

# 91258 / B0385 Natural Language Processing

**Lesson 13. Hands on Word Embeddings** 

Alberto Barrón-Cedeño a.barron@unibo.it

DIT, LM SpecTra

# Table of Contents

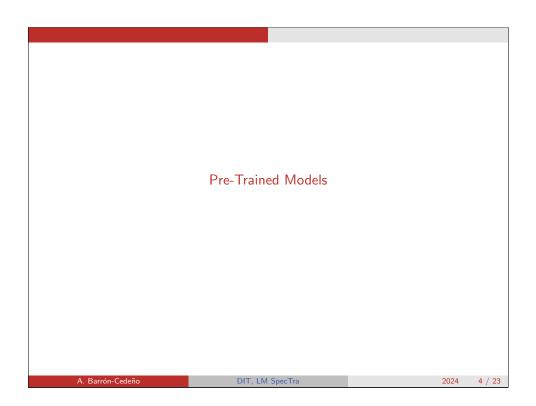
- 1. Pre-Trained Models
- 2. Gensim
- 3. Model Construction
- 4. GloVe
- 5. fastText

Chapter 6 of Lane et al. (2019)

Previously

• Skip-gram

• CBOW



# Some Pre-Trained Models

Model	Provider	Description	
word2vec	Google	300D from English Google News articles <sup>1</sup>	
fastText	Facebook	157 languages from Wikipedia and Common Crawl <sup>2</sup>	
word2vec/GloVe	CNR	Italian embeddings from the Wikipedia	
word2vec	UCampania	Italian embeddings <sup>3</sup>	

There are many pre-trained models and diverse libraries to handle them.

Just query your favorite search engine

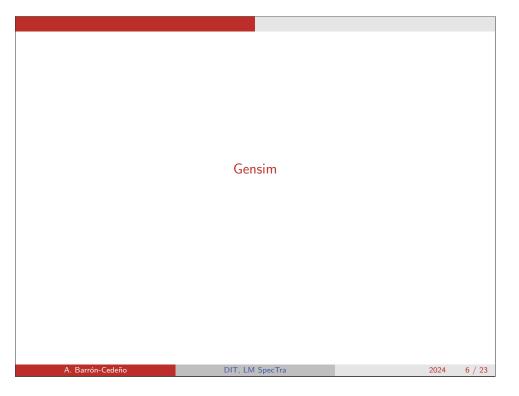
1https://drive.google.com/file/d/OB7XkCwpI5KDYN1NUTTlSS21pQmM

3https://mlunicampania.gitlab.io/italian-word2vec/

A. Barrón-Cedeño DIT, LM SpecTra

### Gensim

- Scalable, open source, and efficient Python library
- It includes many resources, including word2vec, doc2vec, FastText, LDA, and more
- All information, including (very nice) documentation at https://radimrehurek.com/gensim/



### Gensim

Most similar items

word\_vectors.most\_similar()

Among the most interesting parameters:

positive list of vectors to be added together before looking for the neighbours

negative subtraction (or exclusion) of the elements

topn number of elements to retrieve

<sup>&</sup>lt;sup>2</sup>https://fasttext.cc

### Gensim

Least similar items (closed set)

word\_vectors.doesnt\_match()

It returns the element from the input list with the lowest similarity with respect to the rest

Let us see

### Gensim

Getting the Vectors

Gensim (and other libraries) have implemented these interfaces to perform some *standard* operations

To go beyond, one needs to get access to the actual vectors

word\_vectors[word]

■ Let us see

# Gensim

More operations

# Adding and Subtracting

We can use most\_similar() again, this time with the negative parameter

Let us see

# Computing similarities

word\_vectors.similarity()

Let us see

A. Barrón-Cedeño

12 / 23

Model Construction

DIT, LM SpecTra

A. Barrón-Cedeño DIT, LM SpecTra 2024 11 / 23

# Model Construction

#### Considerations

- If you are working in other language than English, Google's provided word2vec is not an option (FastText might be)
- Google's word2vec is built on news; fastText has versions built on the Wikipedia and on common crawl... analysing scientific papers or literature?
   Probably not
- You want to work on COVID-19 or any other recent topic?
   Many relevant terms wont appear

#### **Alternatives**

- Opting for some of the previous representations
- Build your own model

A. Barrón-Cedeño

DIT, LM SpecTra

2024 13 / 23

### Model Construction

#### Training

Training a word2vec model with gensim

Tutorial: https://rare-technologies.com/word2vec-tutorial/



#### Considerations

- Training on relatively large corpora might take some time (Brown is small and took me a bit less than 1 minute on a 2.5GHz Quad-Core i7, 16GB RAM)
- Large corpora (e.g., the Wikipedia) can require a significant amount of time/memory

# Model Construction

Pre-Processing

### Typical pre-processing pipeline

- Tokenisation
- Lowercasing (optional)
- Sentence splitting

Input Embedded list of tokenised sentences

 $[[w_{0,0} \ w_{0,1} \ w_{0,2} \dots w_{0,k}], [w_{1,0} \ w_{1,1} \ w_{1,2} \dots w_{1,l}], \dots [w_{x,0} \ w_{x,1} \dots w_{x,m}]]$ 

A. Barrón-Cedeño

DIT, LM SpecTra

024 14 / 23

### Model Construction

Trimming and Saving

Reminder We do not care about the output

model.init\_sims(replace=True)

- Freezes the model
- Stores the hidden-layer weights
- Discards the output-layer weights

not necessary since gensim 4.0

Now we simply have to save the model with model.save()

Let us see

A. Barrón-Cedeño

DIT IMC T

GloVe

# GloVe

GloVe vs word2vec

### RaRe Technologies comparison<sup>5</sup>

Settings: 600 dims • context window of 10 • 1.9B words of *en* Wikipedia.

	acc (word	wallclock	peak RAM
Algorithm	analogy)*	time	(MB)
I/O only	_	3m	25
GloVe, 10 epochs, Ir 0.05	67.1	4h12m	9,414
GloVe, 100 epochs, lr 0.05	67.3	18h39m	9,452
word2vec, hierarchical skip-	57.4	3h10m	266
gram, 1 epoch			
word2vec, negative sampling	68.3	8h38m	628
(10 samples), 1 epoch			
word2vec, Google 300d	55.3	_	_

<sup>\*</sup> a\_is to b as c is to ?

### DIT, LM SpecTra

# GloVe

# Global Vectors (Pennington et al., 2014)<sup>4</sup>

- It uses a global word-word co-occurrence matrix
- Learning objective: word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence
- It produces similar matrices to word2vec
- It converges, even with smaller corpora
- It is more accurate with the same amount of data

4https://nlp.stanford.edu/projects/glove/

A. Barrón-Cedeño DIT, LM SpecTra

fastText

A. Barrón-Cedeño DIT, LM SpecTra

Frare-technologies.com/making-sense-of-Word2vec/#glove\_vs\_word2vec A. Barrón-Cedeño

# fastText

Predicts the surrounding character [2, 3]-grams rather than the surrounding words (Bojanowski et al., 2017)<sup>6</sup>

- Pre-trained models available in 250+ languages
- Built on Wikipedia editions (variable quality)
- Built on common crawl

Models available at https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md

Let us see

6https://github.com/facebookresearch/fastText

A. Barrón-Cedeño

DIT, LM SpecTra

2024 21 / 23

### References

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Lane, H., C. Howard, and H. Hapkem 2019. *Natural Language Processing in Action*. Shelter Island, NY: Manning Publication Co.

Pennington, J., R. Socherm, and C. Manning 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, Pp. 1532–1543.

A. Barrón-Cedeño DIT, LM SpecTra 2024 23 / 23

# Some Remarks

LSA is a better (faster) option for long documents e.g., for clustering

Online learning An existing model can be *adapted* (but new words cannot be added)

doc2vec possible representation based on linear combinations of word2vec

DIT I M Specific