attr2vec: Jointly Learning Word and Contextual Attribute Embeddings with Factorization Machines

Fabio Petroni, Vassilis Plachouras, Timothy Nugent, Jochen L. Leidner

the answer company™ THOMSON REUTERS®

https://github.com/thomsonreuters/attr2vec

Motivation

The use of word embeddings is considered a *"secret sauce"* for many recent algorithms.

Popular word embeddings models:

- code.google.com/archive/p/word2vec;
- nlp.stanford.edu/projects/glove;
- github.com/facebookresearch/fastText.

Small portion of available context used.

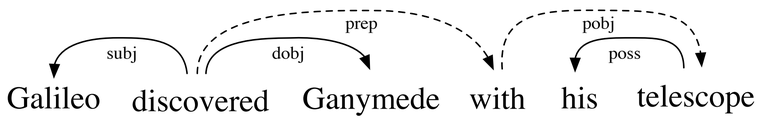
Context integrated in *ad hoc* ways.

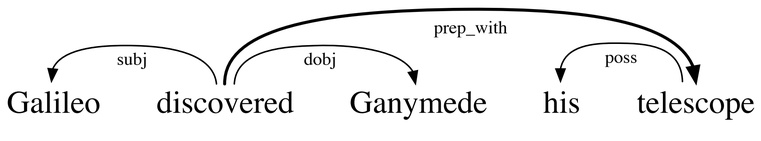
Contextual information not in vector space.

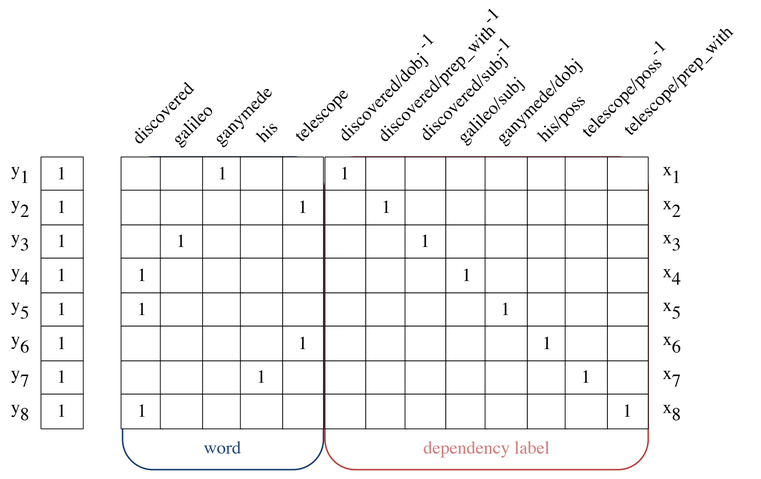
attr2vec jointly associates distributed representations with words and with generic contextual attributes.

Dependency-Based

Co-occurrence count from dependency tree:

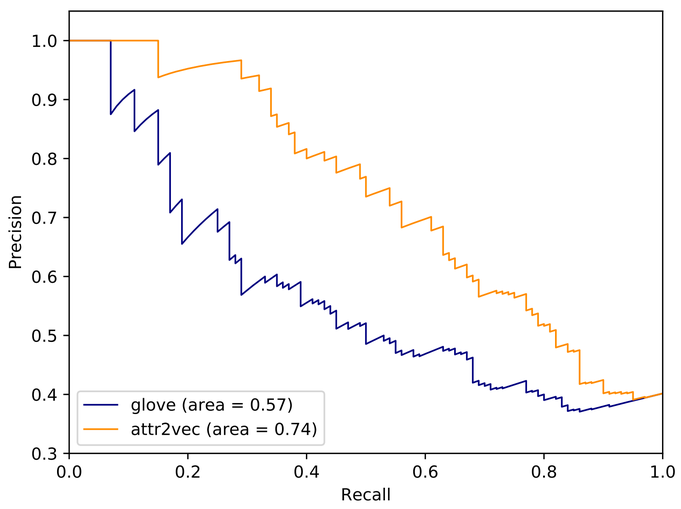






Word Similarity Experiment

Precision-recall curve when attempting to rank the similar words above the related ones on the *WordSim353* dataset.



Dependency-based embeddings exhibit more functional similarity than GloVe embeddings.

Modeling input data

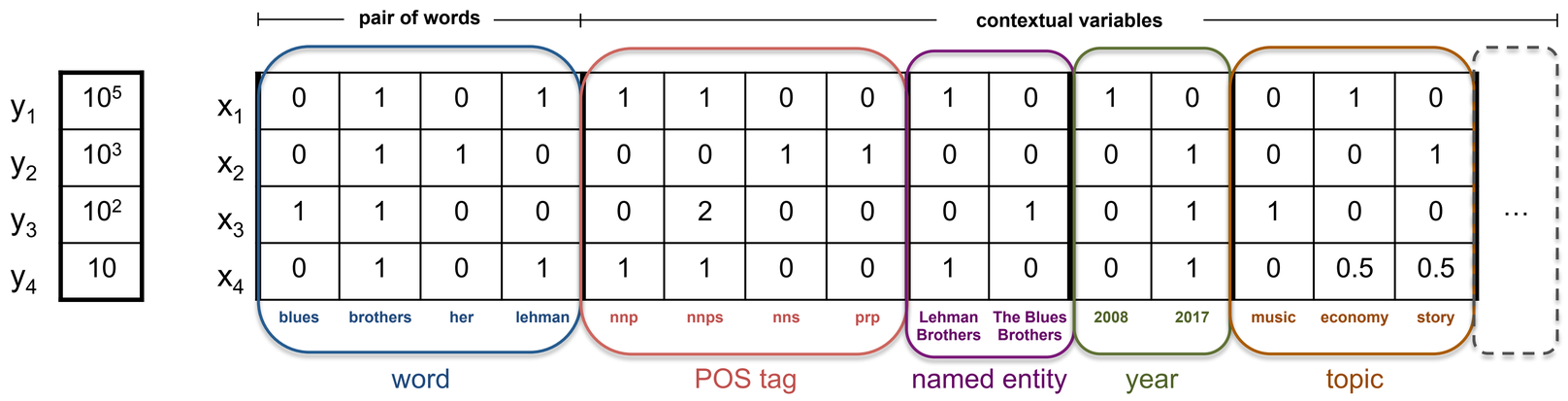
We model the input data in terms of a target vector $Y \in \mathbb{R}^m$ and feature matrix $X \in \mathbb{R}^{m \times n}$, where:

- each row $x_i \in \mathbb{R}^n$ corresponds to a feature vector;
- variables V are the set of all considered words and contextual attributes;
- in X there are as many columns as the number of variables;
- columns grouped according to the type of the variables;
- target value $y_i \in Y$ represents the number of times the feature vector x_i has been observed.

We consider a two-fold way to compute the co-occurrence count of a feature vector x_i:

1. *linear bag-of-words* **2.** *dependency-based*

Linear Bag-of-Words



Factorization Model Based On Factorization Machines

Weighted least squares objective function: $J = \sum_{k=1}^{N} f(y_k)\Big(s(x_k) - \log(y_k)\Big)^2$

The function $f(y)$ is used to reduce the importance of pairs of words that co-occur rarely.

$$s(x) = \sum_{v \in V} x_v \cdot b_v + \sum_{v_1 \in V} \sum_{v_2 \in V \setminus \{v_1\}} x_{v_1} x_{v_2} \cdot \boldsymbol{f}_{v_1}^T \boldsymbol{f}_{v_2}$$

x_v: the value of variable $v \in V$ — b_v: the bias terms — $\boldsymbol{f}_v$: latent factor vectors

Each $\boldsymbol{f}_v$ can be interpret as a dense vector representation of variable $v \in V$.

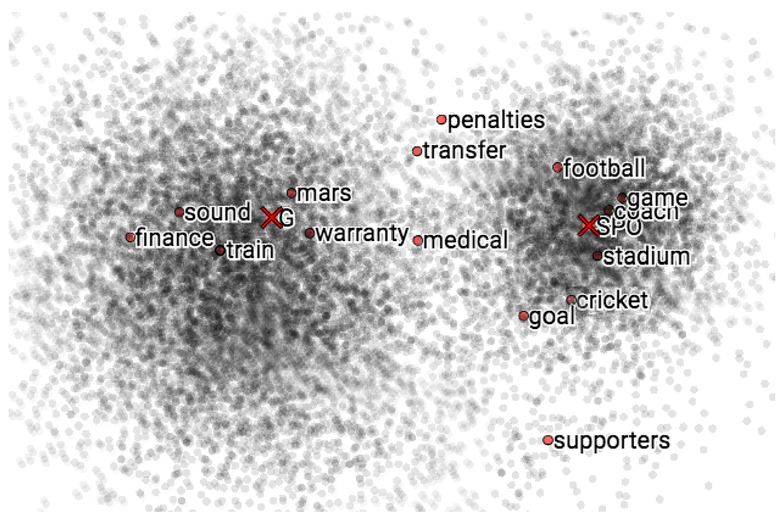
Topic Classification Experiment

embedding	*input*	*logistic regression*	*convolutional neural network*	
			static	*non-static*
random	$\vec{w_r}$	70.5 (69.3)	75.7 (77.5)	74.4 (76.2)
random	$\vec{w_r} \frown \vec{p_r}$	74.0 (73.9)	77.9 (79.7)	77.8 (79.8)
GloVe	$\vec{w_i}$	77.5 (77.5)	79.7 (81.5)	82.7 (84.3)
GloVe	$\vec{w_i} \frown \vec{p_r}$	80.2 (85.4)	82.5 (84.1)	84.5 (86.1)
GloVe	$\vec{w_i} \frown \vec{p_i}$	79.3 (83.3)	84.3 (85.8)	84.9 (86.4)
attr2vec	$\vec{w_j}$	77.5 (77.3)	80.6 (82.3)	82.8 (84.5)
attr2vec	$\vec{w_j} \frown \vec{p_j}$	80.1 (83.1)	84.9 (86.1)	**85.5** (**86.8**)

Average F1 score (and precision in parentheses) for topic prediction on the *Reuters-21578* dataset.

- $\vec{w_r}$ refer to randomly initialized word vectors and $\vec{p_r}$ to randomly initialized POS tag vector;
- $\vec{w_i}$ and $\vec{p_i}$ respectively refer to vectors independently trained with the GloVe model;
- $\vec{w_j}$ and $\vec{p_j}$ respectively refer to vectors jointly trained with attr2vec for words and POS tags.

Qualitative Evaluation



Discussion And Future Work

While attr2vec benefits from structural information, it has a price:

- the number of features is increased;
- computational cost increased w.r.t. context-agnostic models (although linear).

In future work:

- we aim to investigate the effect of adding different contextual information;
- we plan to test the resulting models in various applications.