Enhancing Language Model Representations with Attributed Network Embeddings

Jacob A. Matthews
PEER 2024

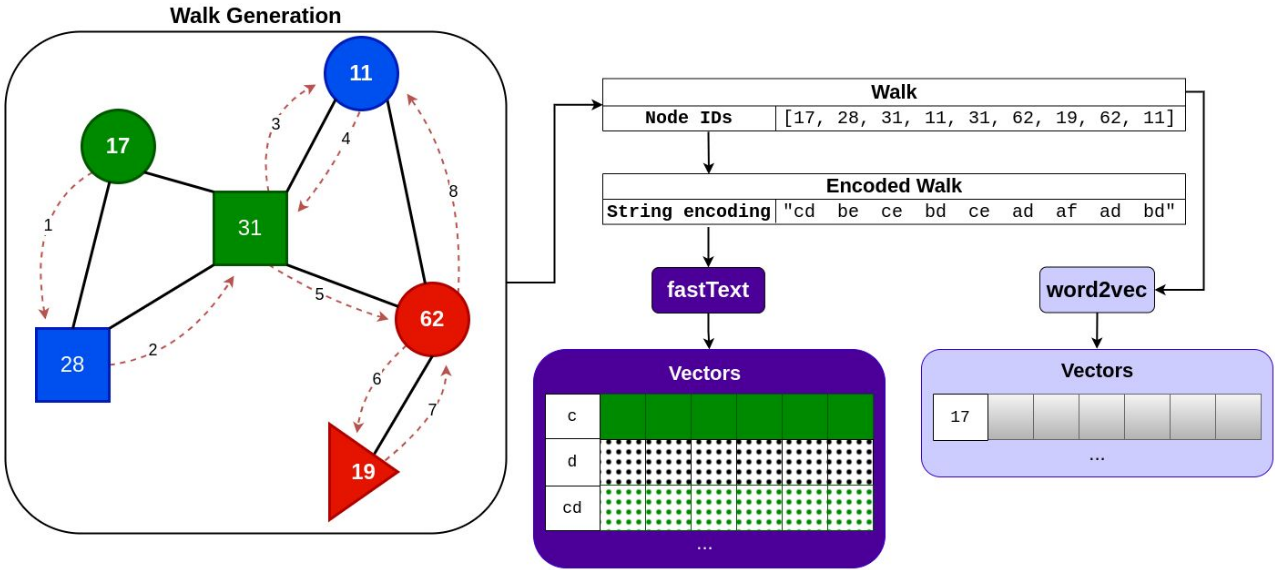
Overview

1. Learning attributed network embeddings with fastText (Bojanowski et al. 2017)
    - A fast and simple way to model graph structure and attributes jointly
2. Generating network embeddings from text
    - Use BERT to predict a node representation given the text associated with that node
        - Example: citation graphs with metadata and abstracts
    - Predicting text attributes with similarity
    - Generating contextualized word embeddings
3. Potential relevant applications
    - Semantic graphs, hypertext, ontologies…

Attributed Network Embeddings

- Network embeddings encode structural information about nodes in a network.
- A simple method still in use is node2vec (Grover and Leskovec, 2016)
  - Trains word2vec (a word embedding model) using random walks on the graph as text data
  - Treats node IDs as words and walks as text sequences
- Like word2vec, node2vec has a fixed vocabulary
  - Cannot represent nodes not seen during training
  - Has no way to represent node attributes (each node is just an arbitrary ID)
- More recent methods address these shortcomings, but can be challenging and expensive to implement.
  - They require proficiency with actual ML libraries (excludes large potential user base)
  - Can be slow and computationally expensive

Using fastText for attributed network embedding



1. Generate walks
2. Encode node IDs as pseudowords, where each character corresponds to an attribute
3. Train fastText on the encoded walks.

Why does this work?

- Unlike word2vec, fastText is a character n-gram model
  - fastText can represent any sequence of valid Unicode characters, whether they were seen in training or not.
  - In addition, fastText shares character n-gram representations across words.

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c.$$

- We can represent attributes and nodes in the same space
  - Can compute similarity values between attributes and nodes freely.
- We can update a model with out-of-sample attributes
  - Other models like attri2vec (Zheng et al. 2021) use fixed-length attribute vectors to learn node representations, which requires knowing the number of possible attributes in advance.

Network embeddings and language modeling

- Many networks are derived from (or can be easily mapped to) text data
  - Citation graphs, semantic graphs, ontologies…
- Can we predict attributed network embeddings from text sequences?
- And can we use these generated embeddings to predict attributes?
  - What happens to the model's hidden states with this approach?
    - Contextualized word embeddings?

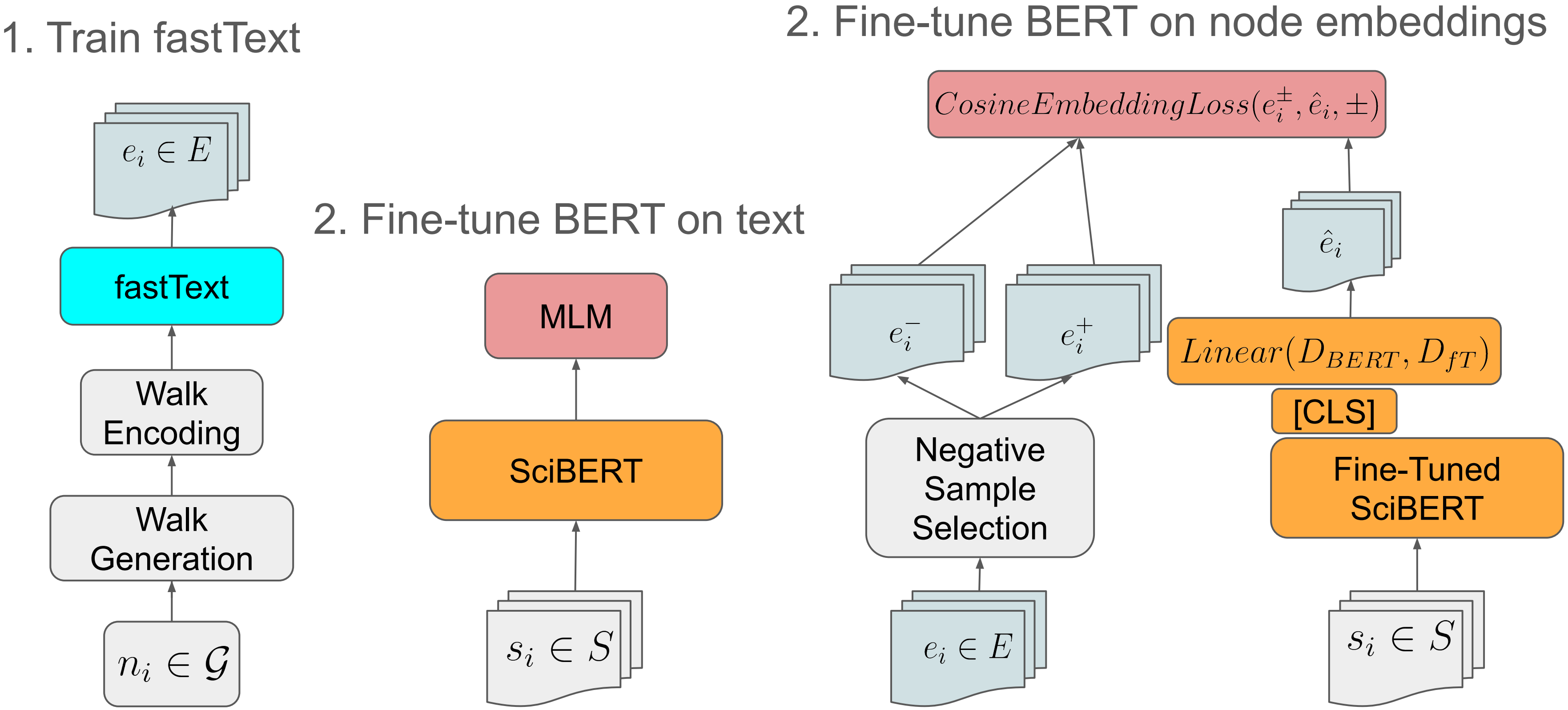
Our approach

1. Build a citation graph dataset using Semantic Scholar (S2)
   a. 25,000 nodes, 53,530 edges

```
{node_id:
      {
      title: ‘Enriching Word Vectors with Subword Information’,
      abstract: ‘Continuous…’,
      authors: [2329288, …, 2047446108],
      venue: ‘TACL’,
      year: 2016
      …
      }
```

2. Train fastText on the citation graph with metadata attributes
   a. Encode nodes as “{Decade}{Year}{Venue}{Authors}”□“ÊËÌÍÎÏÐ”
3. Train BERT to predict node embeddings from paper titles and abstracts

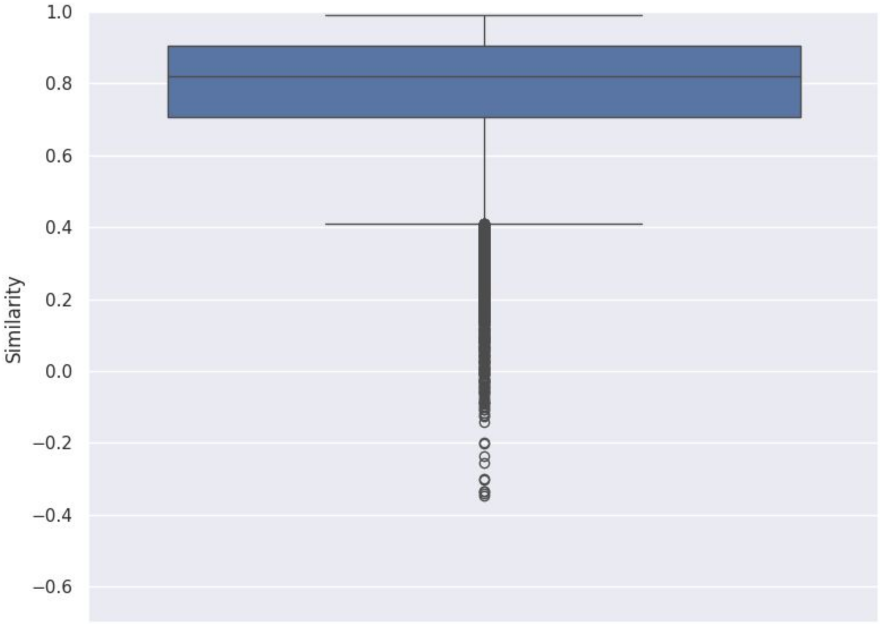
Training



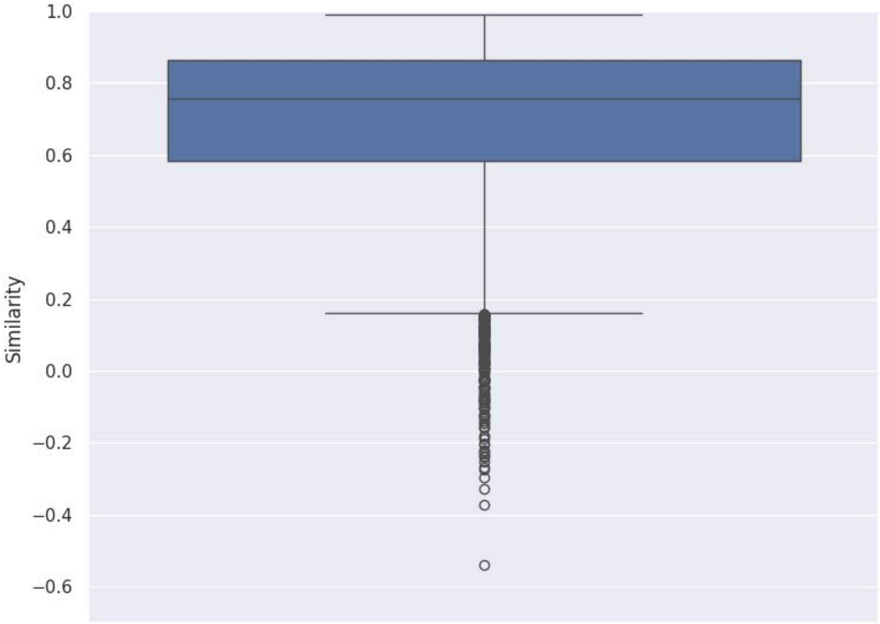
1. Train fastText
$e_i \in E$
fastText
Walk Encoding
Walk Generation
$n_i \in \mathcal{G}$
2. Fine-tune BERT on text
MLM
SciBERT
$s_i \in S$
2. Fine-tune BERT on node embeddings
$CosineEmbeddingLoss(e_i^{\pm}, \hat{e}_i, \pm)$
e_i^-
e_i^+
$\hat{e}_i$
$Linear(D_{BERT}, D_{fT})$
[CLS]
Negative Sample Selection
Fine-Tuned SciBERT
$e_i \in E$
$s_i \in S$

Similarity of BERT-estimated and fastText embeddings

Train (n=20,000)



Test (n=5,000)

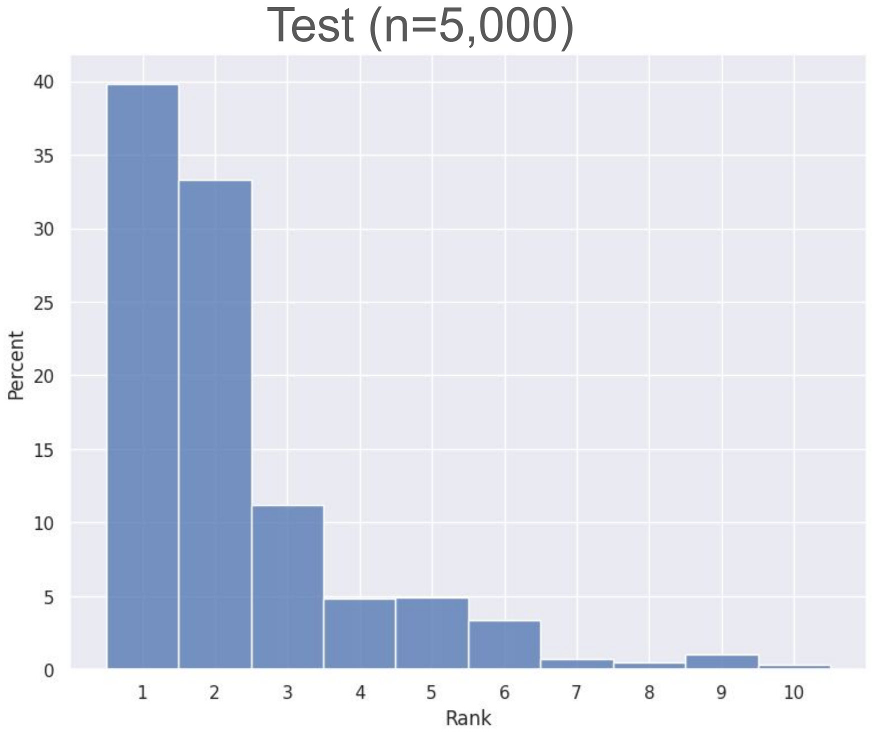


Downstream accuracy

Fine-tuning Strategy	Year Prediction 10 classes, maj. baseline = .44	Venue Prediction 100 classes, maj. baseline = .33
Text	.4474 (F1=.38)	.4074 (F1=.35)
Text + Node Embeddings	**.5110** (F1=**.49**)	**.4374** (F1=**.41**)

Attribute similarity

- >70% of estimated embeddings are most or second-most similar to the correct year embedding.

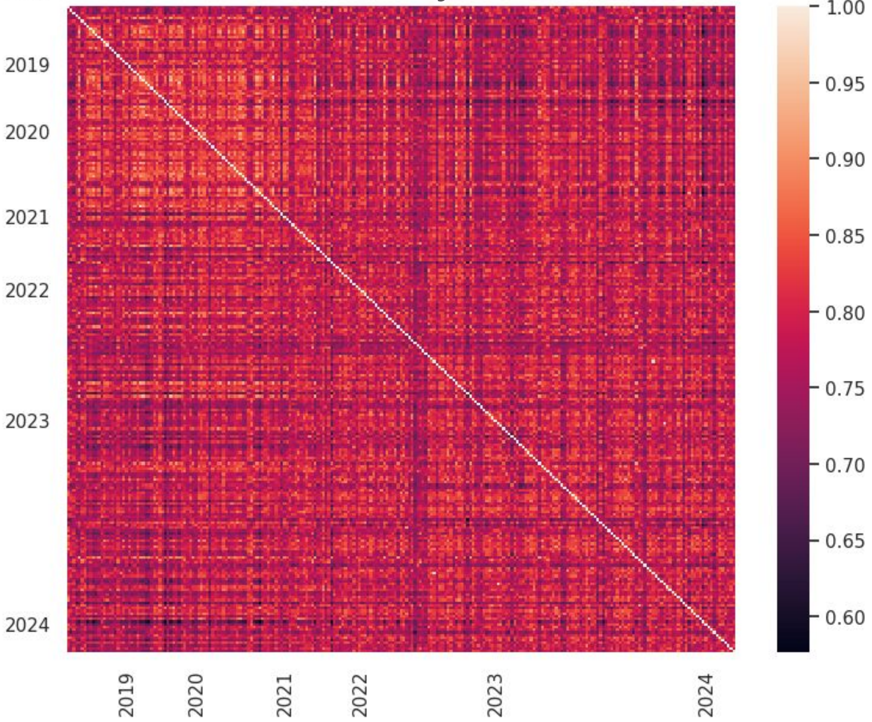


Word-level representations

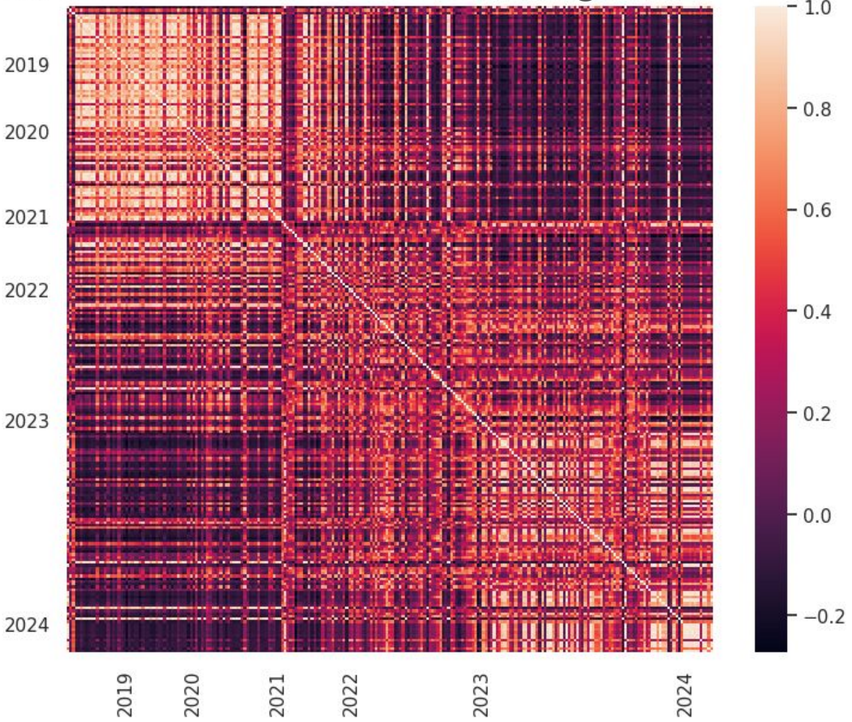
- Does this method affect word-level representations ("word vectors") generated by the language model?
- We generate pairwise similarities for every occurrence of a given word that appears in our test set from models fine-tuned on text and text+embeddings.
- Ideally, fine-tuning LMs on node embeddings will "contextualize" word embeddings as well.

$$cos(\text{“}transformer\text{”}_i, \text{“}transformer\text{”}_j)$$

Text only



Text + node embeddings



Future directions

- Larger citation graph datasets and full-text documents (not just abstracts)
- Improving our understanding of the relationship between attributes, graph structure, and text data.
- Extending our method to ontologies and semantic graphs
    - Unified Medical Language System
    - Wikidata
    - Universal Decompositional Semantics Dataset (White et al. 2019)
        - Predicting semantic type from subsequences

Thanks!

References

1. Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.
2. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135–146.
3. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics.
4. Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
5. Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Proceedings of the International Conference on Learning Representations.
6. White, A. S., Stengel-Eskin, E., Vashishtha, S., Govindarajan, V. S., Reisinger, D. A., Vieira, T., Sakaguchi, K., Zhang, S., Ferraro, F., Rudinger, R., Rawlins, K., & Van Durme, B. (2019). The universal decompositional semantics dataset and decomp toolkit. In Proceedings of the International Conference on Language Resources and Evaluation.
7. Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2019). Attributed network embedding via subspace discovery. Data Mining and Knowledge Discovery, 33, 1953–1980.