Developing Vector Space Models

Maximilian Mozes and Bennett Kleinberg
Department of Security and Crime Science
University College London

{maximilian.mozes, bennett.kleinberg}@ucl.ac.uk

Developing vector space models

- Developing vector space models from scratch is not necessary (but a nice exercise to better understand them)

- Use existing libraries (`numpy`, `NLTK`, `gensim`, `glove-python`, `pytorch`, `TensorFlow`)
  - More efficient
  - Easy to use
  - Well documented
  - Regular maintenance and updates

Data

word2vec: ~3 million 300-dimensional embeddings trained on part of the Google News dataset

(see **`https://code.google.com/archive/p/word2vec/`**)

GloVe: ~2.2 million 300-dimensional embeddings trained on Common Crawl (840 billion tokens)

(see **`https://nlp.stanford.edu/projects/glove/`**)

For obtaining the data, see

`https://github.com/maximilianmozes/word_embeddings_workshop_resources/blob/master/README.md`

word2vec in Python

- We use the **gensim** library
- Available at **https://radimrehurek.com/gensim/**

- Demo at

https://github.com/maximilianmozes/word_embeddings_workshop_resources/blob/master/code/train_word2vec.py

GloVe in Python

- We use the **`glove-python`** package
- Available at **`https://github.com/maciejkula/glove-python`**

- Demo at

`https://github.com/maximilianmozes/word_embeddings_workshop_resources/blob/master/code/train_glove.py`

Visualising embeddings

- We use the **`tensorboardX`** package
- Available at **`https://github.com/lanpa/tensorboardX`**

- Demo at

`https://github.com/maximilianmozes/word_embeddings_workshop_resources/blob/master/code/visualize.py`

Other examples

- We have more examples of how to load pre-trained models at

https://github.com/maximilianmozes/word_embeddings_workshop_resources/blob/master/code/examples.py

Other tips and tricks

- Do not implement everything from scratch (use optimised packages)
- Working with *millions* of vectors can be expensive
- Make use of matrix computations (**`numpy`** in Python)
- Examples:
  - Compute distance matrix for all words in vocabulary **ONCE** instead of computing distances over and over again
  - Use existing functions to compute norms and distances
    - **`scipy.spatial.distance.cosine`** for cosine distances
    - **`numpy.linalg.norm`** for vector norms

GloVe in R

- We recommend python for your work with word embeddings
- Reasons: speed + libraries
- However: you can also do it in R

- Note: for all local approaches, memory is an issue (example: 840B, 300d model is a 5.65GB txt file)
- For faster and more voluminous work: server or cloud

Note: tutorial for running RStudio on AWS in the workshop materials and on **`https://danielhammocks.uk/instructions-and-guides/`** (thanks to Daniel Hammocks)

GloVe in R

- Initialising and using GloVe in R
- Step-by-step guide and wrapper functions:
- **https://github.com/maximilianmozes/word_embeddings_workshop_resources/blob/master/code/glove_in_R.pdf** (pdf and HTML notebook in the workshop materials)

1. Looks for pretrained models
2. Initialises GloVe models as DFM matrix in data.table (+ cleans up obsolete files from the R environment)
3. Uses any of the pre-trained GloVe models for neighbour calculation

Thank you for your attention

Any questions?

- Workshop website: **`https://maximilianmozes.github.io/word-embeddings-workshop/`**
- GitHub repo: **`https://github.com/maximilianmozes/word_embeddings_workshop_resources`**

Contact us:

- **Maximilian Mozes**: `maximilian.mozes@ucl.ac.uk; http://mmozes.net`
- **Bennett Kleinberg**: `bennett.kleinberg@ucl.ac.uk; https://bkleinberg.net`
- **Laura Burdick**: `wenlaura@umich.edu; https://laura-burdick.github.io/`