MSiA414 SEC01
Text Analytics
Lab 3 - Word2Vec and BERT

Timo Wang

Northwestern University

October 8 2021

Overview

- Discussions on Word2Vec and BERT
- Using BERT
- Quiz

Discussions on Word2Vec and BERT

Question1

How is a Skip-gram model trained? What is its training objective? How is BERT trained? What are its training objectives?

Discussions on Word2Vec and BERT

Question 1

How is the Skip-gram model trained? What is its training objective? How is BERT trained? What are its training objectives?

Notes 1

- The training objective of the Skip-gram model is to maximize the average log probability of the context words.
- The training of BERT has two stages: pretraining and fine-tuning. The pretraining stage involves guessing the correct masked out word and predicting if the second sentence follows the first one.

Discussions on Word2Vec and BERT

Question 2

What makes BERT different from Word2Vec models?

Discussions on Word2Vec and BERT

Question 2

What are some differences between BERT and Word2Vec?

Notes 2

- Word2Vec vector provides a vector for each token/word that encodes the meaning of that token/word.
- BERT provides contextual word representations that encode different meanings under different context.

Discussions on Word2Vec and BERT

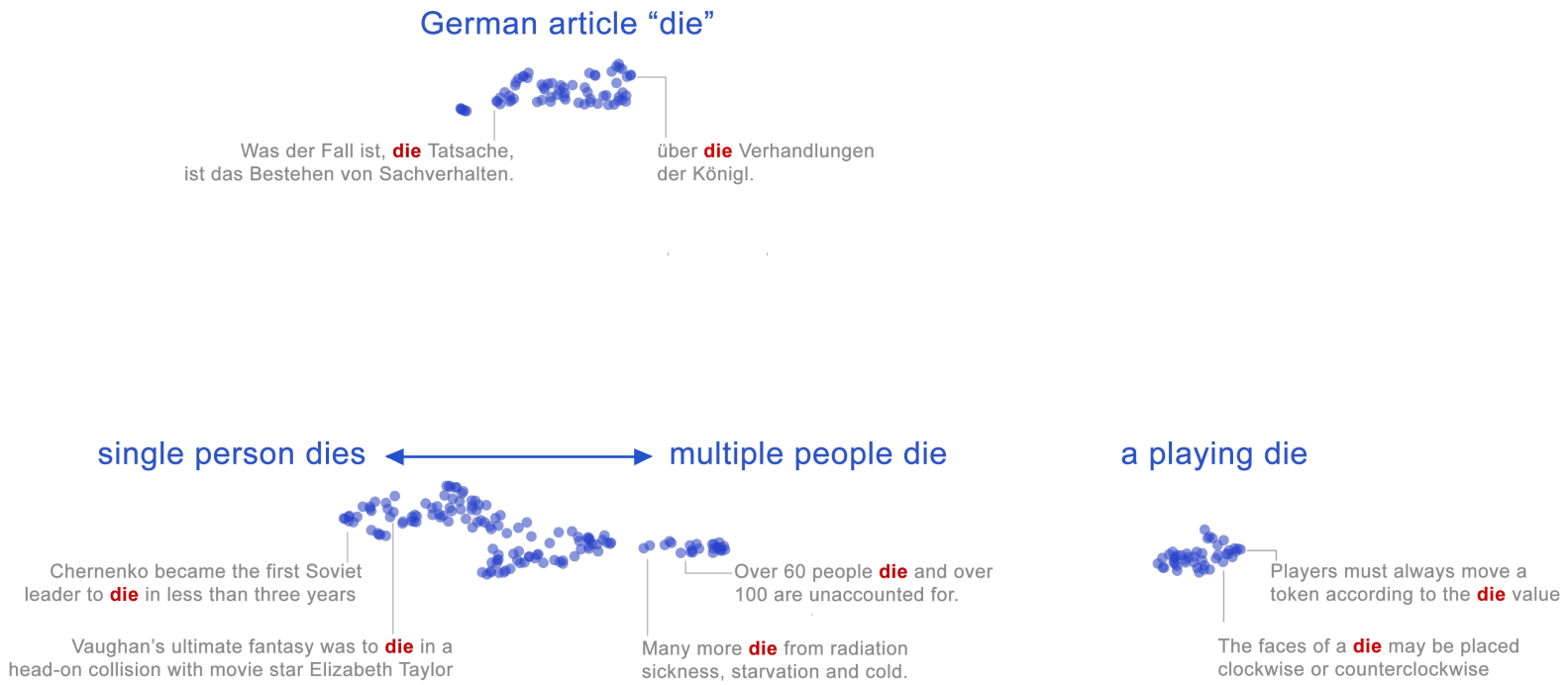

Figure: Embeddings for the word “die” in different contexts.[1]

[1] “Visualizing and Measuring the Geometry of BERT”. Coenen et al.

Discussions on Word2Vec and BERT

Question 3

What are the advantages of pre-training for BERT?

Discussions on Word2Vec and BERT

Question 3

What are the advantages of pre-training for BERT?

Notes

Pretraining BERT provides us with a generalized **language model** that can be used later for a variety of natural language processing tasks. The amount of training data required for those down streaming tasks does not need to be huge as the pre-trained model contains significant information on the language itself.

Discussions on Word2Vec and BERT

Question 4

What are some down streaming tasks we can perform by fine-tuning BERT?

Discussions on Word2Vec and BERT

Question 4

What are some down streaming tasks we can perform by fine-tuning BERT?

Notes

- Single-sentence classification tasks
- Single-sentence tagging tasks
- Question answering tasks

Using Bert

Hugging Face's Transformers

Install

pip install transformers

Example usage

```
> from transformers import BertTokenizer, BertModel

> tokenizer = BertTokenizer.from_pretrained("bert-base-
    uncased")
> model = BertModel.from_pretrained("bert-base-uncased")
> inputs = tokenizer("Hello world!", return_tensors="pt
    ")
> outputs = model(**inputs)
```

Using Bert

Other options/resources

- An alternative – bert-as-service: https://github.com/hanxiao/bert-as-service
- A potentially outdated but still good tutorial on Hugging Face's Transformers: https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/

Quiz

Task 1

Which of the following can be somehow realized with BERT but **not** with Word2Vec?

A Computing vectorized sentence representations

B Computing vectorized word representations

C Computing contextualized word representations

D All of the above

Quiz

Task 2

Which of the following is part of the pretraining stage for BERT?

A Single-sentence tagging

B Masked-out word prediction

C Next sentence prediction

D A and B

E B and C

F A, B and C