Building Joint Spaces for Relation Extraction

Chang Wang, Liangliang Cao, James Fan

{*changwangnk, liangliang.cao, jfan.us*}*@gmail.com*

June 27, 2016

Goals

Detect semantic relations between entities

"birthplace" relation

Born in **Hodgenville, Kentucky**, **Lincoln** grew up on the western frontier in Kentucky and Indiana.

Improve the coverage of existing structured knowledgebases

"birthplace" relation

Person	Birthplace
Lincoln	Hodgenville, Kentucky
...	...

Related Work

Relation Extraction

- Maximum entropy [Kambhatia, 2004]
- Convolution kernel [Colins and Duffy, 2001]
- Distant supervision [Mintz et al., 2009]

Knowledgebase Completion

- RESCAL [NIckel et al., 2001]
- TransE [Bordes et al, 2013]

Challenges

Challenge 1

How to leverage the fact that similar arguments are often associated with similar relations.

For example, similar diseases are often associated with similar treatments, causes, etc.

Challenges

Challenge 1

How to leverage the fact that similar arguments are often associated with similar relations.

Challenge 2

Require relation specific term embeddings.

For example, "sign" and "symptom" may have similar semantics for most scenarios but they are very different for medical domain, where "signs" are what a doctor sees, "symptoms" are what a patient experiences

Challenges

Challenge 1

How to leverage the fact that similar arguments are often associated with similar relations.

Challenge 2

Require relation specific term embeddings.

Challenge 3

Amount of labeled relation data is often very limited.

which makes overfitting a major issue.

Overview of our approach

Input: Given two argument sets $(\mathbf{X}, \mathbf{Y})$ which are associated with the desired relation r. For example:

1. **X**: list of persons
2. **Y**: list of locations
3. r: "birthplace" relations

Term pairs with r

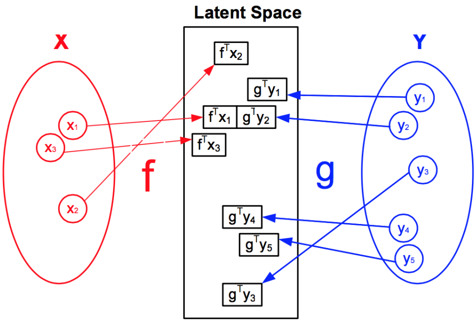
("Abraham Lincoln", "Hodgenville, Kentucky")

Term pairs without r

("Abraham Lincoln", "New York")

Overview of our approach

Construct a joint space and relation specific term embeddings



Note the mapping function f^T and g^T satisfies

- pairs with r are mapped to the same location
- pairs without r are separated from each other
- preserve neighborhood relationships

Note the last item is to alleviate overfitting when labels are not sufficient.

In our study, terms are represented as word vectors, such as Word2Vec or Latent Semantic Analysis embeddings.

Notation

Let W_x and W_y represent the nearest neighbor graphs for $\mathbf{X}$ and $\mathbf{Y}$ such that

$$W_x(i,j) = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

$$W_y(i,j) = \begin{cases} 1 & \text{if } y_i \text{ and } y_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

Then we consider the matrix

$$W = \begin{pmatrix} W_x & 0 \\ 0 & W_y \end{pmatrix}$$

Its corresponding row sum matrix D such as $D(i,i) = \sum_j W(i,j)$ and combinatorial Laplacian

$$L = D - W$$

Notation of Similarity and Dissimilarity Matrix

Similarity matrix

$$W_s^{x,y}(i,j) = \begin{cases} 1 & \text{if } r \text{ is held between } x_i \text{ and } y_j \\ 0 & \text{otherwise} \end{cases}$$

Dissimilarity matrix

$$W_d^{x,y}(i,j) = \begin{cases} 0 & \text{if } r \text{ is held between } x_i \text{ and } y_j \\ 1 & \text{otherwise} \end{cases}$$

Row sum matrix and Laplacian

$$D_s(i,i) = \sum_j W_s(i,j) \qquad D_d(i,i) = \sum_j W_d(i,j)$$

$$L_s = D_s - W_s \qquad L_d = D_d - W_d$$

Objectives

Preserving neighborhood information

$$S_1 = 0.5 \sum_i \sum_j ||f^T x^i - f^T x^j||^2 W_x(i,j) + 0.5 \sum_i \sum_j ||g^T y^i - g^T y^j||^2 W_y(i,j)$$

Related Term to be projected to similar locations

$$S_2 = 0.5 \sum_i \sum_j ||f^T x^i - g^T x^j||^2 W_s(i,j)$$

Unrelated Term to be separated in the new space

$$S_3 = 0.5 \sum_i \sum_j ||f^T x^i - g^T x^j||^2 W_d(i,j)$$

Solution

Overall cost function

$$Cost(f, g) = (\mu S_1 + S_2)/S_3$$

Let $\gamma = [f^T, g^T]]$ be a $(p + q) \times d$ matrix, we have

Theorem (Eigen Decomposition Theorem)

The γ that minimize $Cost(f, g)$ is given by the eigen vectors corresponding to the smallest non-zero eigen-values of

$$Z(\mu L + L_s)Z^T \epsilon = \lambda Z L_d Z^T \epsilon$$

Based on this theorem, we can solve γ to construct the joint space, and then a SVM model will be trained to detect the relationship r.

Experiments

Baselines:

- Affine matching: LSA and Word2Vec
- Feature Concatenation: sum, difference, product,
- RESCAL [Nickel et al., 2001]
- TransE [Bordes et al., 2013]

Experiments of Extracting DBpedia Relations

Table 1: F_1 Scores for DBpedia Relation Extraction Experiment

	Number of Training /Test Examples	LSA concatenated features	word2vec concatenated features	word2vec affine matching no expansion	Joint Space $\mu = 0$ no expansion	Joint Space $\mu = 1$ no expansion	word2vec affine matching	Joint Space $\mu = 0$	Joint Space $\mu = 1$
birthplace	720K/308K	0.3465	0.3866	0.3935	0.3852	0.3985	0.3739	0.3493	0.3734
country	737K/316K	0.4028	0.3641	0.3716	0.4040	0.4425	0.4218	0.4597	0.4206
hometown	877K/376K	0.3546	0.3315	0.3336	0.3796	0.3818	0.3386	0.3372	0.3601
instrument	959K/435K	0.0195	0.5295	0.5289	0.6307	0.6176	0.5672	0.5818	0.5847
militarybranch	765K/353K	0.3348	0.3991	0.3998	0.4370	0.4299	0.4152	0.4026	0.4079
nationality	1.1M/468K	0.4759	0.4257	0.4294	0.4760	0.4760	0.4370	0.4416	0.4506
occupation	762K/327K	0.0292	0.2432	0.2470	0.4095	0.3452	0.3907	0.4372	0.3516
religion	1.2M/525K	0.3055	0.2945	0.2887	0.3634	0.3620	0.3259	0.3253	0.3370
average	892K/389K	0.2836	0.3718	0.3741	**0.4357**	0.4317	0.4088	0.4168	0.4107

On average, each relation has 0.89M training and 0.39M testing examples. Since training data is sufficient, preserving neighborhood does not help. Joint space model outperforms the other approaches.

Experiment of Extracting Medical Relations

Table 2: F_1 Scores for Medical Relation Extraction Experiment

	Number of Training /Test Examples	LSA concatenated features	word2vec concatenated features	word2vec affine matching no expansion	Joint Space $\mu = 0$ no expansion	Joint Space $\mu = 1$ no expansion	word2vec affine matching	Joint Space $\mu = 0$	Joint Space $\mu = 1$
treats	461K/198K	0.2493	0.5215	0.5223	0.6181	0.6254	0.6756	0.7936	0.7913
prevents	63K/27K	0.2637	0.5734	0.5699	0.5507	0.6483	0.7661	0.6917	0.7671
causes	14K/6K	0.6596	0.4220	0.4667	0.4706	0.4587	0.2069	0.2697	0.4235
location_of	1.26M/539K	0.3982	0.3072	0.3111	0.4287	0.4145	0.4762	0.6969	0.6919
diagnoses	9K/4K	0.0370	0.5051	0.4299	0.40000	0.4286	0.4524	0.3226	0.3944
symptom_of	865K/371K	0.1865	0.3509	0.3417	0.4031	0.3943	0.3711	0.5489	0.5220
average	447K/218K	0.2991	0.4467	0.4403	0.4785	0.4950	0.4914	0.5539	**0.5984**

On average, each relation has 0.45M training and 0.22M testing examples.
Preserving neighborhood alleviates overfitting, especially for three relations with few examples.
Joint space model outperforms the other approaches.

Summary

In this paper, we propose an approach to

- detect relations from entity pairs
- construct relation specific term embedding

Benefits:

- Our method provides a close-form solution
- Our method is able to handle the situation when labeled data is not sufficient