Information Retrieval Systems

Lecture 6: Semantic Representations

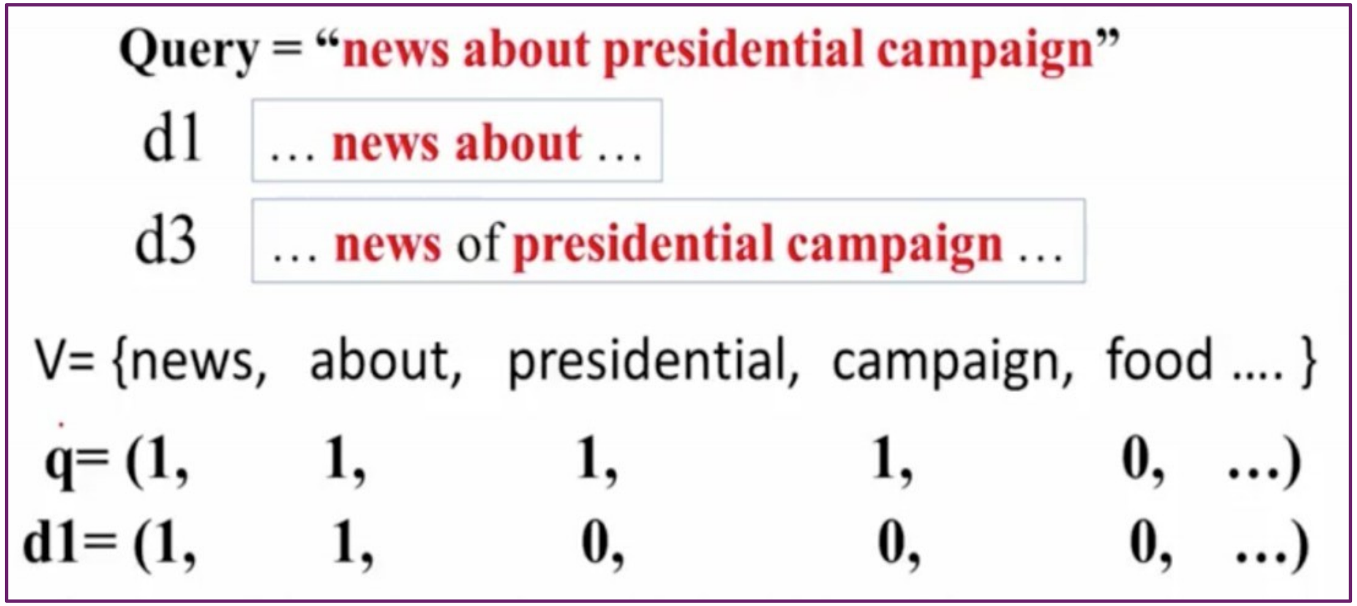
Ashish Upadhyay, Dr Stewart Massie
{a.upadhyay1, s.massie}@rgu.ac.uk

Outline

- Limitations of Lexical Methods
- Semantic Similarity
  - Pre-Deep Learning Era
  - Deep Learning Methods
    - Word Embeddings
    - Contextual Embeddings
- Using Word Embeddings in IR

Lexical Methods

- Only focuses on the occurrence of a term in document
- Completely ignores the semantic information
- Larger vocabulary means high dimensional document representations

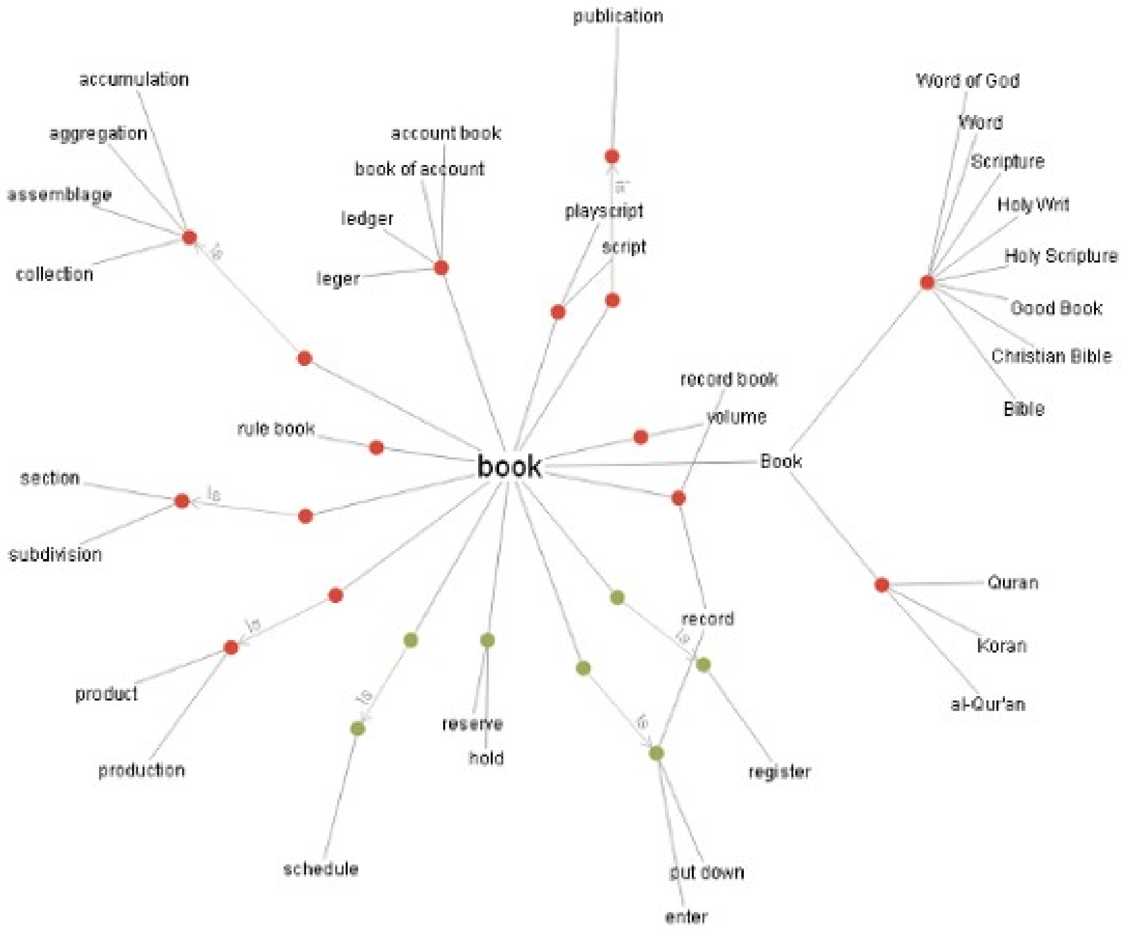


d4: Joe Biden held a drive-in rally in Philly

Semantic Similarity

Pre-DL: WordNet

- A taxonomy of similar terms grouped into sets of cognitive synonyms
- Words grouped into synsets: collection of synonyms
- Pro: very reliable
- Con: very difficult to develop



Pre-DL:
Term Co-Occurrence

- Term co-occurrence in the document
- A term appears with another in a document, meaning they are similar
- Create a term-term matrix
- Pros: fast training
- Cons: size proportional to vocab; relies of frequency

	...	cat	dog	labrador	fur	park	bark	...
...	...							
cat		23	4	0	12	0	0	
dog		4	28	23	13	22	28	
labrador		0	23	25	16	23	22	
fur		12	13	16	16	0	0	
park		0	22	23	0	21	3	
bark		0	28	22	0	3	16	
...								...

Word Embeddings

Embeddings

- Predict similarity between people
- Represent people based on five personality traits
- We're **"embedding"** each person into a five-dimensional vectors
- Calculate similarity between vectors (using cosine similarity)

	Trait #1	Trait #2	Trait #3	Trait #4	Trait #5
Jay	-0.4	0.8	0.5	-0.2	0.3
Person #1	-0.3	0.2	0.3	-0.4	0.9
Person #2	-0.5	-0.4	-0.2	0.7	-0.1

Word Embeddings

- Let’s take an example first
- Embedding of a word, “king”, is shown here
- Generated using GloVe algorithm (one of the many available)

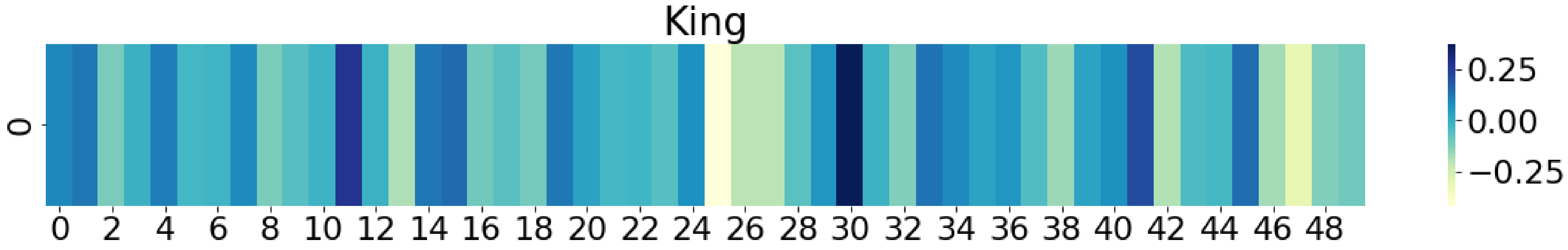
“king”

```
array([ 0.0938714,  0.1276532, -0.11074  , -0.0042425,  0.1117243,
       -0.025115 , -0.0163979,  0.0881518, -0.1149841, -0.0577023,
       -0.0142648,  0.2777942, -0.0063614, -0.1826651,  0.1269499,
        0.1520556, -0.0965191, -0.0586159, -0.1038407,  0.1235859,
        0.0364872, -0.0251094, -0.0213528, -0.0564594,  0.0766158,
       -0.4136213, -0.2001309, -0.2006333, -0.0639206,  0.0623409,
        0.3707707, -0.007878 , -0.1196748,  0.1323383,  0.0914674,
        0.0311732,  0.063902 , -0.0477497, -0.1585827,  0.0309053,
        0.0746156,  0.2174163, -0.1886135, -0.040162 , -0.0281981,
        0.1457275, -0.1697671, -0.2996754, -0.1198739, -0.094971 ],
      dtype=float32)
```

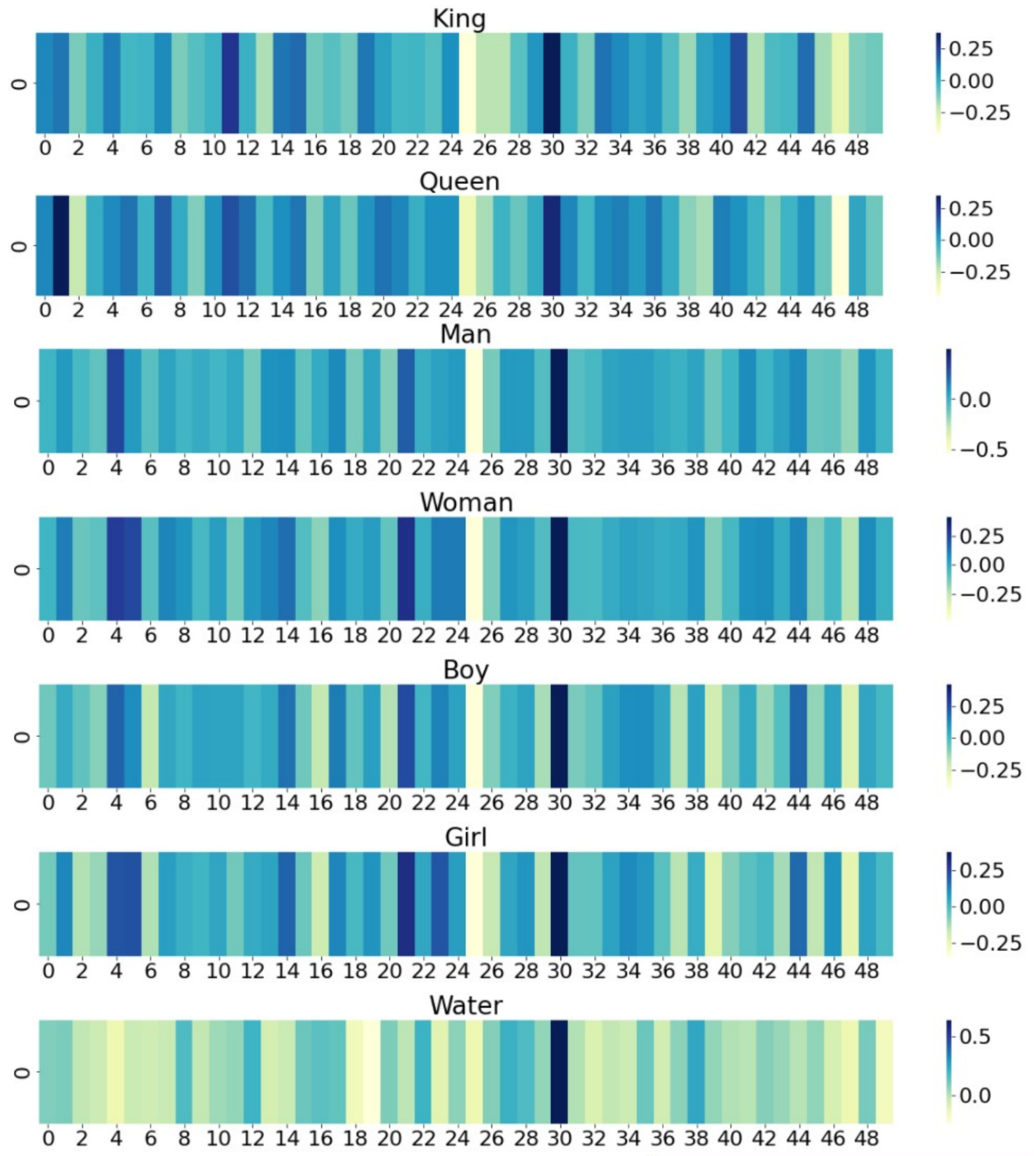
Embedding: GloVe
Dimensions: 50
Trained on: Wikipedia + Gigaword 5 (6B tokens)
Number of vectors: 400,000

Word Embeddings

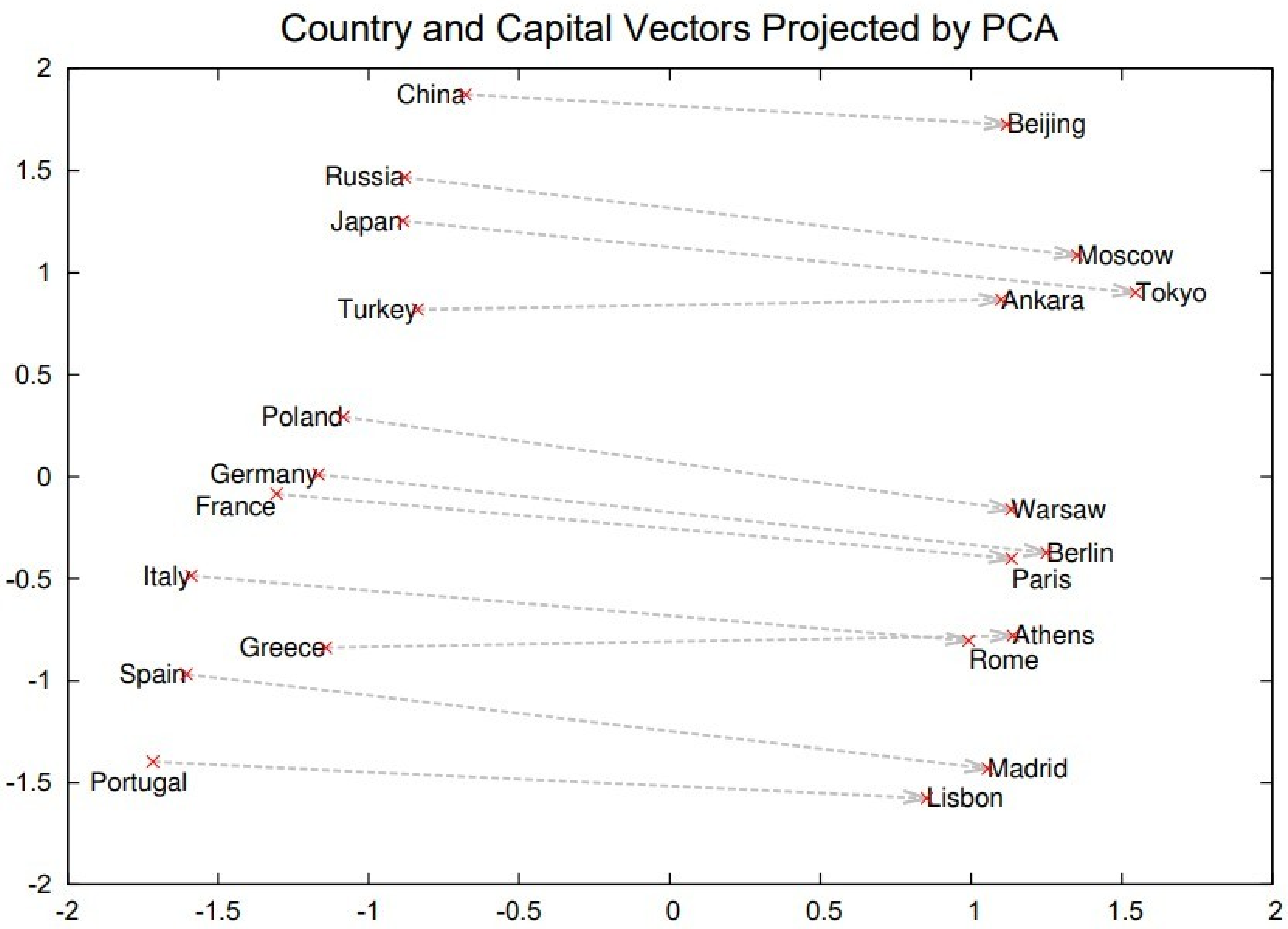
- What does that vector mean?



Word Embeddings

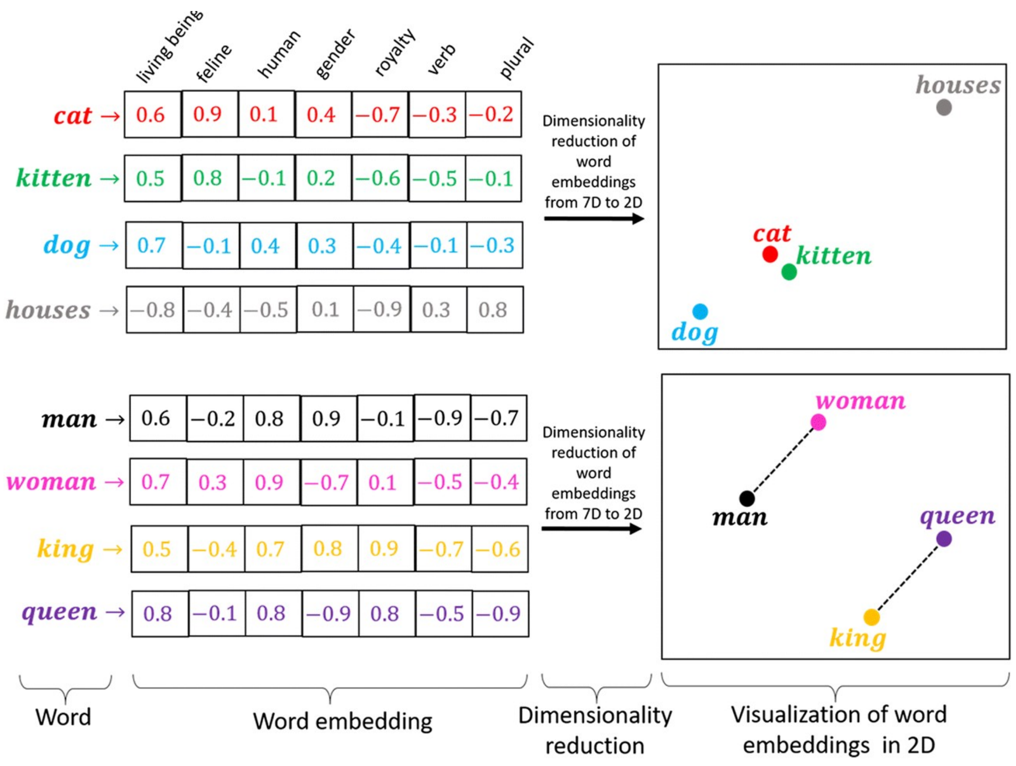


Word Embeddings



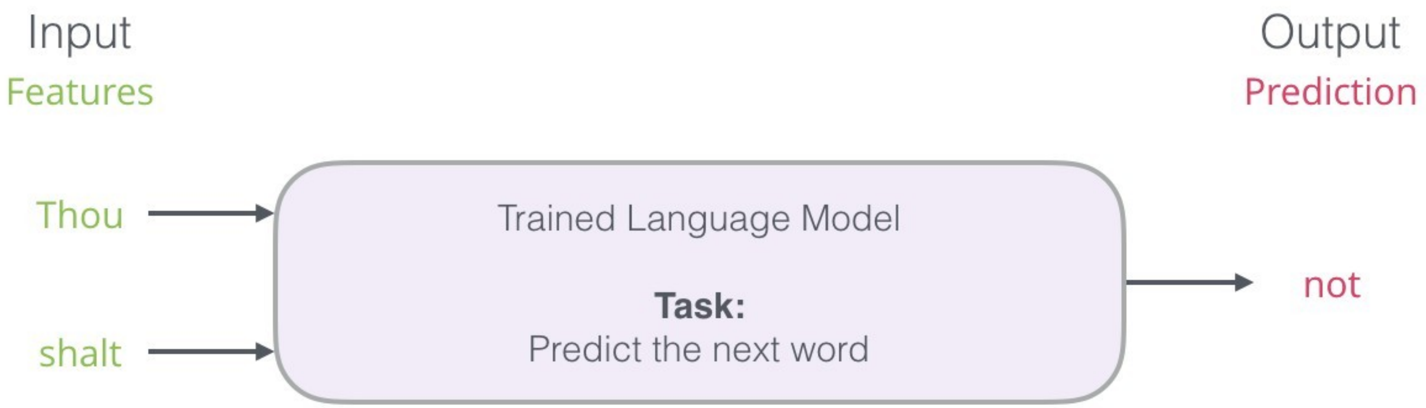
Word Embeddings

- Encode semantic representation of words in a dense vector
- Learned from the corpus of huge text based on the co-occurrence of terms
- Automatically learns the semantic similarity of words based on its surrounding
- Most popular ones: Word2Vec; GloVe; fastText
  - Use them off-the shelf to get embeddings of different words



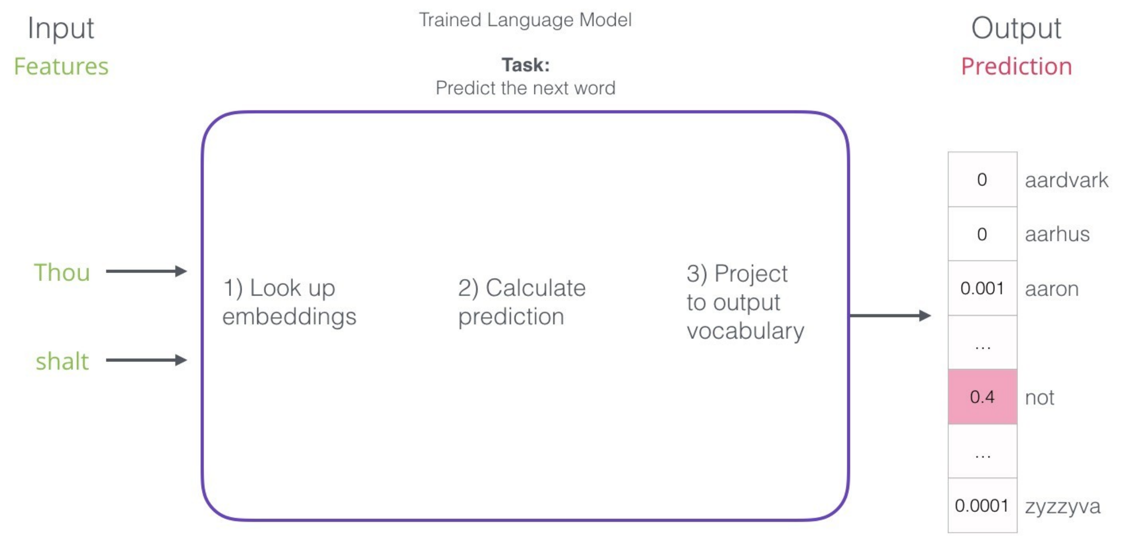
Learning Word Embeddings

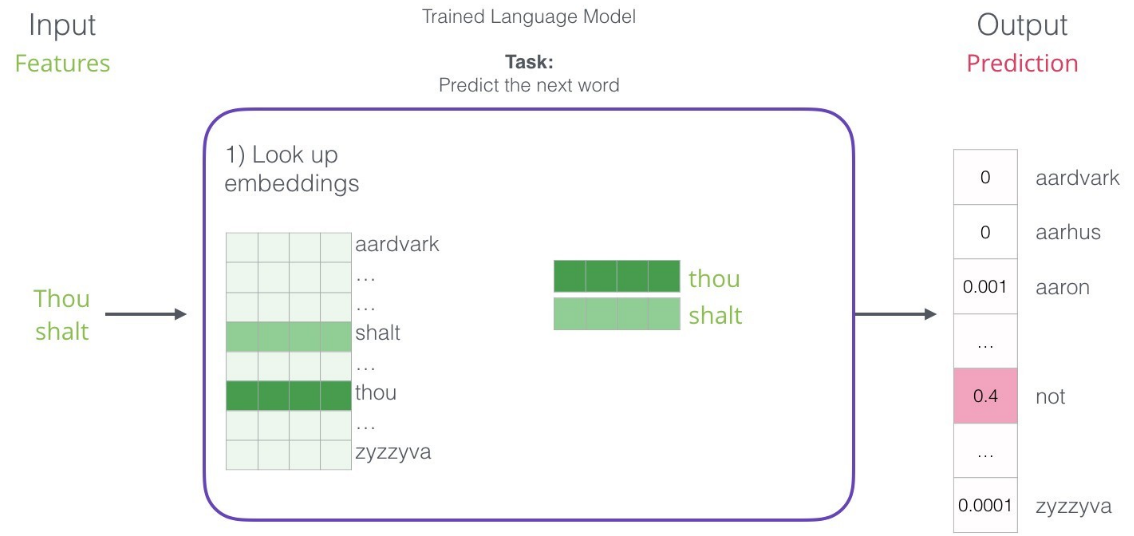
- Language Modelling:
  - Predicting the next word based on previous ones
- For example:
  - *"Thou shalt not make a machine in the likeness of a human mind"*
  - Take first two words from the sentence as input
  - Predict the third word as output



Language Models

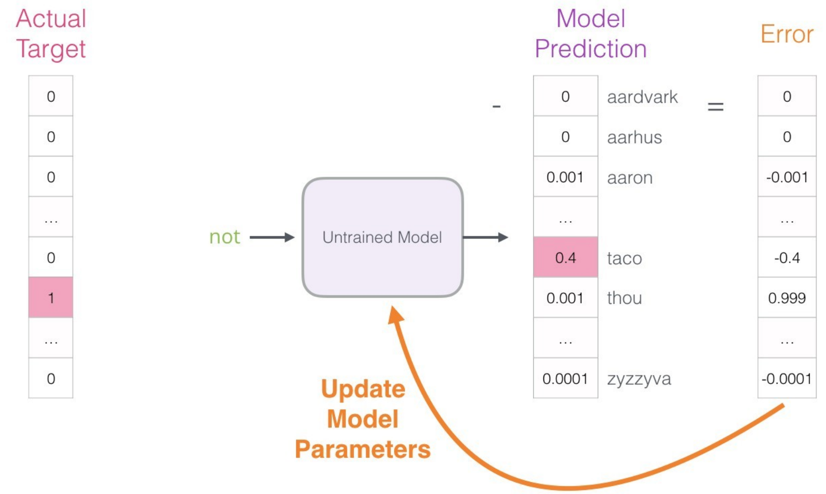
- Predict the probability of each word from vocabulary appearing next given some input sentence
- Embeddings are high-dimensional vectors
- Initially random



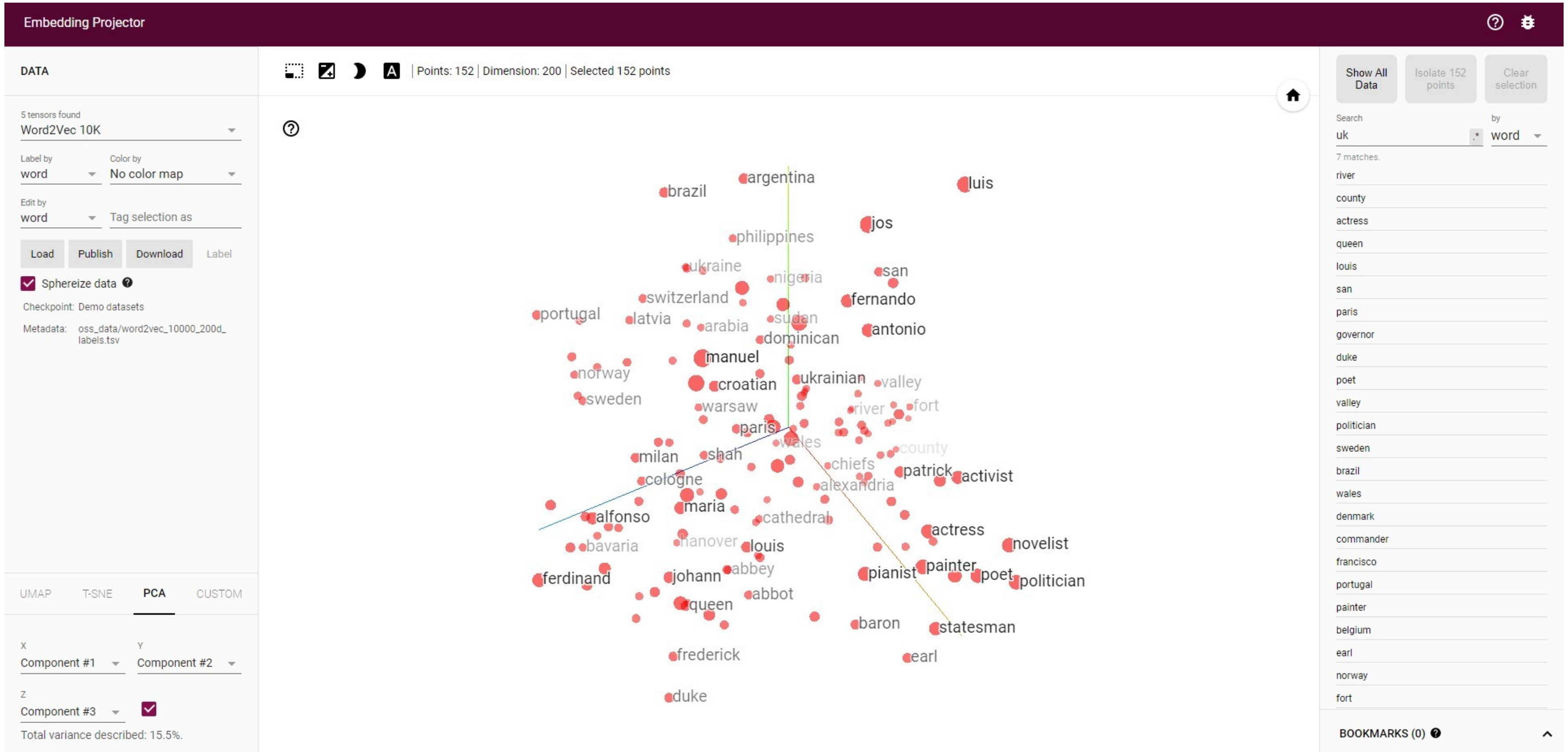


Language Models

- Give the model huge corpus of text data
- Calculate the errors made by model in predicting the next word
- Use the errors to modify the model parameters
- After many iterations: we get the right vectors for each word in the vocabulary
- These look-up embeddings can be saved separately to be used as the semantic representation of words



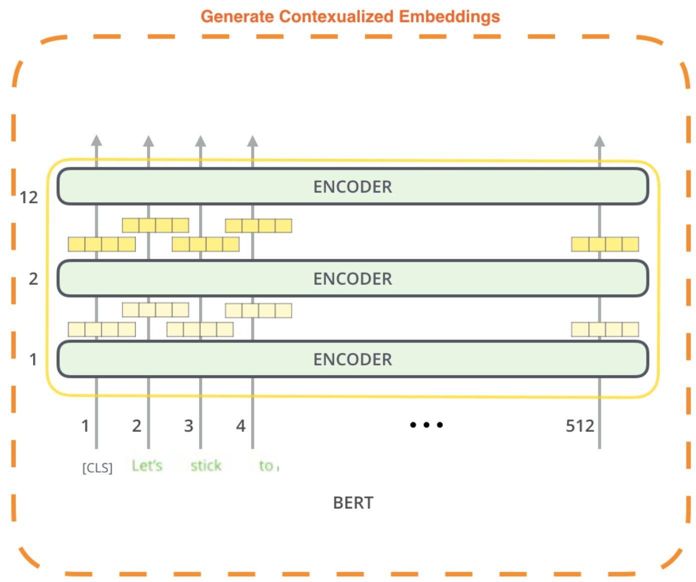
Embedding Projector



https://projector.tensorflow.org

Contextual Word Embeddings

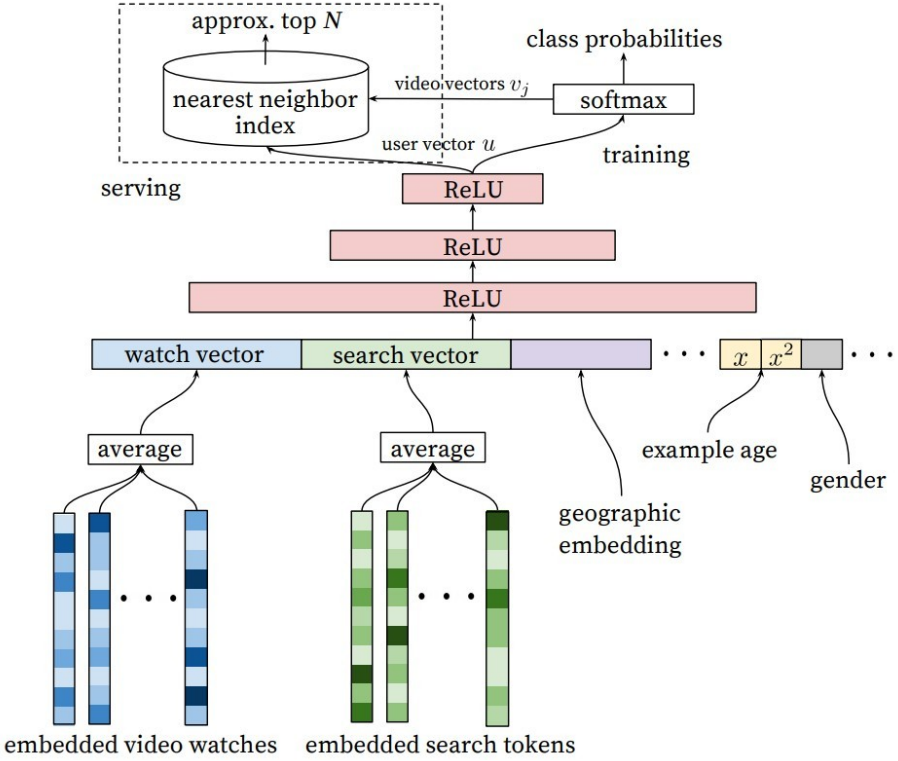
- Previously discussed embedding approaches miss the contextual information of the words
  - Embeddings for word ‘Bank’ in ‘Bank of a river’ and ‘Bank of Scotland’ will be same
  - But clearly, they have different meanings
- Instead of using just the embeddings from the Language Models, use the whole Language Model itself
  - Same word being used in two times in a sentence will have different embedding
  - Also, same word being used in two different sentences will have different embedding
- Popular Contextual Embedding tools: ELMo; BERT; GPT

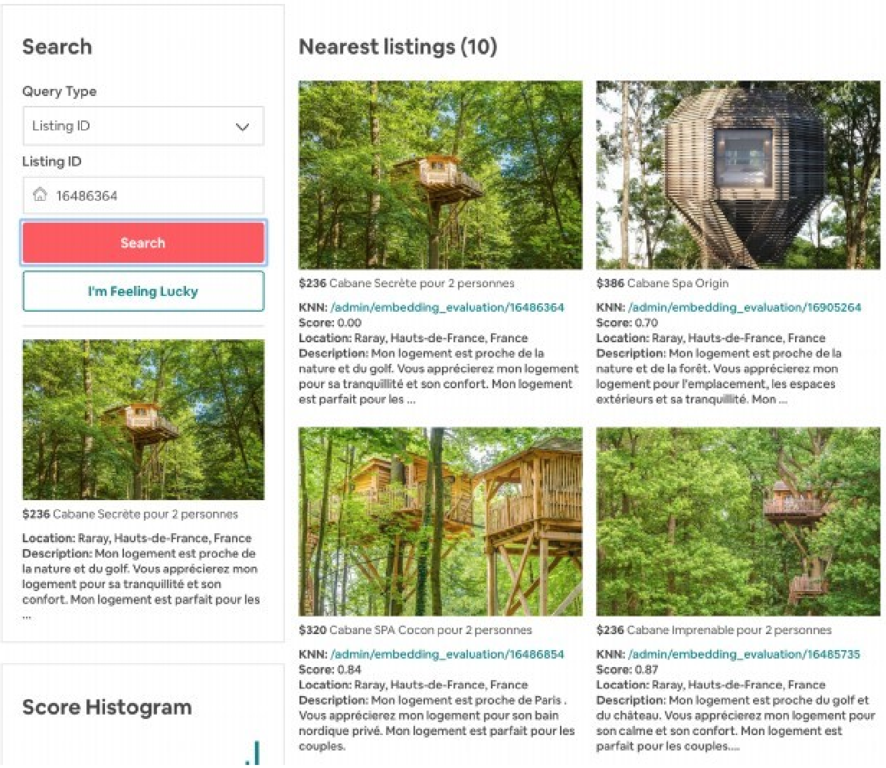


Using Word Embeddings in IR

Using Word Embeddings in IR

- Query Expansion
  - Add similar terms in the query
- Document Ranking
  - Embed the query as well as documents into similar vectors
- Some cool applications of embeddings: YouTube's & Airbnb's recommendation system





Thank You

Feel free to ask any questions.
If not, let’s head on to the lab then