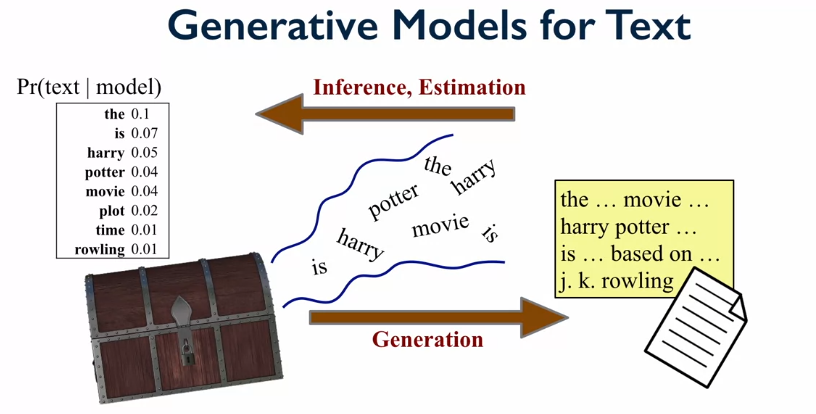
**Generative Models:**

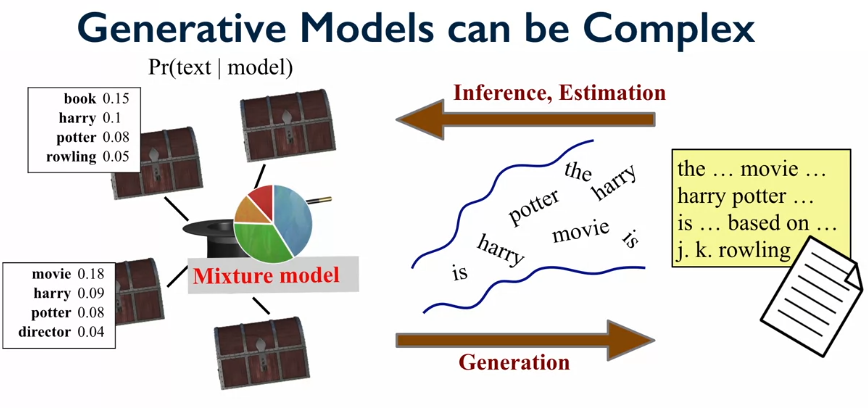
A generative model includes the distribution of data in a document, and then tells you how likely a given example is. An example is models that are able to predict the next word or phrase in a sentence e.g. Gmail: “Kind Regards,” at the end of an email.

**Generative Models for Text:**

You take a corpus of words and use the generative model to make a sentence or phrase based on the probabilities of the words in the corpus. If you reversed the process and used the sentence to generate a set of words with probability distributions (**inference, estimation**). The below shows the process from just one set of words (**unigram model**­)

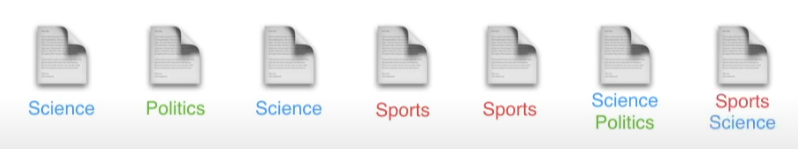


Let’s look at the example of having multiple topic models (chests in the above example) of words and see how the document is created from these sets. Now in the generative process words are pulled out based on some logic to create the document. For **inference** we now need to try a figure out which words belong to which topic models and what the topic models even are. This is called a **mixture model**.

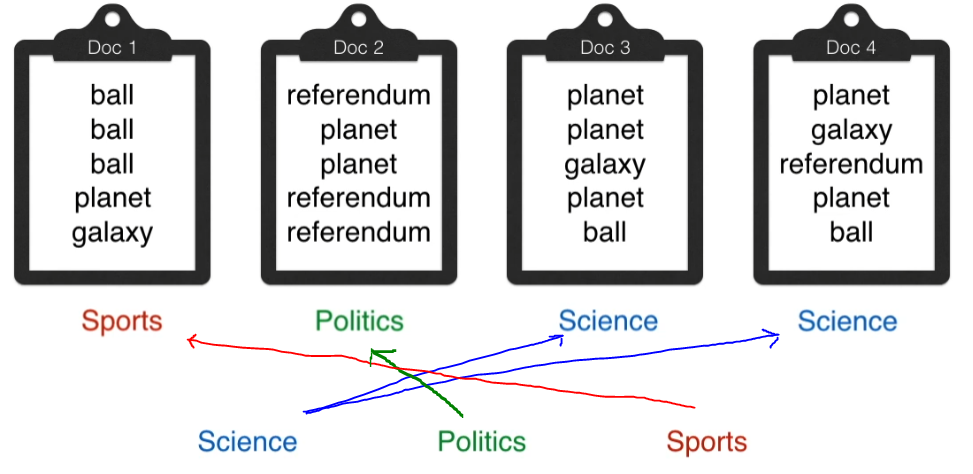


**Latent Dirichlet Allocation (LDA):**

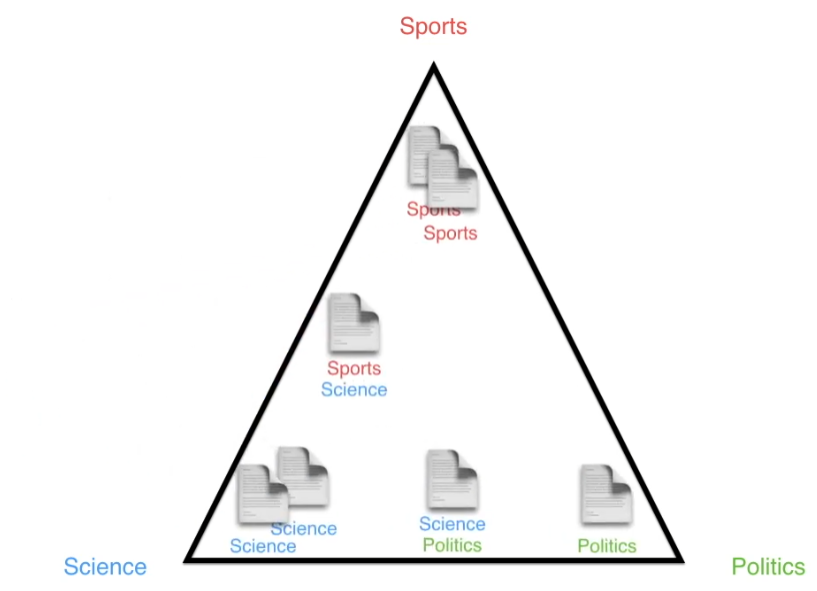
Imagine we have collection of documents e.g. new articles, and each has a topic (sports, science, politics, or some combination. The problem is that we don’t know the topics to begin with, we only know the words inside the articles, and we want to find out the topic!



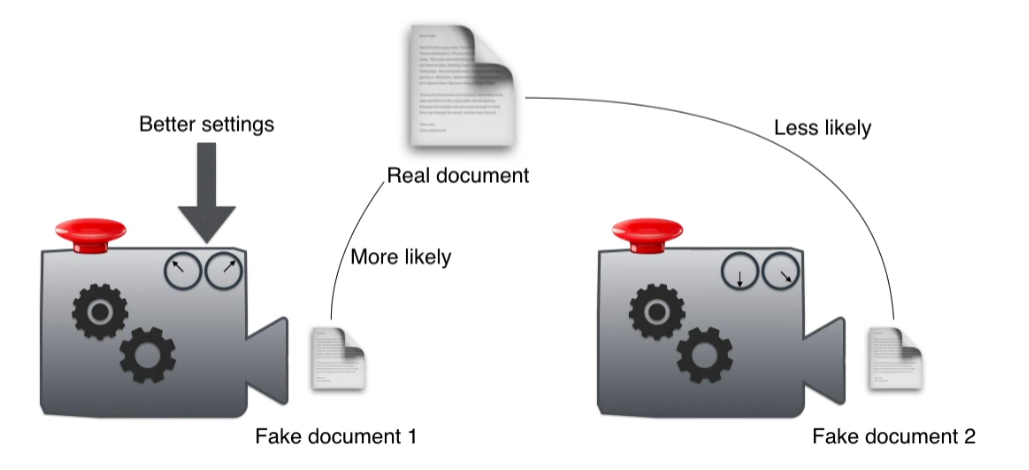
Imagine that the documents given only have a few simple words, then we might be able to easily assign a topic to this document. However, a machine only knows if two words are the same or different, or if the words appear in the same document or not. To find the topics we can use LDA.

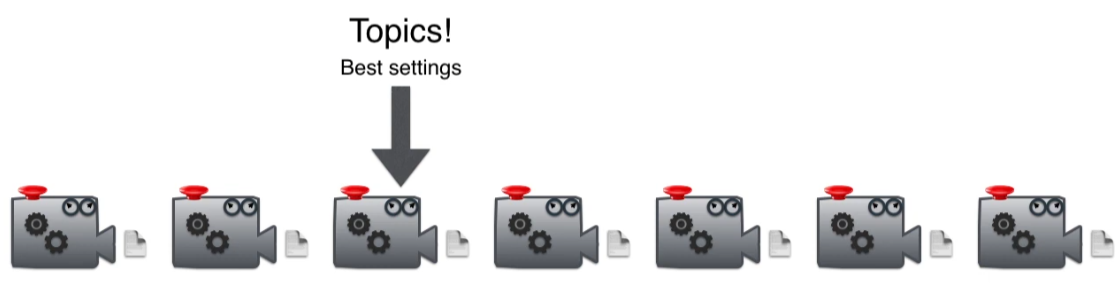


LDA solves this problem by using a geometric approach, this can be visualized bellow:



A way to think about LDA is that it’s a machine that generates documents, this machine has some setting that can be adjusted. Its most likely that the document that is generated is nonsense, with a small probability of generating the original articles. What LDA does is that is runs hundreds of these machines and then compared the documents produced with the original document, for the machine that generated a document that is most similar we know that that machine has better settings. From these best settings we can figure out the topics!





For more information watch is incredible video! <https://www.youtube.com/watch?v=T05t-SqKArY>

**Topic Modelling Summary:**

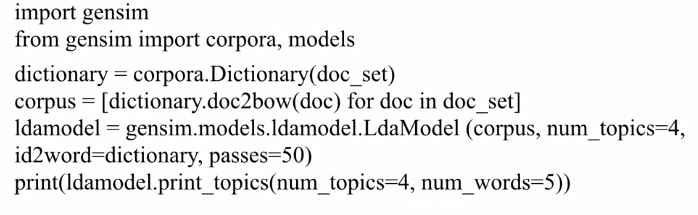
* It is a great tool for exploratory text analysis as it can tell you what the documents are about e.g. tweets, reviews, news, articles.
* There are many ways to do this in python.

**Python for LDA:**

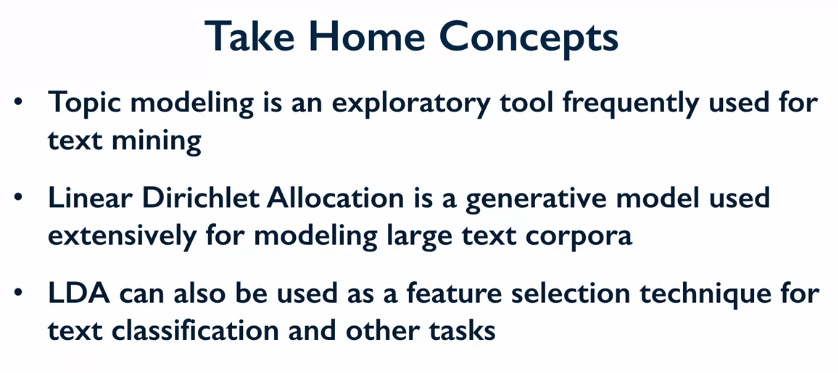
* Packages: genism, Ida
* We’re going to us genism.

Before you use any of these packages, we need to pre-process the text using some of the following: tokenization, normalization (lowercase or uppercase), Stop words removed, Stemming, Lemmatization. Once the pre-processing is done, then the words need to be converted into a matrix for, this could be Count Vectorize or TF-IDF or some other method.

For more information about the code below: https://radimrehurek.com/gensim/auto\_examples/core/run\_core\_concepts.html#sphx-glr-auto-examples-core-run-core-concepts-py



First import genism, then create a dictionary of all the words in the document to their ID’s (where “doc\_set” is this set of words). “doc2bow” creates a bag-of-words (BOW) of the documents found in the collection of documents. The bag-of-words model is a list of the number of times a word appears. E.g. image we have “coffee, milk, sugar, spoon” and the document “coffee milk coffee” then the BOW would be [2,1,0,0]. The variable Corpus is therefore a list of lists of all the BOW for every document in a collection of documents. The corpus is then used in the LDA model with the number of topics we want to learn, the dictionary mapping is also required. Then the topics can be printed out. LDA could also be used to find the topic distributions of the documents.



LDA is Latent Dirichlet Allocation, and is often used as the first step to try and understand what the corpus of documents is about.