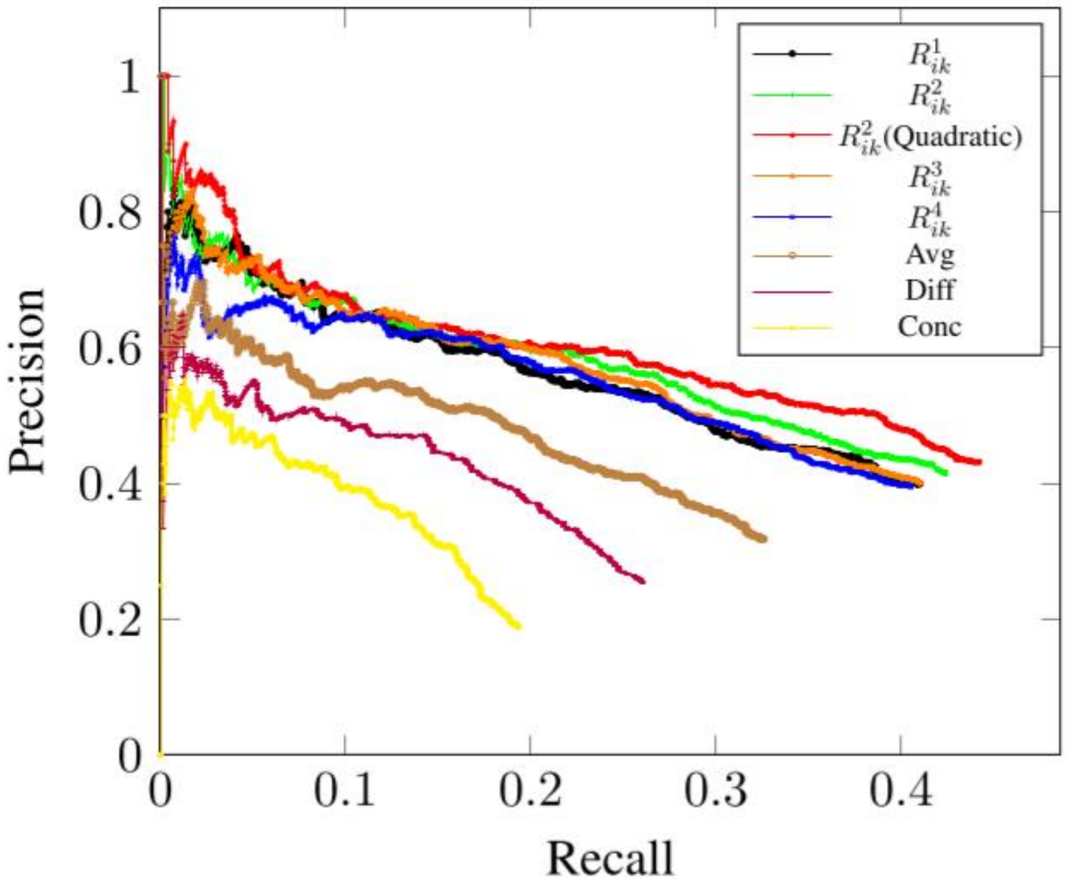


Figure 1: Results for the relation extraction from the NYT corpus: comparison with the main baselines.

Evaluation: relation extraction

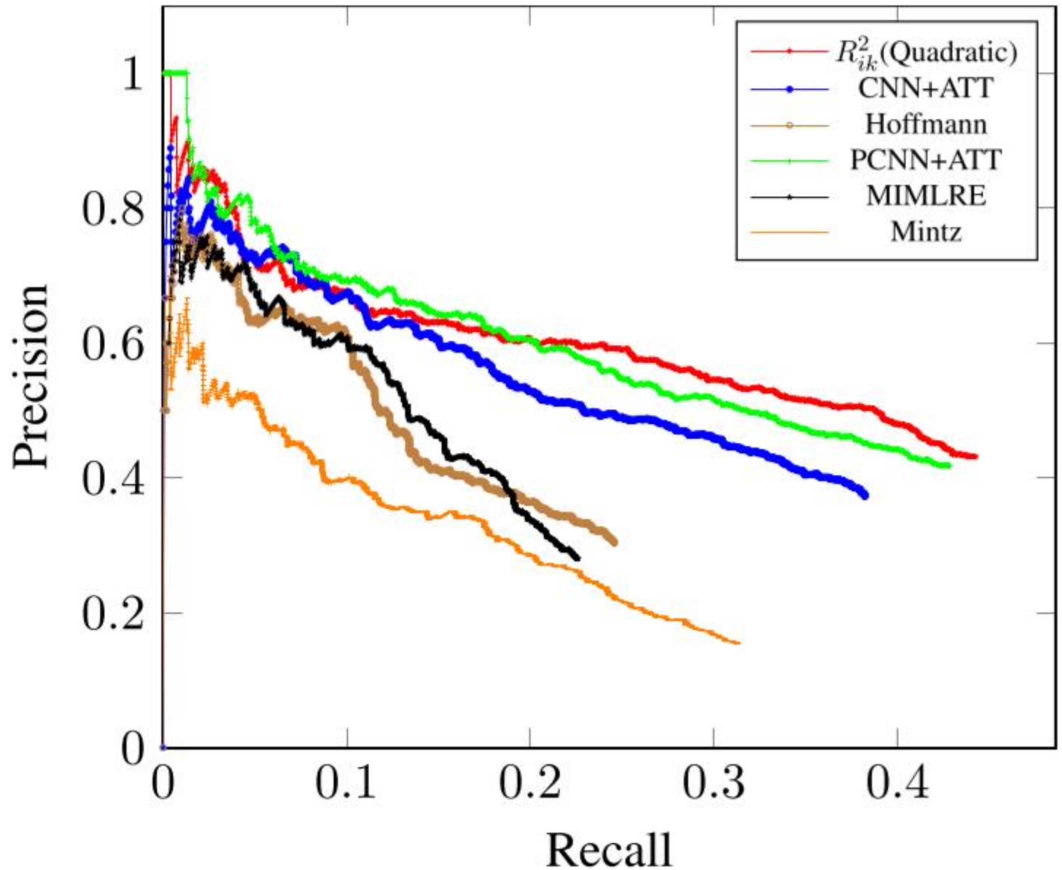


Figure 2: Results for the relation extraction from the NYT corpus: comparison with state-of-the-art neural network models.

Conclusions

- Unsupervised method to learn relation vectors from co-occurrence statistics
- Main motivation:
  - Supporting analogical inferences for knowledge base completion
  - Supporting relation induction for knowledge base completion
  - Use relation vectors to complement word vectors in NLP tasks
- Future Work:
  - Dimensionality reduction of relation vectors
  - Learn commonsense knowledge from relation vectors