NLP for me

PWYC Microcourse in Natural Language Processing
October 2024

Part 4 – Unsupervised Methods for Natural Language

Monday, October 28th, 2024

python

scikit learn

TensorFlow

Hugging Face

`nlpfor.me`

NLP from scratch

Agenda

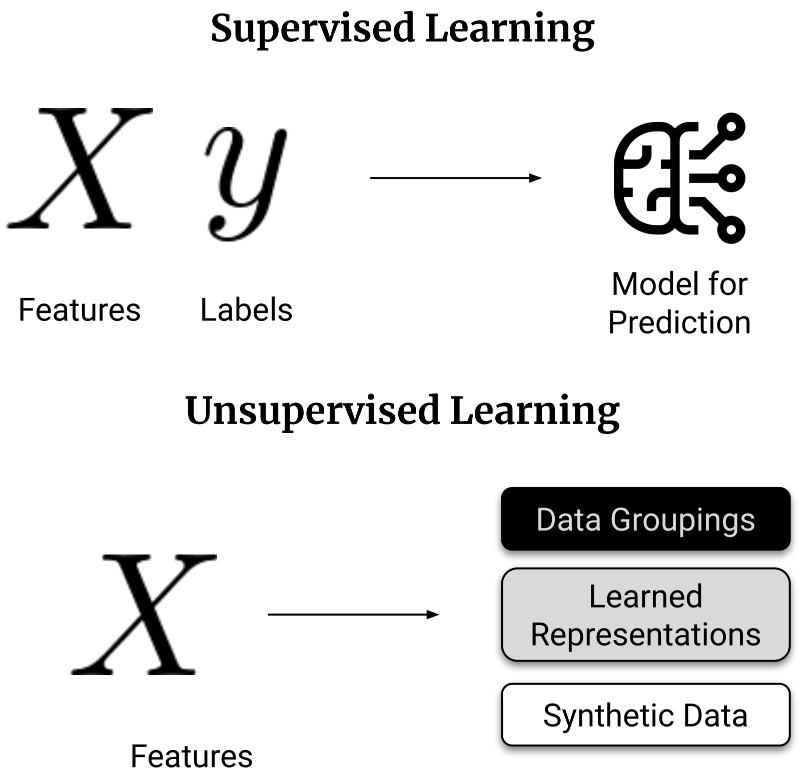
UNSUPERVISED LEARNING

Unsupervised Learning

Recall as we saw earlier in Part 3 that in *supervised learning*, we use our set of input features, X, and an associated set of *data labels*, y, to train a model to make predictions about unseen data.

In *unsupervised learning*, there are no data labels to be predicted; we are simply given the features. As such, unsupervised methods use statistical techniques to uncover patterns in data or learn alternative representations of the original data. There also unsupervised approaches for generative AI tasks as well.

There are a number of applications of unsupervised learning in the domain of NLP, however two major ones are that of *topic modeling* and learning *embeddings*.

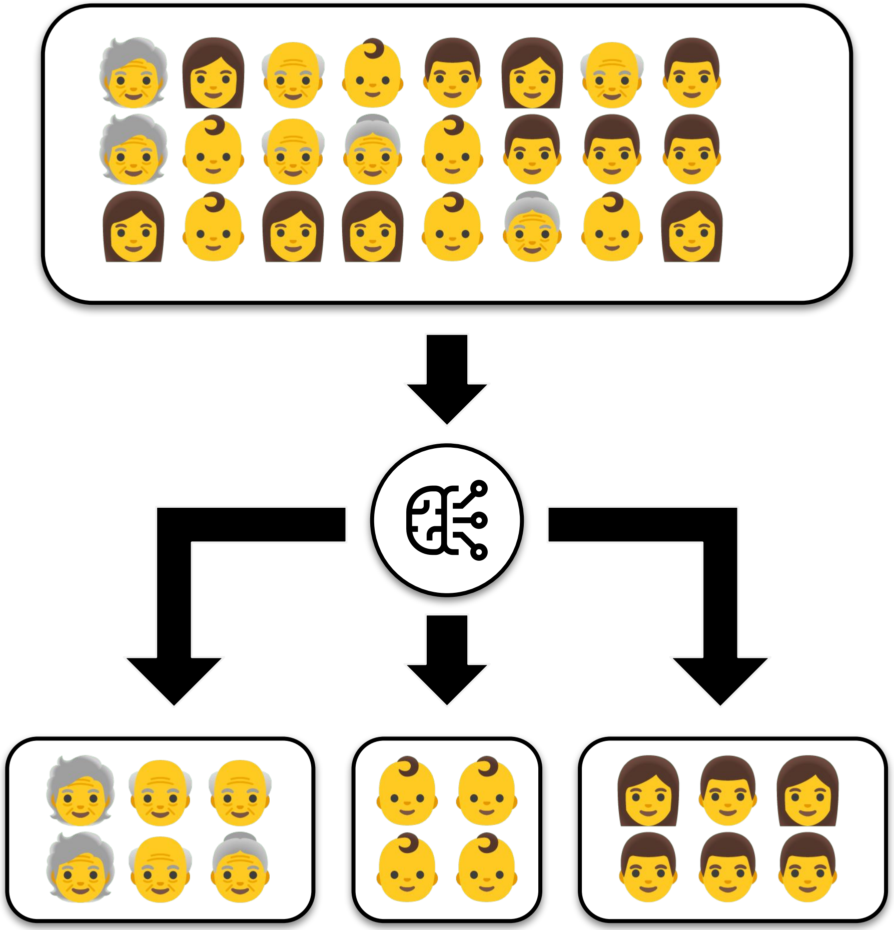


Clustering

The most commonly known type of unsupervised learning is that of *clustering*: given a dataset of observations of measurements of different features, can we find similar groupings of observations?

Clustering has many different applications in machine learning problems, a well known example being finding similar groups of customers based their attributes.

But wait, what is meant by the term "similar" in this case? What does it mean for two documents to be similar? We must first define this before delve deeper into topic modeling approaches.

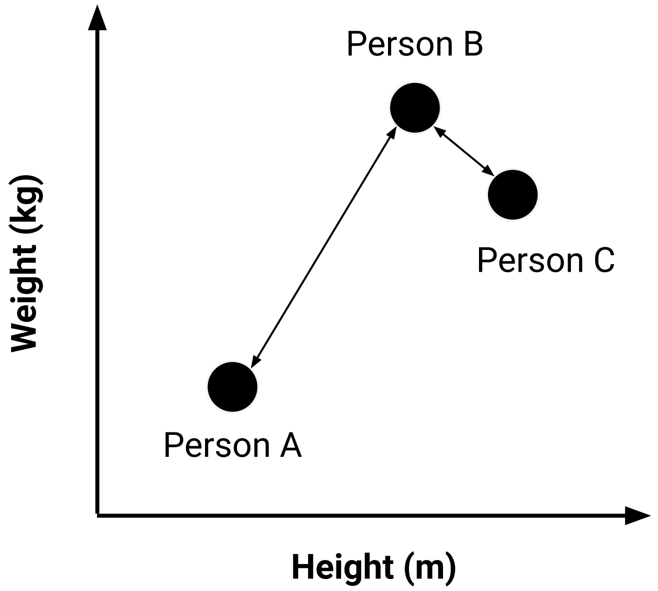


Similarity

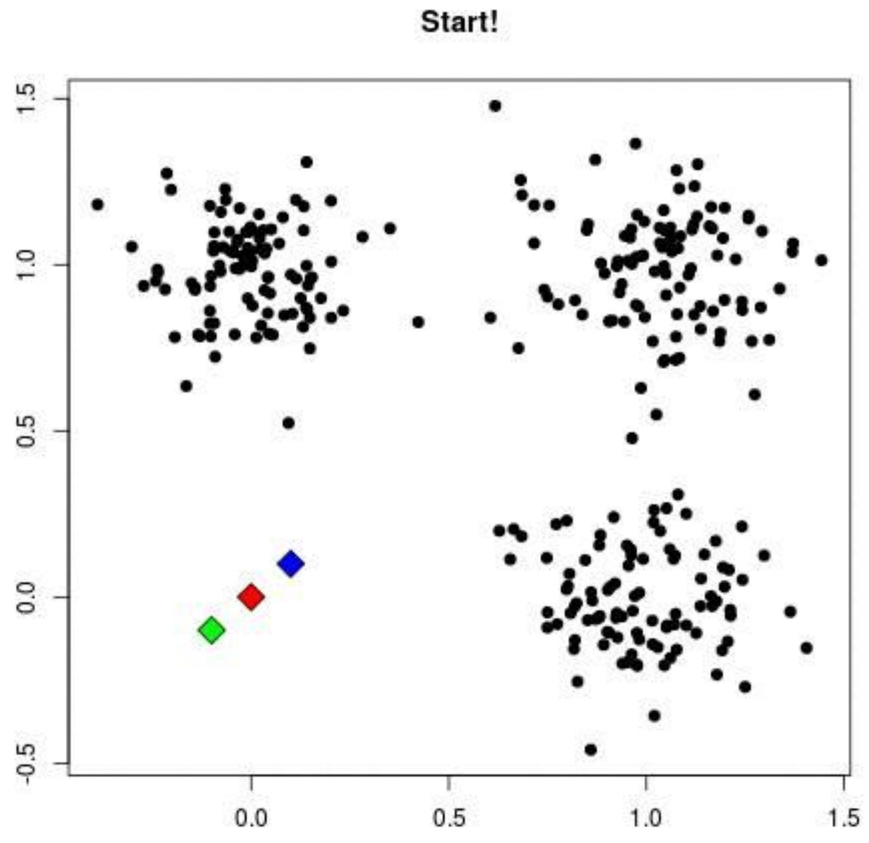
In machine learning, the notion of *similarity* and closeness are equivalent. We say two observations are more similar if they appear closer together in some *feature space* - a spatial representation where each dimension corresponds to a feature of the data.

A simple example: if three people have their height and weights measured, we can represent them as points in a 2-D space, where the x-axis represents their height in metres (m) and the y-axis their weight in kilograms (kg).

Here, we can see Persons B and C are more similar, based on where they appear in the feature space. This notion of similarity intuitively makes sense: if we were describing people based on only these attributes, we would say people who have about the same height and weight are more alike than those who do not.

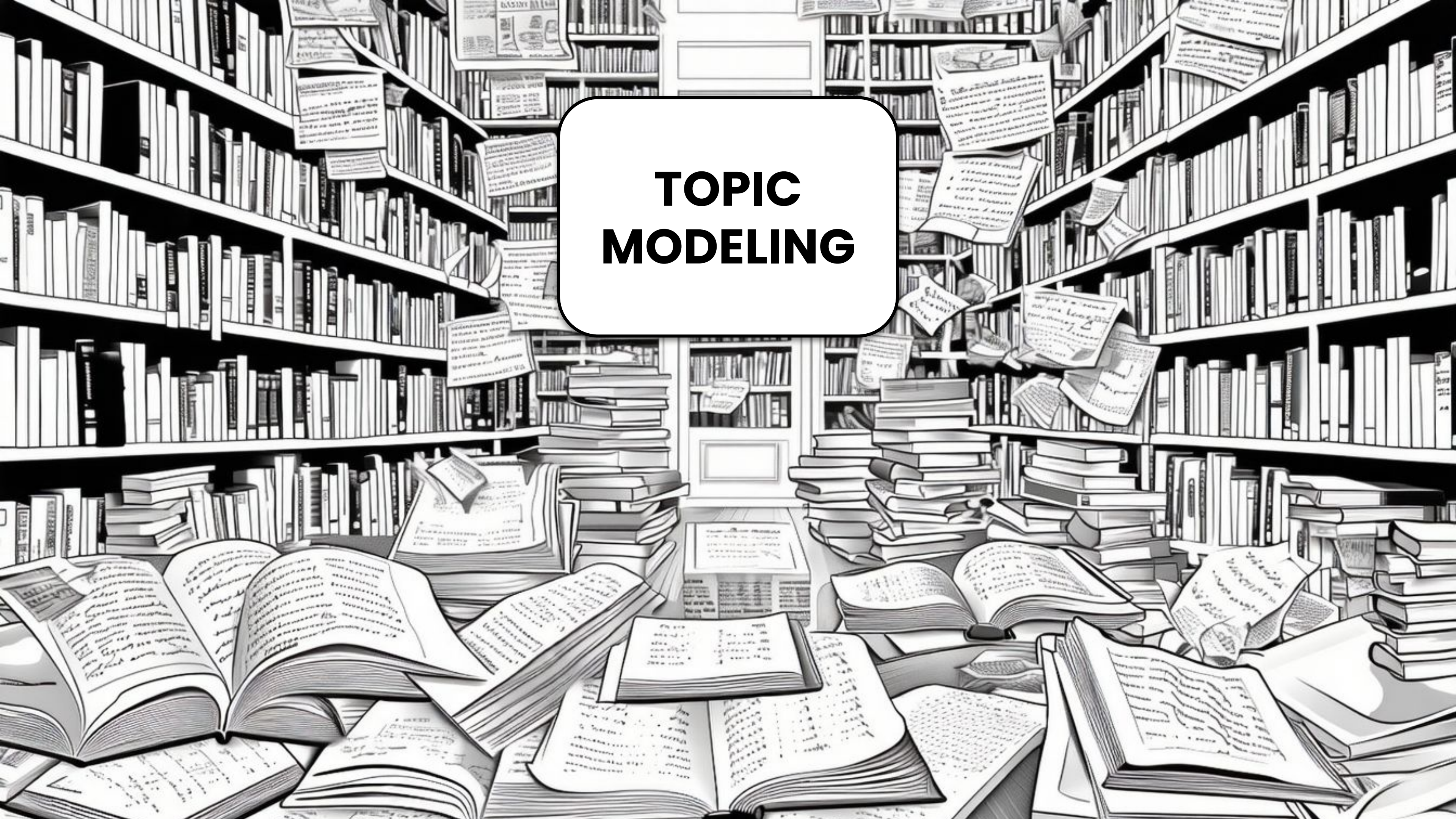


K-Means Clustering



mubaris.com/posts/kmeans-clustering/

NLP from scratch

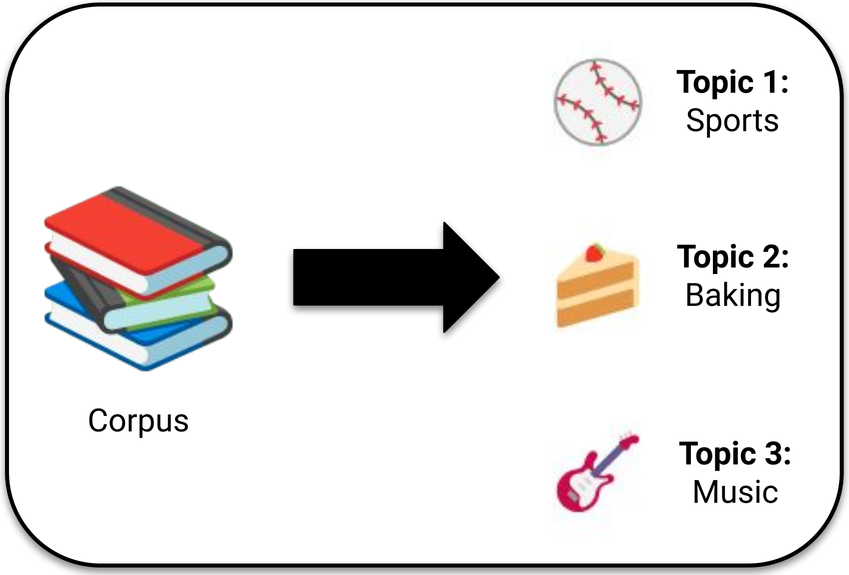


TOPIC MODELING

Topic Modeling

In NLP, we applying clustering specifically to a corpora of text with the goal to find *topics* that exist within.. This is usually done by looking at collections of words which commonly occur together in documents which are similar.

In this case, as in supervised learning, we use the *document-term* matrix, each row being a document and each column a token, and find groups of similar documents based on their content.

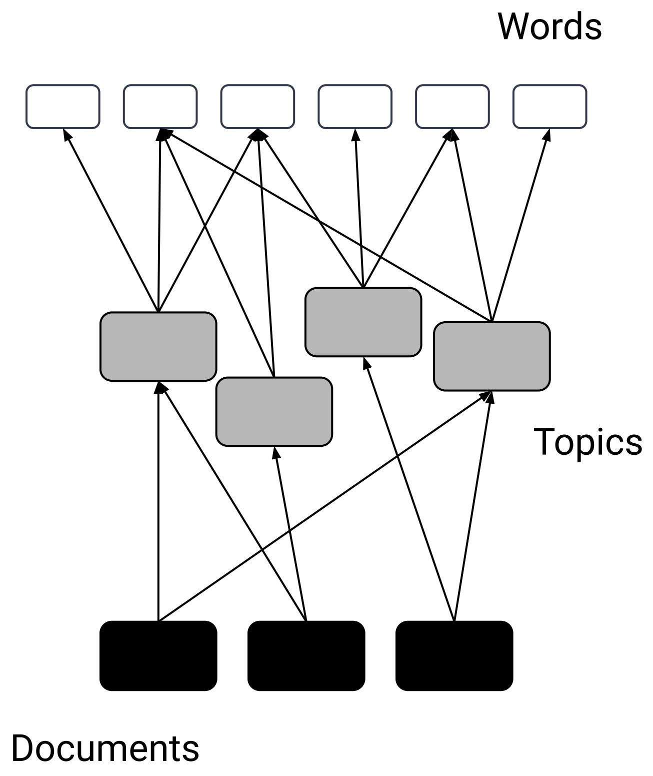


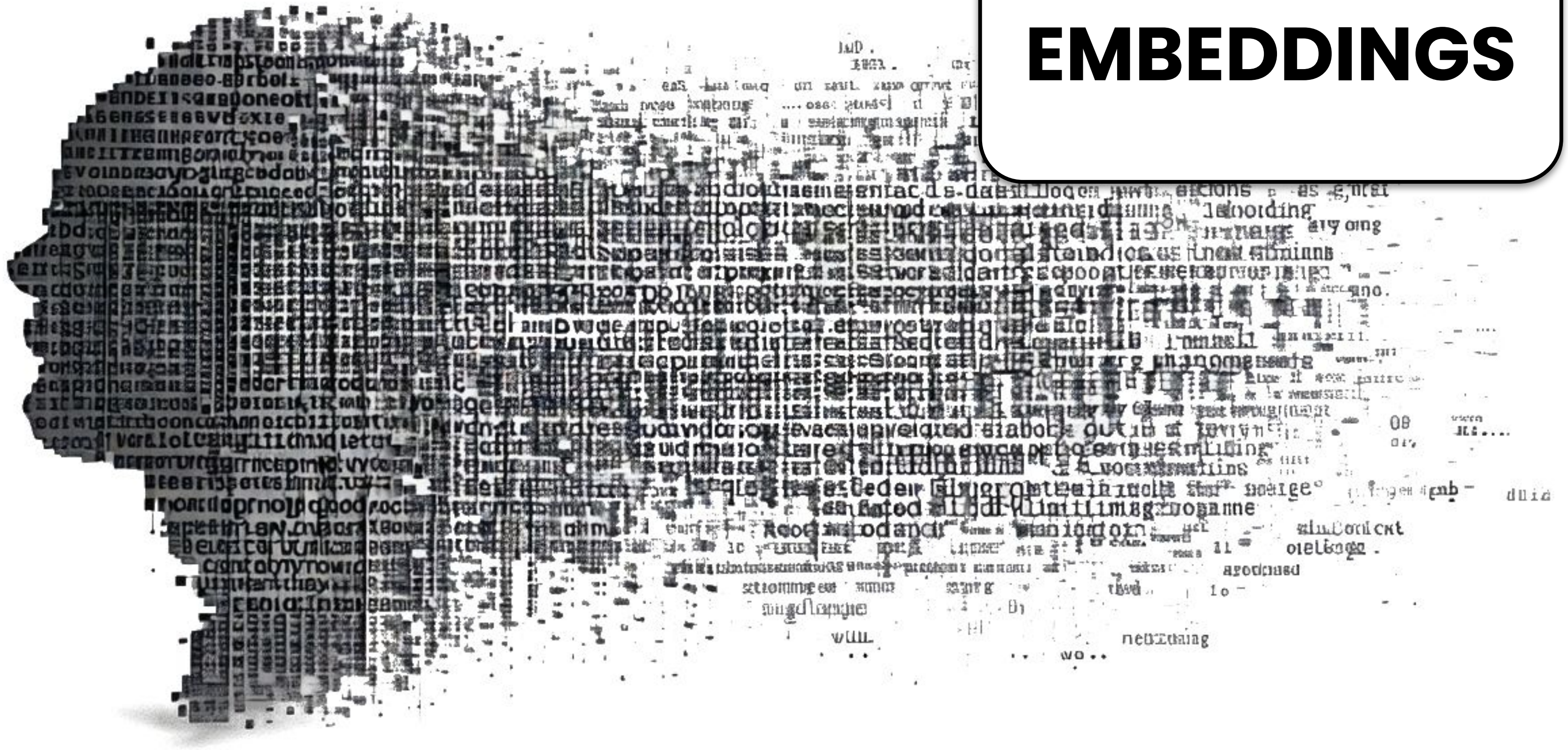
Traditional Approaches - LDA

Latent Dirichlet Allocation (LDA) is a statistical approach, which considers the probabilities of topics appearing in documents in a corpus, and those topics being composed of words with associated probabilities of occurring in each topic. These probabilities are initialized randomly at first, then updated them iteratively based on the actual occurrences of words in the documents in the corpus.

LDA treats documents as "bags of words" - the order of the words and syntax are not taken into account. Note also that topics are not 1-1 with documents: they are considered mixtures of topics, and so a given document can be include multiple different topics. The number of topics is not determined by the model but is a hyperparameter and must be specified in advance.

We will apply LDA to perform topic modeling on a sample dataset in the notebook for this part of the course.





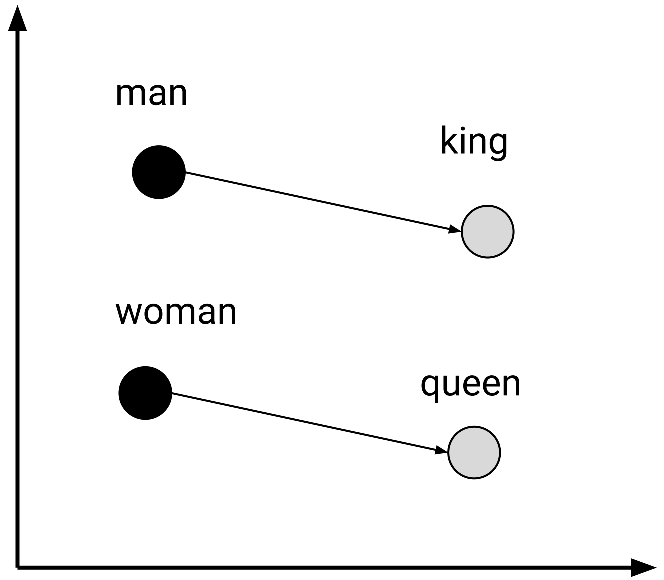
EMBEDDINGS

What is an Embedding?

An *embedding* is a way of representing text data numerically. We already saw a way of doing this previously, using vectorization methods. So how are embeddings different?

Embeddings are a more general way of representing text based upon its statistical properties over a given corpus. They are also not bound to a given dataset like a document-term matrix - you can create an embedding model from one body of text, and then use it on other corpora.

They have two distinct advantages over the approaches we've already seen. One was mentioned above, and the other is that they are not naive like the bag-of-words approach. Embeddings can capture the semantics (meaning) of words or sentences within a set of documents in a way that vectorization methods cannot.



Why use Embeddings?

For machine learning

Save on computation (maybe)

Capture semantics

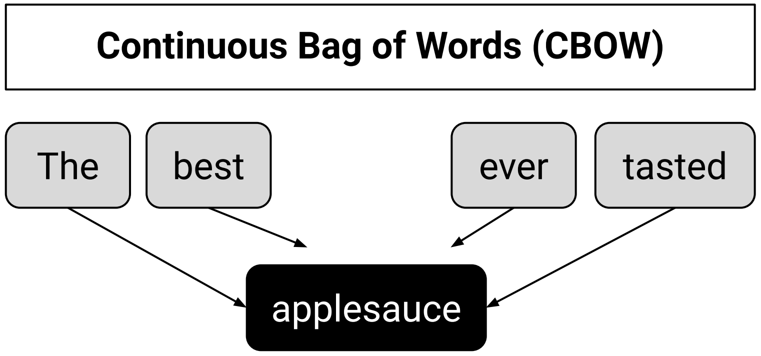
Domain specific

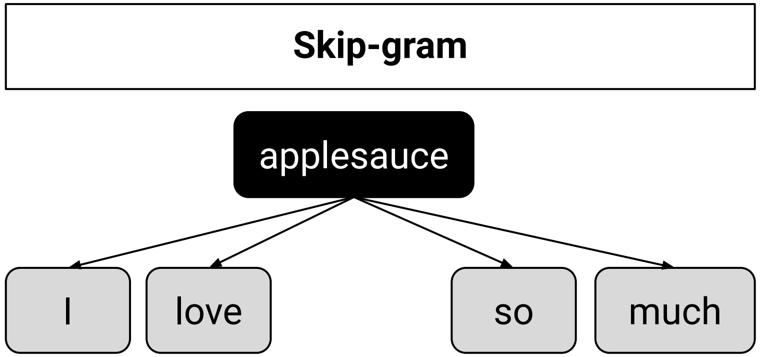
word2vec

Developed at Google in 2013, word2vec[1] was one of the first well-known and widely adopted embedding methods.

Though neural network approaches for text embeddings had existed for some time, word2vec had a number of advantages, including its ability to be applied on large datasets and trained incrementally (online learning).

word2vec can be applied in 2 ways: *continuous bag of words (CBOW)* and *skip-gram*. In the former, a target word is predicted using those around it - the *context* - and here the task is to fill in the missing word in a sentence. Skip-gram, on the other hand, does the opposite by predicting the most likely context words to appear around a given target word.





SBERT and Sentence Transformers

- Sentence Transformers (a.k.a. SBERT) is a Python library text and image embedding models
- Compute embeddings using Sentence Transformer models based on SBERT (Sentence-BERT) (Reimers & Gurevych, 2019[1]) from researchers at Ubiquitous Knowledge Processing Lab at University of Technical University of Darmstadt, Germany (UKP-TU)
- SBERT was an extension of the foundational BERT model from Google Research (2018) with a modified architecture and training objectives is better performance
- There now exist many Sentence Transformers models and these may be fine-tuned for specific use cases and corpora

1. Reimers, Nils, and Iryna Gurevych. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. arXiv:1908.10084, arXiv, 27 Aug. 2019. arXiv.org, https://doi.org/10.48550/arXiv.1908.10084.

End of Part 4

NLPfor.me
PWYC Microcourse in Natural Language Processing
October 2024

Part 4 - Unsupervised Methods for Natural Language

Monday, October 28th, 2024

python

scikit learn

TensorFlow

fastText

nlpfor.me

NLP from scratch