FastText for Taxonomy enrichment

Statistical NLP 2020

Dmitriev Anton, Pronkin Alexey

Skoltech

TABLE OF CONTENTS

Introduction

- Node2Vec is a powerful tool to encode nodes of a graph as vectors
- However, it suffers from out-of-vocabulary problem
- Idea: try to map FastText embeddings of words to Node2Vec embeddings
- Project topic: make experiments to find such mapping

Experimental data

- We used RuWordNet dataset
- We build vocabulary with the following way
  - We trained Node2Vec algorithm and loaded pretrained FastText
  - Then we made pairs with FastText word embedding and corresponding Node2Vec synset embedding
  - Finally, we had a set of pairs and the task was to catch a relations between them
- We tried to optimise model to make fasttext vector after transformation close to synset embedding vector in terms of cosine distance

Used metrics

In this task we used 2 ranking metrics. In this presentation and in report them named as Soft and Hard ranking scores.

- Hard Ranking Score: penalizes for distance of true from first position of ranking top
- Soft Ranking Score: just checks is the true result in top *N* ranking results

Soft ranking score of *N* could be interpreted as percent of words for which the correct match was in top *N* ranking results so it's important because of it's interpretability.

Consequently, Hard Ranking Score *N* $\leq$ Soft Ranking Score *N*

Experiments: Baseline

The idea of baseline model was following. Just try to fit linear map between two spaces.

Unfortunately, the idea didn't work well because of relationship is too difficult for linear transformation.

Scores

- Hard ranking score

Top 1	0.014
Top 5	0.019
Top 10	0.021

- Soft ranking score

Top 1	0.014
Top 5	0.028
Top 10	0.044

Experiments: Baseline improvement

The straightforward idea: just add nonlinarity. The model configuration

- Linear from 300 to 400
- ReLU
- Linear from 400 to 500
- ReLU
- Linear from 500 to 500
- ReLU
- Linear from 500 to 500
- ReLU
- Linear from 500 to 500
- ReLU
- Linear from 500 to 400
- ReLU
- Linear from 400 to 300

Experiments: Baseline improvement

Scores

- Hard ranking score

Top 1	0.108
Top 5	0.134
Top 10	0.136

- Soft ranking score

Top 1	0.108
Top 5	0.175
Top 10	0.194

Experiments: Another approach

The adding more layers can improve score but this approach has it's limit which is very close to the previously presented model. So we need to try something different.

Idea: We can map FastText embeddings and Node2Vec ambeddings to the third space and measure cosine similarity in that space. For this we used 2 different encoders.

However, the first experiment shows that this approach maps everything to one point and it's not what we need. So we need to add negative samples.

Encoder 1:

- Linear from 300 to 400
- ReLU
- Linear from 400 to 500

Encoder 2:

- Linear from 300 to 400
- ReLU
- Linear from 400 to 500

Experiments: Another approach

Scores

- Hard ranking score

Top 1	0.192
Top 5	0.286
Top 10	0.306

- Soft ranking score

Top 1	0.192
Top 5	0.461
Top 10	0.612

Conclusion

- We tried several approaches to solve Node2Vec completion
- We proposed an approach which significantly outperforms the baseline
- However, the quality is still not good enough to use model in practice