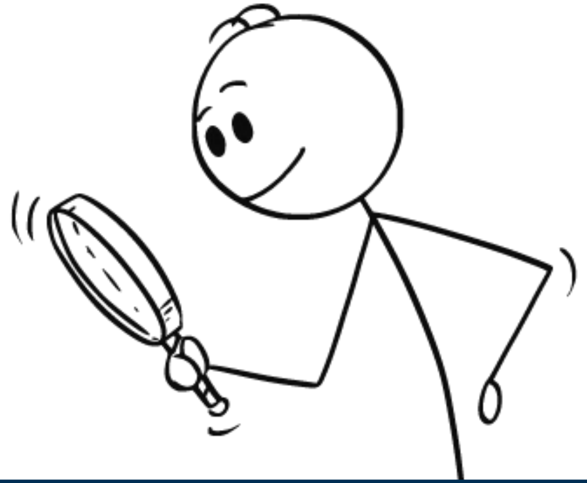


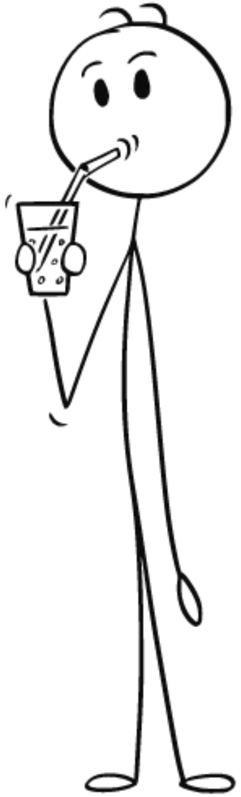
CSE 4/535
Information Retrieval

Sayantan Pal
PhD Student, Department of CSE
338Z Davis Hall

University at Buffalo
Department of CSE

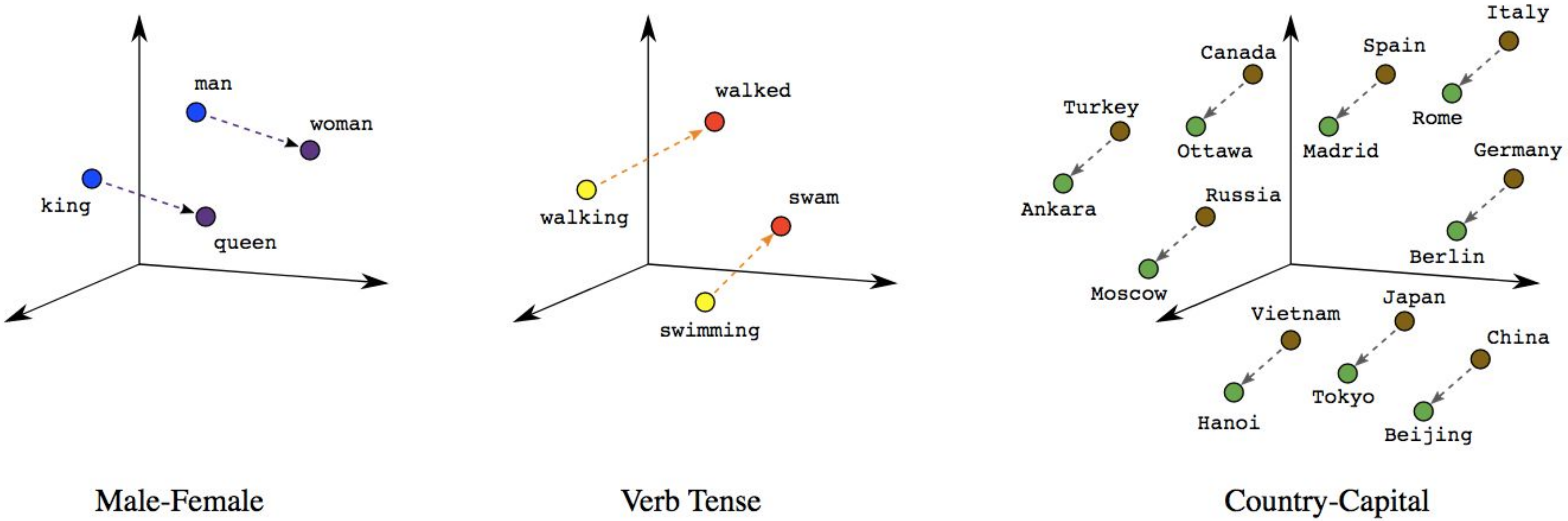
Before we start

1. Midterm 1 Grades released (will be curved).
2. Project 1 will be released by 9th October.
3. Good performance (Both Projects and Midterm)
4. Project 2 will be will be available on Website by 9th October.
5. No office hours this Friday (will be shifted to next week and posted on Piazza)
6. Today's lecture - Midterm discussion & Exciting things (might be helpful for final project)



Midterm Discussion - No Recording

Embeddings in Information Retrieval & Transformers



Male-Female

Verb Tense

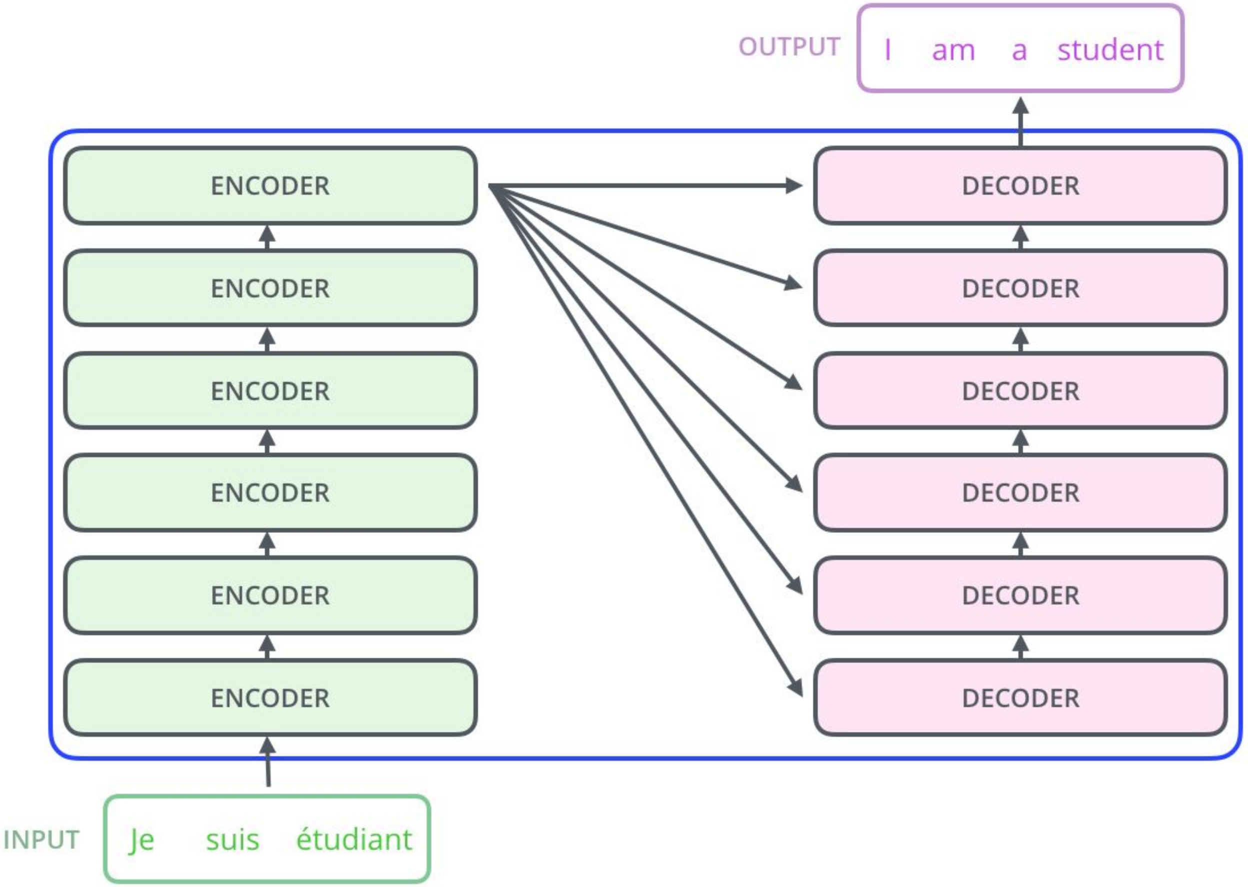
Country-Capital

Source: *https://towardsdatascience.com/a-guide-to-word-embeddings-8a23817ab60f*

Demo Time

Link:https://www.youtube.com/watch?v=znNe4pMCsD4

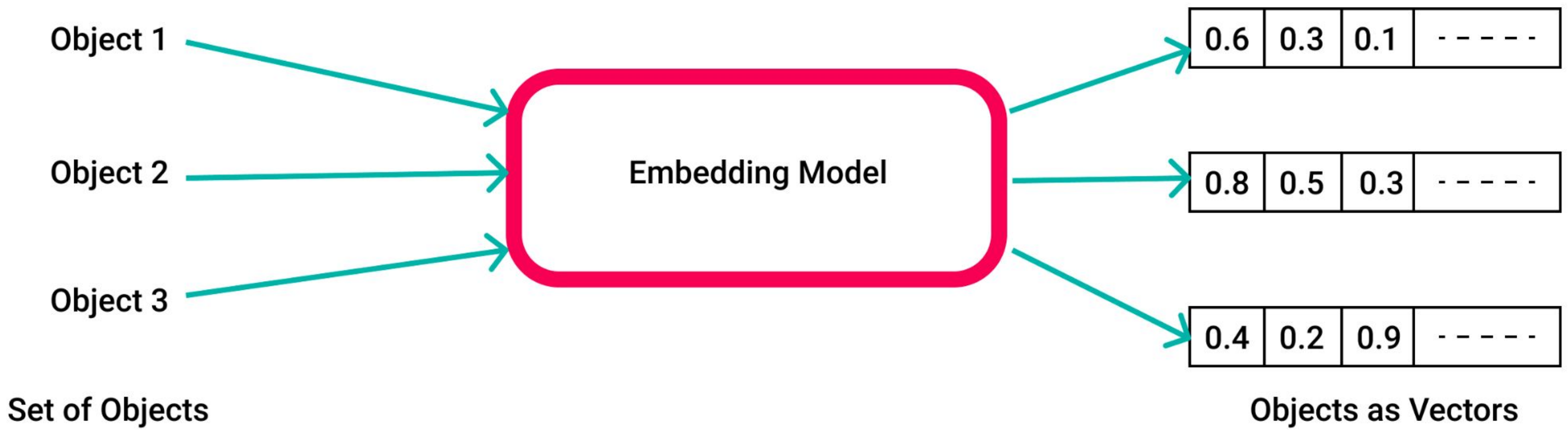
Introducing Transformers



Does this go in?

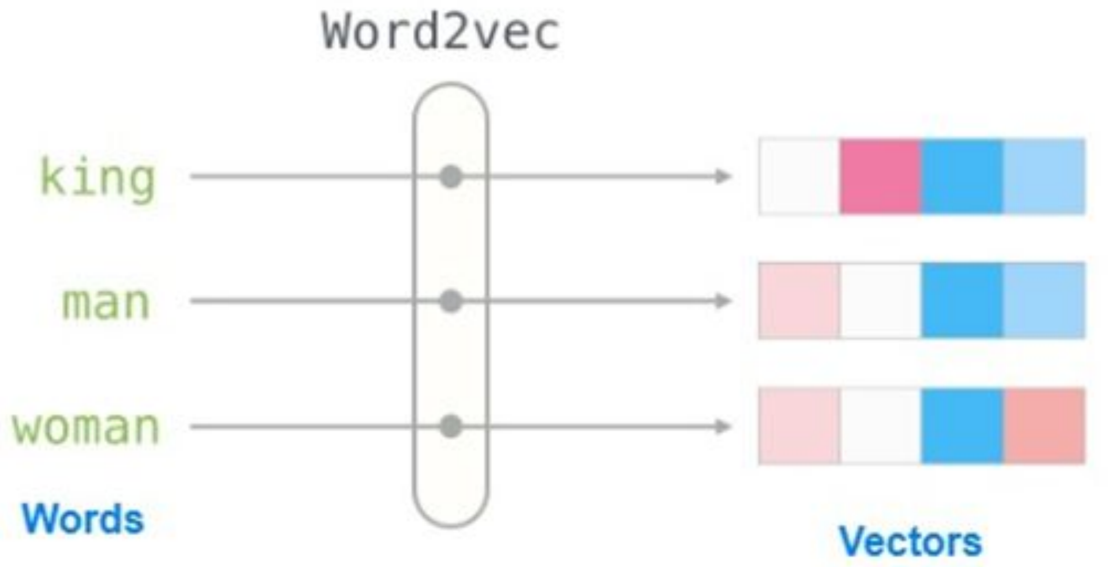
Introduction to Embeddings

1. **Definition**: Representing words, sentences, or documents as vectors in a continuous vector space.
2. **Purpose**: Facilitates measuring semantic similarities and performing mathematical operations on words or sentences.



Well known Text to Vectors algorithms

- **Word2Vec**
- **GloVe** (Global Vectors for Word Representations)
- **FastText**
- **ELMo**
- **BERT**

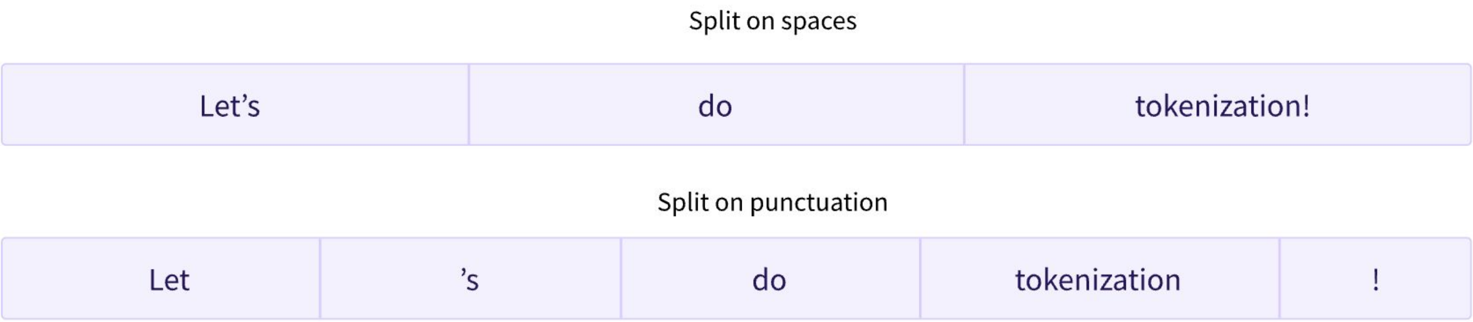


Tokenizers

- Tokenizers are one of the core components of the NLP pipeline. **They serve one purpose**: to translate text into data that can be processed by the model
- Models can only process numbers, so tokenizers need to convert our text inputs to numerical data.
- The goal is to find the most meaningful representation — that is, the one that makes the most sense to the model — and, if possible, the smallest representation

No. of Tokens … Is it small or large?

- A vocabulary is defined by the total number of independent tokens that we have in our corpus
- Each word gets assigned an ID, starting from 0 and going up to the size of the vocabulary. The model uses these IDs to identify each word.



Split on spaces

Let’s | do | tokenization!

Split on punctuation

Let | ’s | do | tokenization | !

No. of Tokens … Is it small or large?

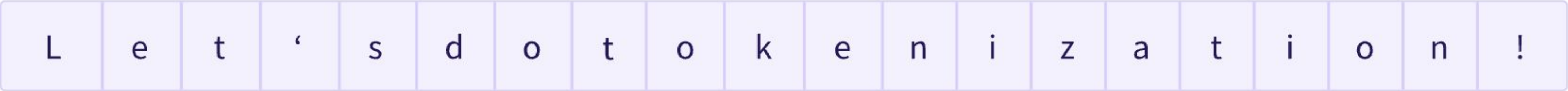
- A vocabulary is defined by the total number of independent tokens that we have in our corpus
- Each word gets assigned an ID, starting from 0 and going up to the size of the vocabulary. The model uses these IDs to identify each word.
- So, if there are 500,000 words in Modern English Language, will we have 500,000 IDs??

No. of Tokens … Is it small or large?

- A vocabulary is defined by the total number of independent tokens that we have in our corpus
- Each word gets assigned an ID, starting from 0 and going up to the size of the vocabulary. The model uses these IDs to identify each word.
- So, if there are 500,000 words in Modern English Language, will we have 500,000 IDs??
- How difficult will it be to predict a token by the model?
- What will we do when we face a word that is not in the vocabulary?

No. of Tokens … Is it small or large?

- What about characters?
- The vocabulary is much smaller.
- There are much fewer out-of-vocabulary (unknown) tokens, since every word can be built from characters.



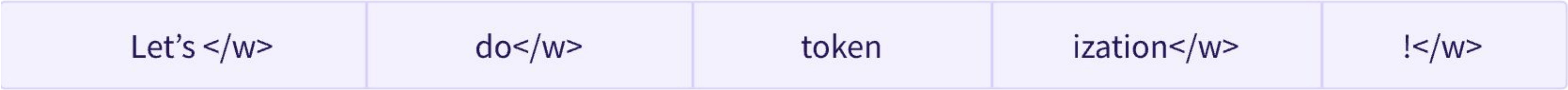
L	e	t	‘	s	d	o	t	o	k	e	n	i	z	a	t	i	o	n	!

No. of Tokens … Is it small or large?

- What about characters?
- The vocabulary is much smaller.
- There are much fewer out-of-vocabulary (unknown) tokens, since every word can be built from characters.
- It’s less meaningful: each character doesn’t mean a lot on its own
- So… What now?

Subword Tokenization

- Subword tokenization algorithms rely on the principle that frequently used words should not be split into smaller subwords, but rare words should be decomposed into meaningful subwords.
- For instance, “annoyingly” might be considered a rare word and could be decomposed into “annoying” and “ly”. These are both likely to appear more frequently as standalone subwords, while at the same time the meaning of “annoyingly” is kept by the composite meaning of “annoying” and “ly”.



Let’s </w>	do</w>	token	ization</w>	!</w>

Read More…

- WordPiece is a subword tokenization method that breaks words into smaller units, allowing models to handle rare and out-of-vocabulary words effectively. It starts by initializing a vocabulary with individual characters present in the dataset and gradually merges to form subwords. WordPiece is notable for its use in prominent models like BERT, which leverages it to produce embeddings that can generalize across various linguistic contexts by representing words as combinations of subword units.
    - Link: https://huggingface.co/learn/nlp-course/chapter6/6?fw=pt
- Byte-Pair Encoding (BPE) was initially developed as an algorithm to compress texts, and then used by OpenAI for tokenization when pretraining the GPT model. It’s used by a lot of Transformer models, including GPT, GPT-2, RoBERTa, BART, and DeBERTa.
    - Link: https://huggingface.co/learn/nlp-course/chapter6/5?fw=pt

To be continued …

- In the next episode
  - What happens after we have the subwords
  - What are encodings?
  - How is encoding different from embedding?
  - What is decoding and post-processing?

References

1. https://huggingface.co/learn/nlp-course/chapter2/4?fw=pt
2. https://jalammar.github.io/illustrated-transformer/
3. Colab Notebook by Hugging Face: https://colab.research.google.com/github/huggingface/notebooks/blob/master/course/en/chapter2/section4_pt.ipynb