NATURAL LANGUAGE PROCESSING WITH DEEP LEARNING

Dr.Minaei, IUST, Fall 2020

Presenting by Sara Rajaei

Most materials of this mini-course are provided by "Natural Language Processing with Deep Learning" course, Stanford, Winter 2020 and "Natural Language Processing" course, Mohammad Taher Pilehvar, IUST, Fall 2019

IN THIS MINI-COURSE

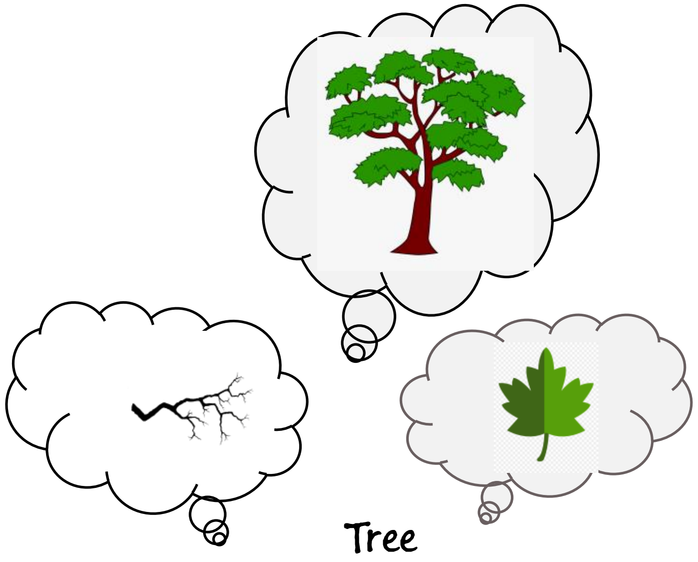
- A big picture understanding of human languages
- An introduction to different tasks in NLP
- Advanced methods used in NLP: Recurrent Neural Networks, LSTMs, Attention mechanism, Transformers, etc.

PRESUMPTION

- You should be familiar with Python and Numpy. If you have a lot of programming experience but in a different language(e.g. C, C++, Matlab), you'll have an easy way to learn python.
- You should be comfortable with taking (multivariable) derivatives and understanding matrix/vector notation and operations.

HUMAN LANGUAGE

- Word as a symbol



HUMAN LANGUAGE

- Word as a symbol
- Why understanding of human languages are so hard for machine?
  - Ambiguity!

    A word's meaning depends on its context.
  - External Knowledge

Bank

DIFFERENT TASKS IN NLP

- Downstream tasks are problems designed by experts to evaluate a model on different linguistic features
- Consider an NLP model as a black box(we'll explain a model in the next session)
- We want to train a model that can answer to the problems(Downstream tasks)

DIFFERENT TASKS IN NLP

- Part-of-speech tagging
- Dependency parsing
- Semantic role labeling
- Sentiment analysis / opinion mining
- Word sense disambiguation/induction
- Named-entity recognition/classification
- Co-reference resolution
- Summarizing
- Textual entailment
- Question answering
- Machine translation
- ❖ Language Model
- ❖ Masked Language Model

DIFFERENT TASKS IN NLP

- Part-of-speech tagging
- Dependency parsing
- Word sense disambiguation/inductio
- Machine translation
- Semantic role labeling
- Sentiment analysis / opinion mining
- Named-entity recognition/classification
- Co-reference resolution
- Textual entailment
- Question answering

❖ Language Model

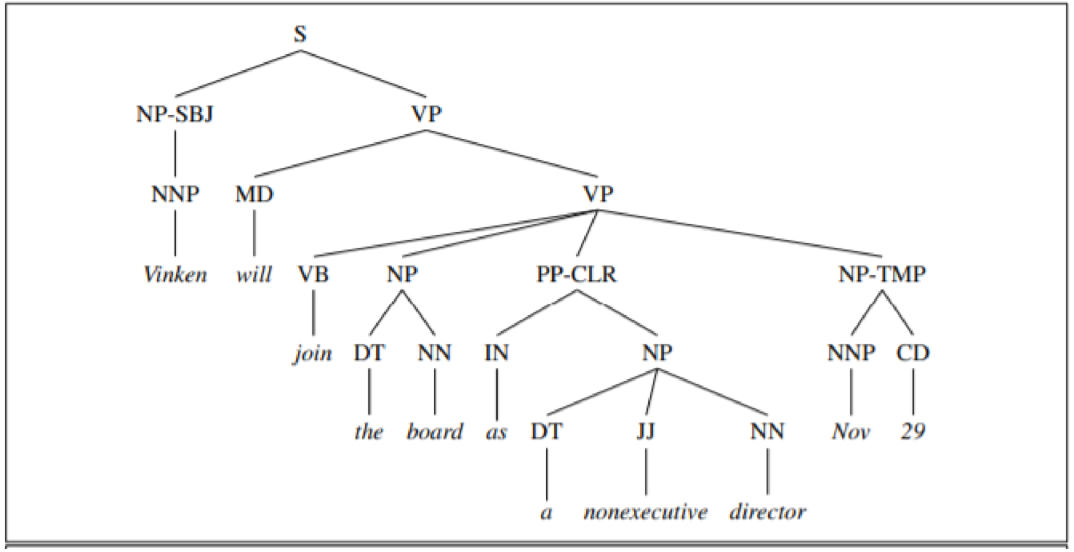
❖ Masked Language Model

```python
sen = sp("I like to play football. I hated it in my childhood though")
tmp = ''
for j in range(len(sen)):
        tmp = tmp +' '+ sen[j].pos_
print(tmp)
```

```
PRON VERB PART VERB NOUN PUNCT PRON VERB PRON ADP DET NOUN ADV
```

DIFFERENT TASKS IN NLP

- Part-of-speech tagging
- Dependency parsing
- Word sense disambiguation/induction
- Machine translation
- Semantic role labeling
- Sentiment analysis / opinion mining
- Named-entity recognition/classification
- Co-reference resolution
- Textual entailment
- Question answering
- Language Model
- Masked Language Model



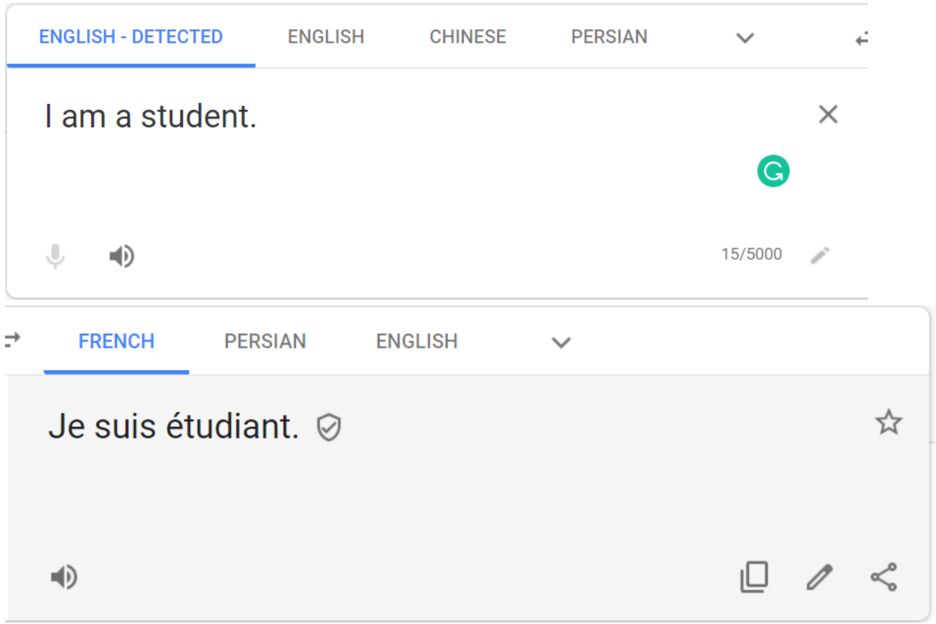
DIFFERENT TASKS IN NLP

- Part-of-speech tagging
- Dependency parsing
- Word sense disambiguation
- Machine translation
- Semantic role labeling
- Sentiment analysis / opinion mining
- Named-entity recognition/classification
- Co-reference resolution
- Textual entailment
- Question answering
- Language Model
- Masked Language Model

The car hit the pole while it was moving

DIFFERENT TASKS IN NLP

- Part-of-speech tagging
- Dependency parsing
- Word sense disambiguation
- Machine translation
- Semantic role labeling
- Sentiment analysis / opinion mining
- Named-entity recognition/classification
- Co-reference resolution
- Textual entailment
- Question answering
- ❖ Language Model
- ❖ Masked Language Model



DIFFERENT TASKS IN NLP

- Part-of-speech tagging
- Dependency parsing
- Word sense disambiguation
- Machine translation
- Semantic role labeling
- Sentiment analysis / opinion mining
- Named-entity recognition/classification
- Co-reference resolution
- Textual entailment
- Question answering
- Language Model
- Masked Language Model

Allen NLP demo

REPRESENTING WORDS AS VECTORS

- Neural networks and deep models accept numbers(tensors) as inputs, So we need to convert raw texts to vectors.
- Text Segmentation
  - Word-level

    Segment text into words, and transform each word into a vector.
  - Character-level

    Segment text into characters, and transform each character into a vector.
  - SubWord-level

    Segment text into subwords, and transform each subword into a vector.

TOKENIZATION

Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords.

The biggest box

- Words
  - In English, you can consider space as a delimiter.

"the", "biggest", "box"

- Characters

T-h-e-b-i-g-g-e-s-t-b-o-x

- Subwords

"the", "big", "est", "box"

ONE-HOT ENCODING

Representing each token with an unique index, and then turning this index to a vector of size N.

- Advantages
  - Simple
- Disadvantages
  - Inefficient in using memory.
  - Do not reflect the words' meanings
  - …

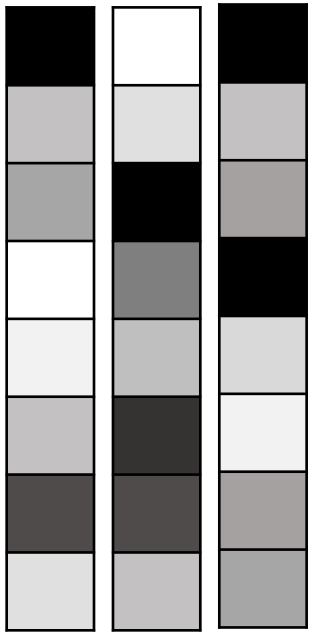
The biggest box

	1	2	3
The	1	0	0
biggest	0	1	0
box	0	0	1

WORD EMBEDDING

Representing each token with a low dimensional vector which has useful traits

- Embed more information in lower dimensions
- Can jointly be learned with the target task. In this approach, word embeddings are initialized randomly and learned while neural network's weights are being set.
- Or you can use pre-trained word embeddings as the initial weights of word embeddings.

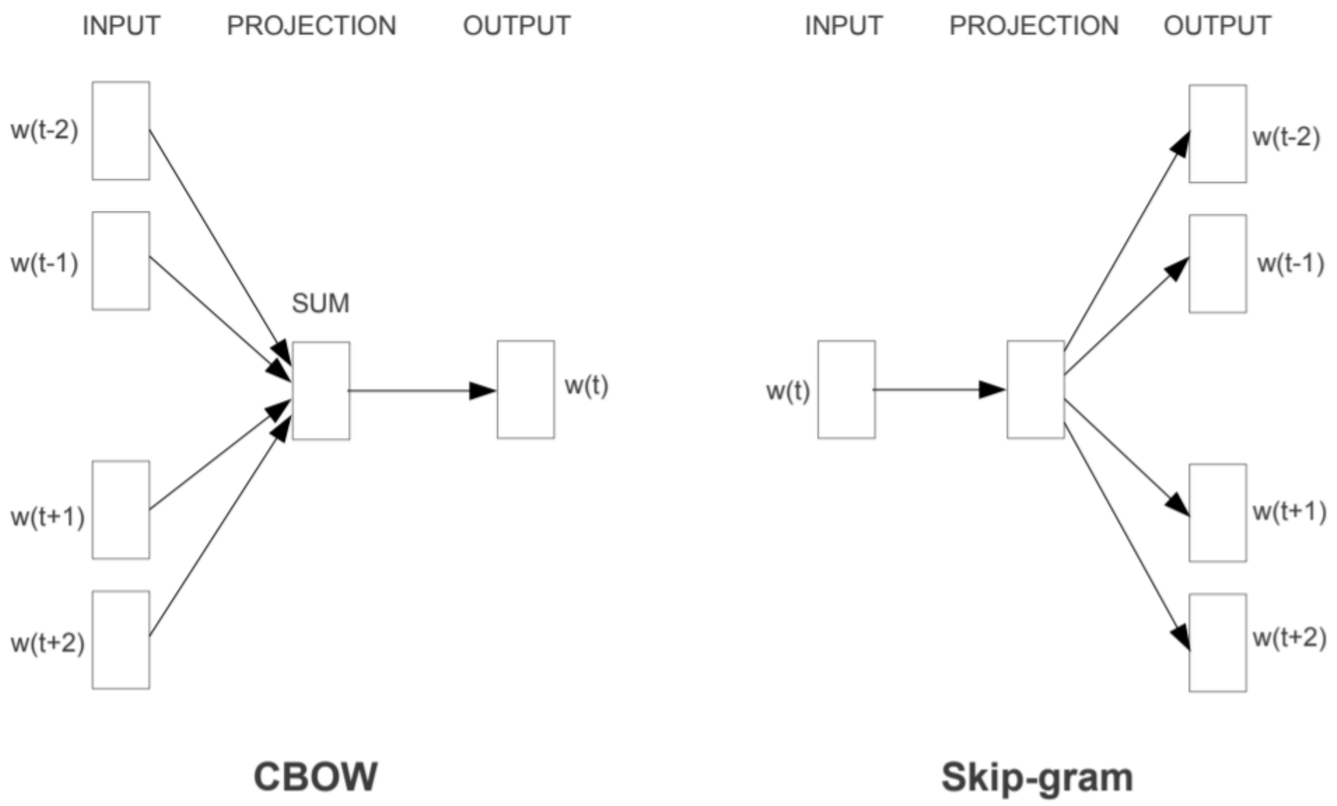


PRE-TRAINED WORD EMBEDDING

Pre-trained word embeddings are trained on huge datasets with a specific target task and then are used in other tasks.

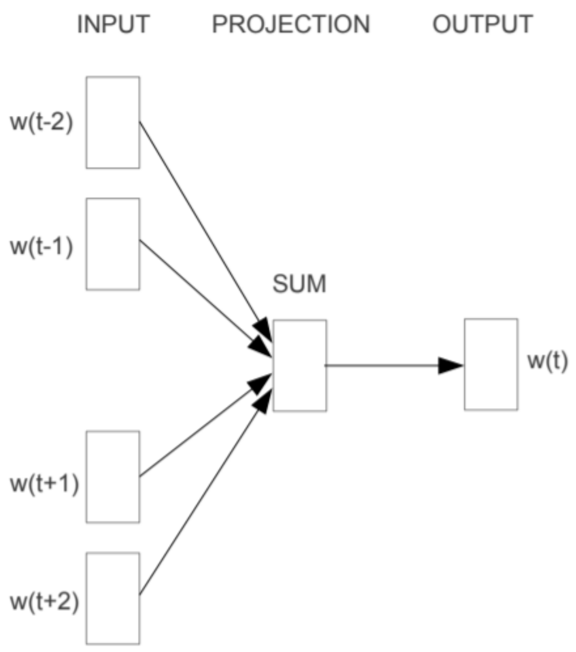
- A form of transfer learning
- Why they are useful?

PRE-TRAINED WORD EMBEDDING – WORD2VEC



WORD2VEC – NEGATIVE SAMPLING

- Instead of updating hole neural network's weights, we modify a small percentage of the weights
- In other words, instead of updating whole words in vocabulary, we update a small subset of words.
- These chosen words are called negative words

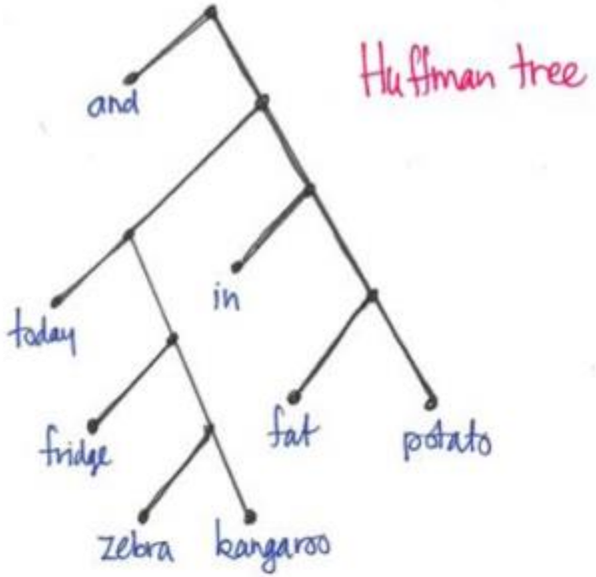


CBOW

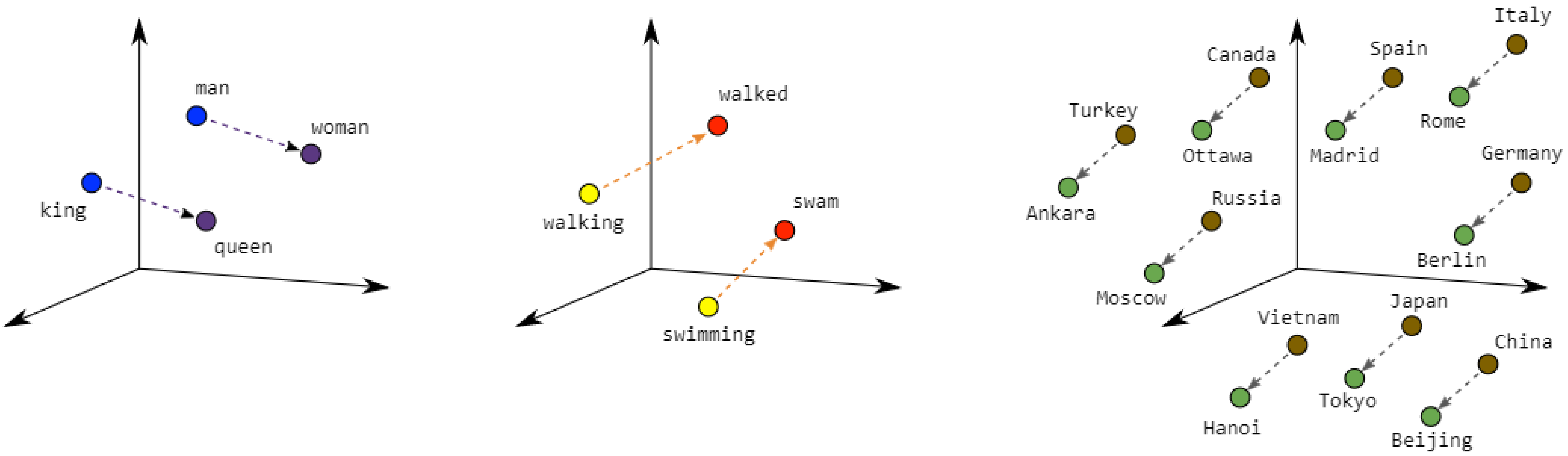
WORD2VEC – HIERARCHICAL SOFTMAX

word	count
fat	3
fridge	2
zebra	1
potato	3
and	14
in	7
today	4
kangaroo	2

→



WORD2VEC FEATURES

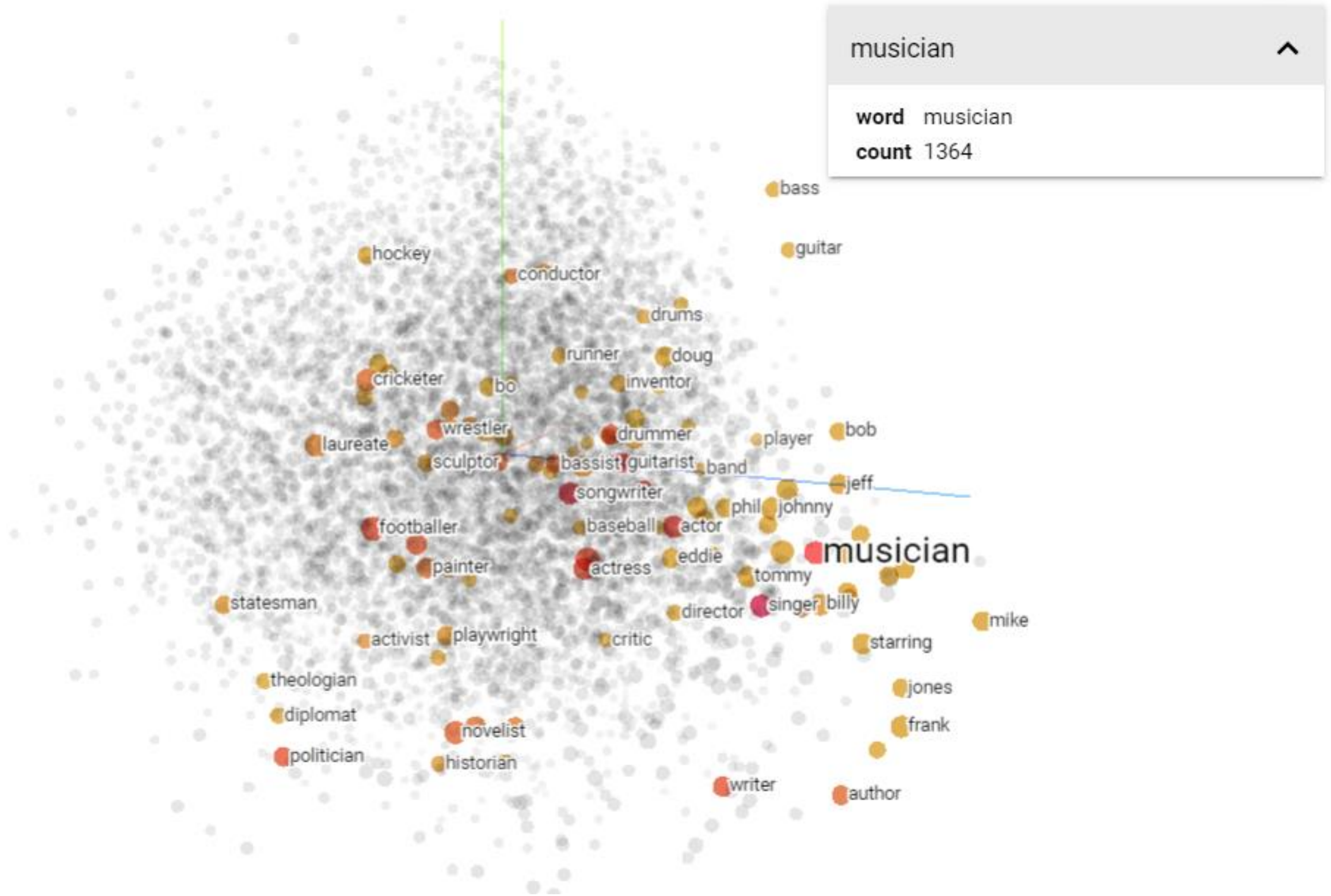


Male-Female

Verb Tense

Country-Capital

WORD2VEC FEATURES



PRE-TRAINED WORD EMBEDDING - GLOVE

- Using local information (like Word2Vec) and global information (word co-occurrence)

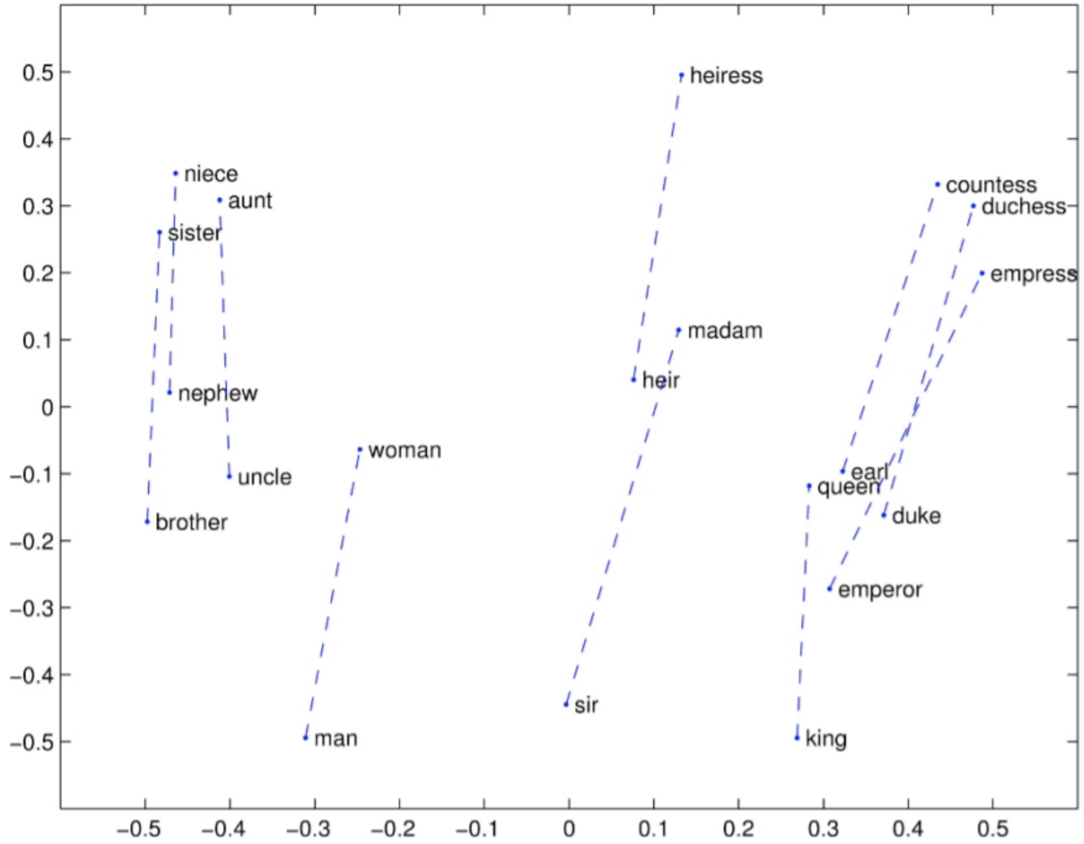
The biggest box

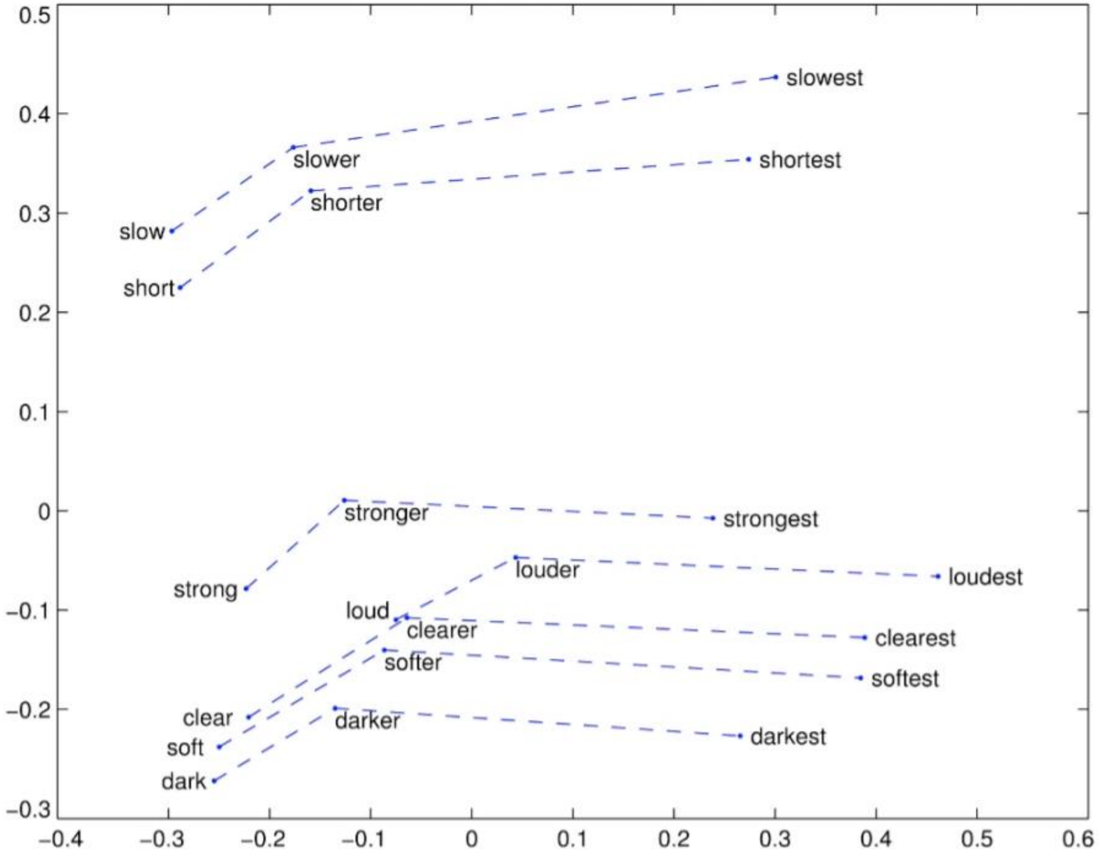
- Is "The" a special context for "box" or it is just a stopword?

  Word2Vec cannot answer to this question!

- Glove uses a co-occurrence matrix representing how often word *i* appears in context of word *j*
- Using a cost function which help us to prevent learning only from extremely common word pairs. (stopwords)

GLOVE


heiress
niece
aunt
sister
countess
duchess
empress
madam
heir
nephew
woman
uncle
earl
queen
duke
brother
emperor
sir
man
king


slowest
slower
shortest
shorter
slow
short
stronger
strongest
louder
loudest
strong
loud
clearer
clearest
softer
softest
clear
darker
darkest
soft
dark

REFERENCES AND FURTHER RESOURCES

Websites:

1. http://nlpprogress.com/
2. https://demo.allennlp.org/reading-comprehension
3. https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/
4. https://towardsdatascience.com/word-embeddings-for-nlp-5b72991e01d4
5. https://nlp.stanford.edu/projects/glove/
6. https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010

Papers and Books:

1. Efficient Estimation of Word Representations in Vector Space
2. GloVe: Global Vectors for Word Representation
3. Embedding in Natural Language Processing
4. From Frequency to Meaning: Vector Space Models of Semantics