

# 91258 Natural Language Processing

## Lesson 14. From word back to document representations

Alberto Barrón-Cedeño

Alma Mater Studiorum-Università di Bologna  
a.barron@unibo.it @albarron\_

14/11/2023



## Previously

- Training and loading (existing) embeddings

## Table of Contents

Doc2vec

End of Chapter 6 of Lane et al. (2019); after skipping visualisation

**Doc2vec**

## Doc2vec

**Objective** Computing a vectorial representation for a document

Same idea as with word2vec: a NN to predict words

### Input

- ▶  $k$  context words (optional)
- ▶ A unique ID of the sentence/paragraph/document

### Output

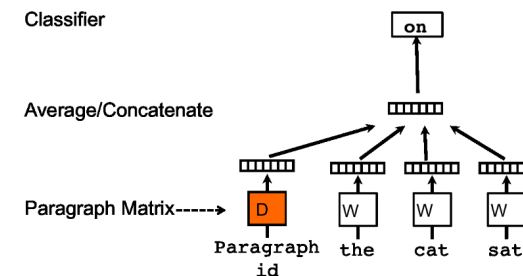
- ▶ 1 target word
- ▶ The paragraph vector is unique among all documents
- ▶ The word vectors are shared among all documents
- ▶ The document vector is computed **on the fly**

(Le and Mikolov, 2014); (Lane et al., 2019, p. 215)

## Doc2vec

Distributed Memory Model of Paragraph Vectors (PV-DM)

Derived from CBOW

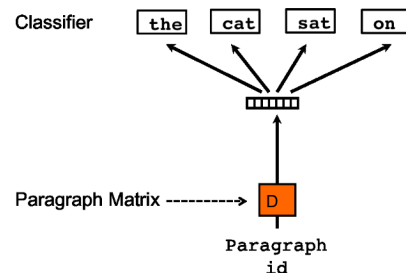


- ▶ Each column in the paragraph matrix is a vector representing one paragraph
- ▶ We can average or concatenate the word and paragraph vectors


## Doc2vec

Distributed Bag of Words version of Paragraph Vector (PV-DBOW)

Similar to skip-gram



- ▶ Iteration: a text window and a random word from the text window are sampled, forming a classification task given the paragraph vector.
- ▶ No word vectors: faster + lower memory requirements

 Let us see

## References

- Lane, H., C. Howard, and H. Hapkem  
2019. *Natural Language Processing in Action*. Shelter Island, NY: Manning Publication Co.
- Le, Q. V. and T. Mikolov  
2014. Distributed representations of sentences and documents.