Towards Learning to Rank Approach in Paraphrase Detection

Aleksandr Ogaltsov

Antiplagiat Company, Higher School of Economics

Introduction

We describe a method that aims to rank paraphrased text fragments. The problem arises in plagiarism-paraphrase text reusing. Given query text fragment may have several candidate paraphrases with different semantic similarity degree. We use deep learning methods in combination with metric learning and listwise learning to rank approaches. We test our framework on PPDB dataset. Our results are proposed as a baseline for such problem statement.

Problem statement and features description

Problem.

Let $Q = \{q_i\}_{i=1}^N$ be set of queries. Each q_i has the list of paraphrased text fragments d^i with associated semantic similarity degrees y^i:

$$d^i = (d_1^i, d_2^i, \dots d_{n_i}^i), \quad y^i = (y_1^i, y_2^i, \dots y_{n_i}^i),$$

where n_i is the size of d^i. Let's construct a query-fragment feature space:

$$\boldsymbol{\phi} : (q_i, d_j^i) \to \mathbf{x}_j^i \in \mathbb{R}^m.$$

Consider the dataset:

$$\mathfrak{D} = \{(\mathbf{x}^i, y^i)\}_{i=1}^N,$$

where $\mathbf{x}^i = (\mathbf{x}_1^i, \mathbf{x}_2^i \dots \mathbf{x}_{n_i}^i)$. Let f be ranking function, that produces predicted rank $\hat{y}$. Optimized objective is mean $NDCG@K$:

$$Q(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} NDCG@K_i.$$

The task is to find ranking function $\hat{f}$, such that

$$\hat{f} = \arg\max_{f \in \mathfrak{F}} Q(f(\mathbf{x}), y).$$

Features.

When we have vectors of query and paraphrased fragments we construct feature space for ranking as follows.

For each query-fragment pair $\mathbf{x}_j^i$:

$$\mathbf{x}_j^i = \begin{bmatrix} \mathbf{q}_i \\ \mathbf{d}_j^i \\ \mathbf{q}_i^T \mathbf{M} \mathbf{d}_j^i \end{bmatrix}$$

For each query-list of fragments $\mathbf{x}^i$:

$$\mathbf{x}^i = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \mathbf{x}_1^i & \mathbf{x}_2^i & \dots & \mathbf{x}_{n_i}^i \\ \vdots & \vdots & \dots & \vdots \end{bmatrix}$$

Since paraphrased fragments can have arbitrary initial ranking, the training object for ranking neural network is 3-tensor $\mathbf{X}^i$ with sub-matrices, formed by columns permutations.

Experimental results

Algorithm.

Each sub-matrix of $\mathbf{X}^i$ is mapped to scores list by fully-connected layer:

$$\mathbf{h} = \mathbf{f}(\mathbf{W}\mathbf{x}^i + \mathbf{b})\mathbf{w}.$$

We convert this scores into probability distribution over the set $\Omega = \{\pi_1, \pi_2, \dots \pi_n\}$ of scores permutations:

$$P(\pi) = \prod_{c=1}^{n} \frac{\exp(\mathbf{h}_c)}{\sum_{k=c}^{n} \exp(\mathbf{h}_k)}.$$

Therefore $\mathbf{X}^i$ generates matrix $\mathbf{P}^i$, where each row is a distribution, produced by each sub-matrix of $\mathbf{X}^i$. We learn parameters of ranking network by minimization cross-entropy between produced distribution and true distribution $\mathbf{Y}^i$:

$$L = -\mathbf{e}^T \mathbf{Y}^i \odot \log \mathbf{P}^i \mathbf{e},$$

where $\mathbf{e}$ — unit column-vector.

Evaluation.

We used Wikipedia for *fasttext* word embeddings training and 10M text fragments for unsupervised autoencoder pretraining. Train and test datasets for ranking were obtained from PPDB Paraphrase Pack which provides query and candidates with similarity scores. For each query phrase we collected five candidates with highest degree of similarity. Finally, dataset consisted of 50K pairs (q^i, d_i) with discrete similarity degree from 4 to 0, where 4 is the closest fragment.

Results.

Algorithm	Mean $NDCG@5$
Proposed approach	**0.81**
Cosine measure baseline	0.72
Random ranking	0.64
Worst case	0.51

Future work

In future, we will try to create dataset for proposed task formulation, that would be based on expert assessment and consists of longer text fragments. We will also explore ways to speed up learning in listwise setting and investigate effects of initialization of bilinear similarity matrix.

Contact information

Aleksandr Ogaltsov: ogaltsov@ap-team.ru, avogaltsov@edu.hse.ru