Gensim—Statistical Semantics in Python
Radim Řehůřek, Petr Sojka
NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic
{xrehurek,sojka}@fi.muni.cz

Abstract

Gensim is a pure Python library that fights on two fronts: 1) digital document indexing and similarity search; and 2) fast, memory-efficient, scalable algorithms for Singular Value Decomposition and Latent Dirichlet Allocation. The connection between the two is unsupervised, semantic analysis of plain text in digital collections. Gensim was created for large digital libraries, but its underlying algorithms for large-scale, distributed, online SVD and LDA are like the Swiss Army knife of data analysis—also useful on their own, outside of the domain of Natural Language Processing.

The Digital Library Stuff


Developed for the DML-CZ and EuDML1 (digital math libraries) projects, as a vector-space alternative to the “find similar articles” functionality:

- Python server that runs as a daemon
- Python/Java/C# clients (communication via Pyro/Pyrolite)
- Clients train a semantic model on the server
- Clients issue add/remove/replace documents requests
  - documents converted to “semantic” vectors using the model
- Clients issue queries for the most similar documents

An eye on performance (numbers using my MacBookPro laptop C2D@2.53Ghz, vecLib for BLAS):
- Memory efficient data streaming
  - generators+iterators everywhere
  - train/index on corpora larger than RAM
- Fast semantic model training (see to the right)
- Efficient incremental indexing
  - 1.2k docs per minute, biggest part of it parsing and tokenizing input
  - At the lowest level, queries = matrix multiplications
    - index shards as NumPy&SciPy.sparse matrices mmap’ed from disk

The Math Stuff

Statistical analysis of co-occurrence patterns to identify latent structure. In NLP: word co-occurrence over a corpus of plain text documents (no metadata).

- Training corpus as an implicit word-document matrix
  - sparse, much larger than RAM, streamed over sequentially
- Create a semantic model that captures corpus structure
- Unique Latent Semantic Analysis (truncated SVD) and Latent Dirichlet Allocation implementations:
  - one-pass: each observation seen only once during training
  - incremental: can update model with new observations efficiently
  - distributed: can use Pyro to split the work over several machines/cores
  - constant memory: no $O(#observations)$ required
  - online training, can process infinite data streams!
- Using a trained model, can transform any plain text document to its “semantic” representation (see to the left)

Efficiency: training LSA (truncated SVD) over the full version of English Wikipedia on my MBP laptop with vecLib BLAS:
- 3.5M docs, 100K vocab, 5.4G sparse non-zeroes
- Training: 400 factors in 6.5h
- Transforming: 18k docs/m using the 400-factor LSA model

Similar articles to article

Col. Theme

We appreciate your feedback which determines similarity of articles (e.g. which method is better...). Please include it. It will be helpful for future development.

Back to article

Credits

Gensim is built on top of an excellent open-source Python stack: NumPy, SciPy and Pyro. Our work has been partially supported by the Ministry of Education of Czech Republic within the Center of Basic Research LC536 and by the European Union through its Competitiveness and Innovation Programme (Policy Support Programme, “Open access to scientific information”, Grant Agreement No. 250503). Many thanks to gensim contributors and testers. Gensim is licensed under LPGL—get it from PyPI or clone from github (just google it).

References
