

Natural Language Processing and Text Mining: HW#3

By J. H. Wang

May 15, 2023

Programming Exercise #3: Word Embeddings

- Goal: Deriving word embeddings for estimating word similarity and analogy prediction on open datasets
- Input: (to be detailed later)
 - *Word embeddings: fine-tuning pretrained models, or trained on your own*
 - *Text dataset*
- Output: (to be detailed later)
 - Result of word similarity and analogy prediction

Tasks and Data

- Tasks
 - Deriving word embeddings for estimating **word similarity** and **analogy prediction** on open data (as detailed in the following slides)
- Data: open datasets
- You have to submit the result of word similarity and analogy prediction

Example Word Embedding Models

- Word2Vec
- GloVe
- fastText
- ...

Input Data

- Data:
 - **[WordSimilarity-353 Corpus]** by Evgeniy Gabrilovich
 - Available at: <https://gabrilovich.com/resources/data/wordsim353/wordsim353.html>
 - Two sets of word pairs with their similarity scores
 - **[Bigger Analogy Test Set (BATS)]** by Vecto team
 - Available at: <http://vecto.space/projects/BATS/>
 - 99,200 questions in 40 morphological and semantic categories
- Format:
 - WordSim-353: Each set is available in two formats: CSV or Tab-delimited
 - The first two columns: word pairs
 - The third column: mean score for similarity
 - BATS: Word pairs with 40 different relations in 40 files

Tasks in this Homework

- Tasks:
- (40pt) (1) Deriving a word embedding model
 - Either fine-tuning a pretrained model
 - Or training a new model
- (30pt) (2) Using word embedding for word similarity estimation
- (30pt) (3) Using word embedding for analogy prediction
- Optional:
 - (25pt) (4) Compare with other document similarity estimation methods
 - For example, co-occurrence matrix with TF-IDF, SVD, ...
 - (25pt) (5) Apply word embeddings in other tasks
 - For example, classification, NER, ...

Note on the implementation

- You can write your own models or call existing APIs in your program
- The program could be written in any programming language
- Please specify the platform and compilation instructions in your documentation

Output

- Results
 - Word embeddings
 - Word similarity
 - Correlation on WordSim353
 - Analogy prediction
 - Accuracy for BATS categories

Implementation Issues

- You can train your Word2Vec models using packages like genism
- You can also implement your own codes using platforms like PyTorch, Keras, or TensorFlow
- You can use the pretrained word embeddings from the following:
 - GloVe: <https://nlp.stanford.edu/projects/glove/>
 - Pretrained on Twitter, Wikipedia, ...
 - Word2Vec: pretrained on Google News

Notes on Analogy Prediction

- The accuracy of many categories in BATS will be zero
- Please focus on the categories with nonzero accuracies

Homework Submission

- Due: two weeks, **May 29, 2023 (Mon.)**
- For programming exercises, please submit it online to [iSchool+](#)
 - Under the item [Assignments]\[HW#3]
- Please include program source codes and documents
 - specifying your team members and responsible parts in the homework
 - Indicating configuration and installation steps of necessary packages on the specified platform

References

- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín, "**Placing Search in Context: The Concept Revisited**", *ACM Transactions on Information Systems*, 20(1):116-131, January 2002.
- Gladkova, A., Drozd, A., & Matsuoka, S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In Proceedings of the NAACL-HLT SRW (pp. 47–54). San Diego, California, June 12-17, 2016: ACL. <https://www.aclweb.org/anthology/N/N16/N16-2002.pdf>

Thanks for Your Attention!