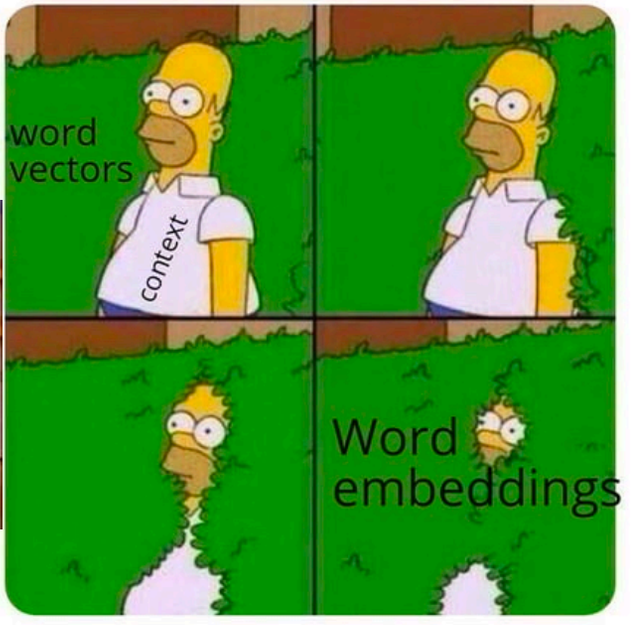
May 1, 2023.

DSML : NLP module.

Topic Modeling.

Class starts @ 9:05



When you penalize your Natural Language Generation model for large sentence lengths

Word embeddings in a nutshell

Recap:

* Document Vectorization: Bag of Words (BoW)
  TF/IDF.
* Word Vectorization: Continuous BoW,
  Skip-gram.
* Language Modeling: Naïve Bayes to predict
  the next word.

Agenda:

Classical NLP (No deep learning).

→ Parts-of-speech tagging.

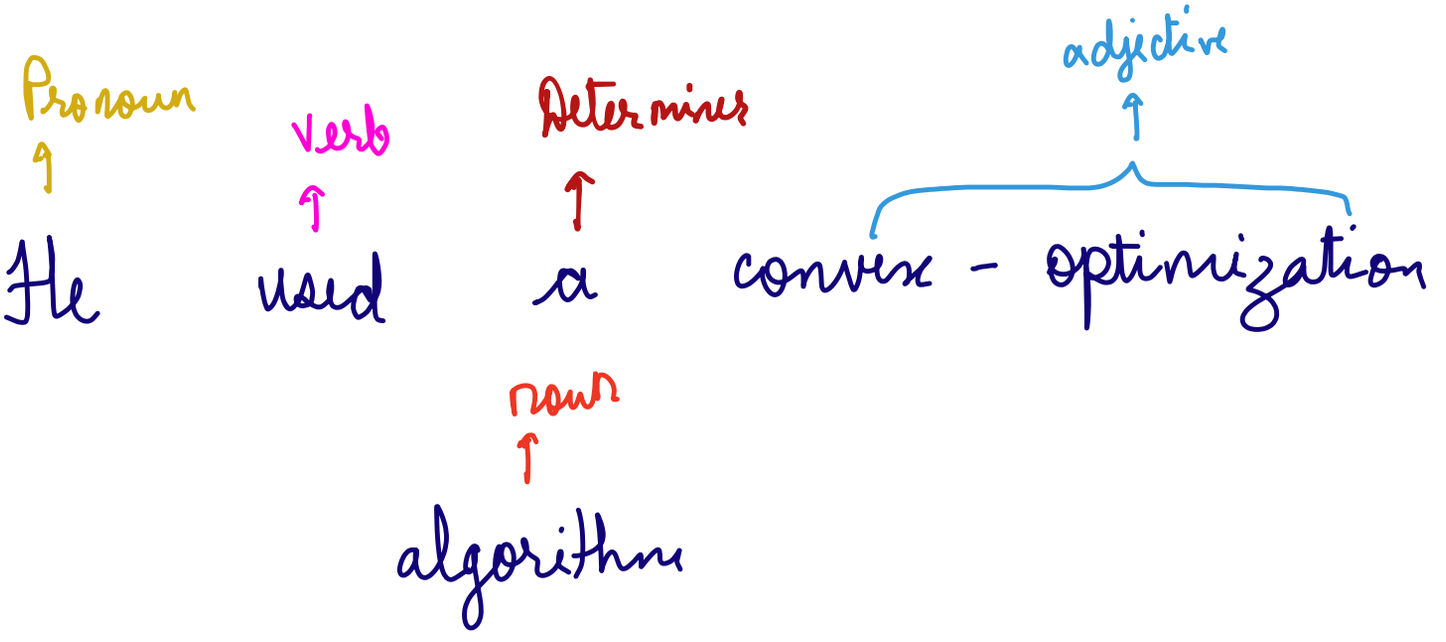
* Naive approach, using heuristics and our knowledge of English grammar.

→ Topic Modeling.

* Statistical approach to build a generative model.
* The approach is called Latent Dirichlet Allocation (LDA).

* Business Case: Extract "topics" of discussion.

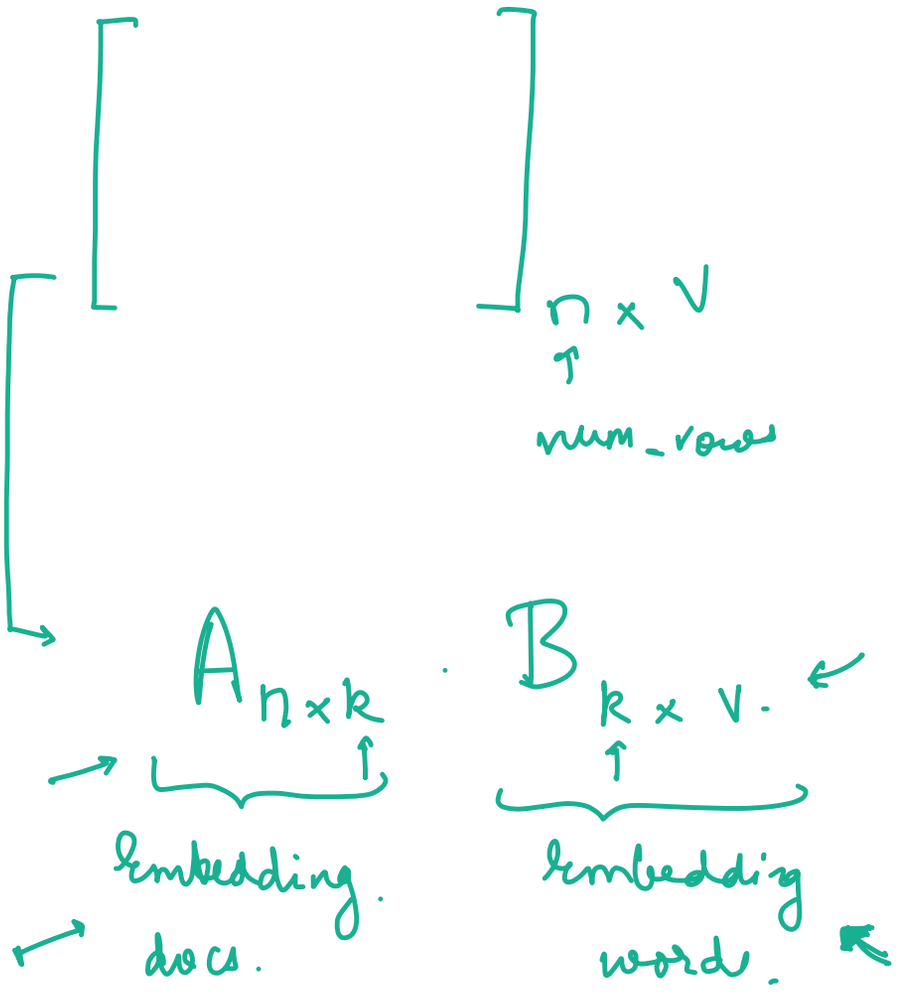
Parts of speech tagging

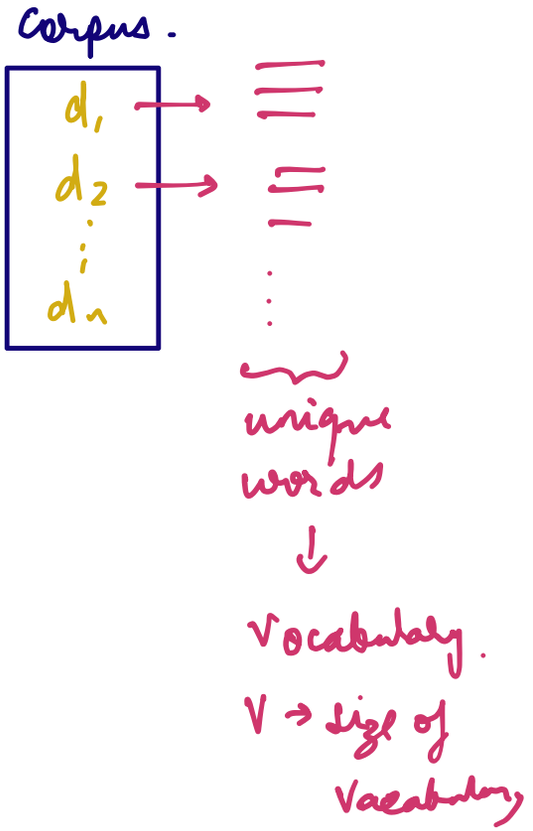


Hidden Markov models.

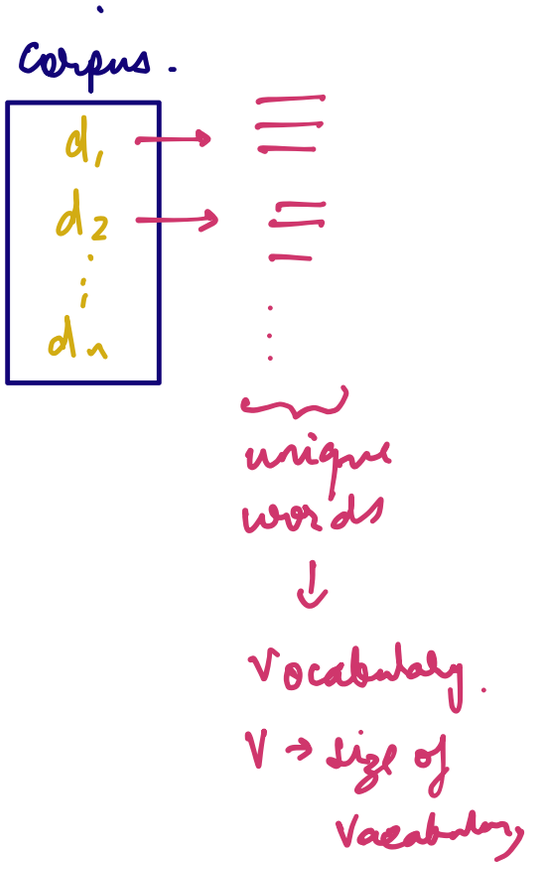
Topic Modeling.

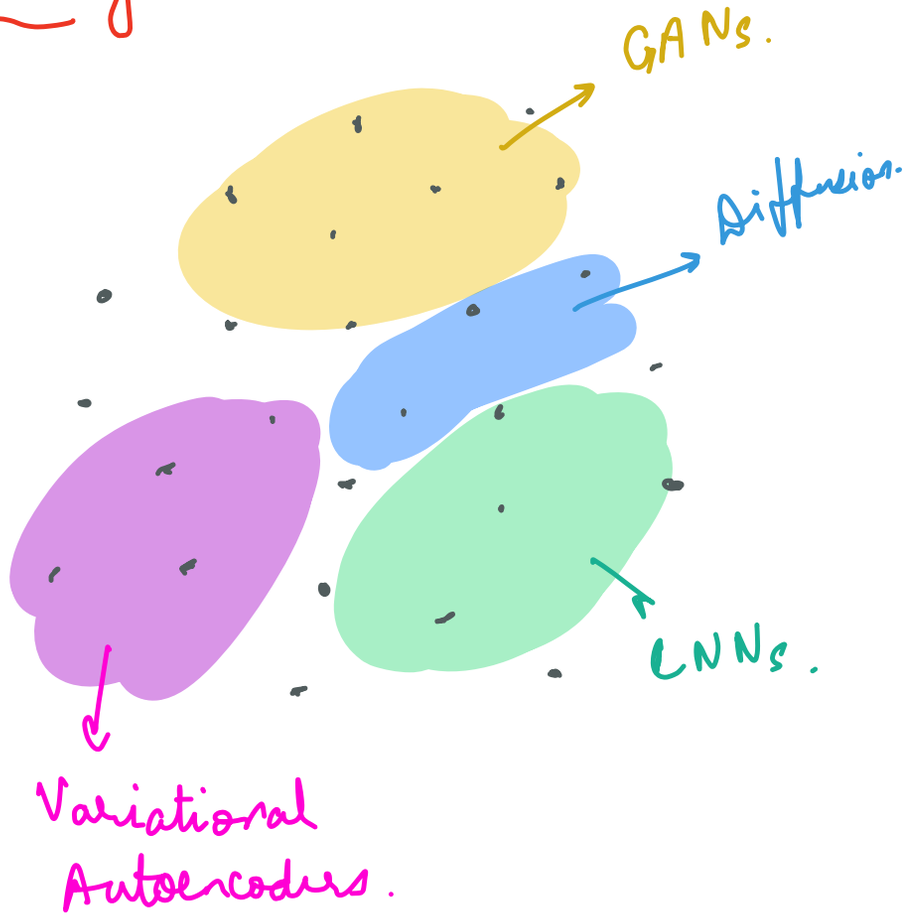
→ Matrix factorization.





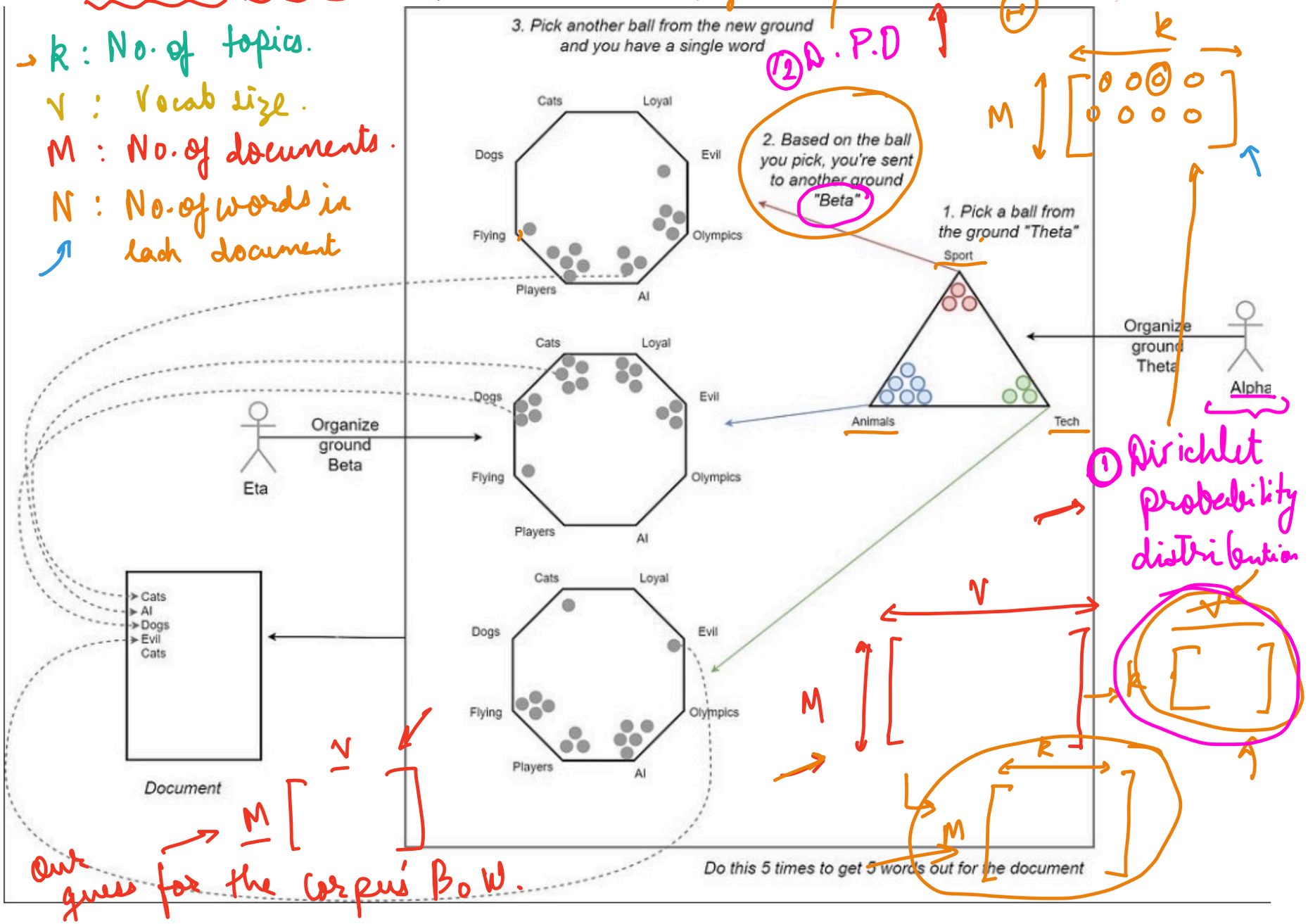
Topic modeling as Clustering.





Latent Dirichlet Allocation:

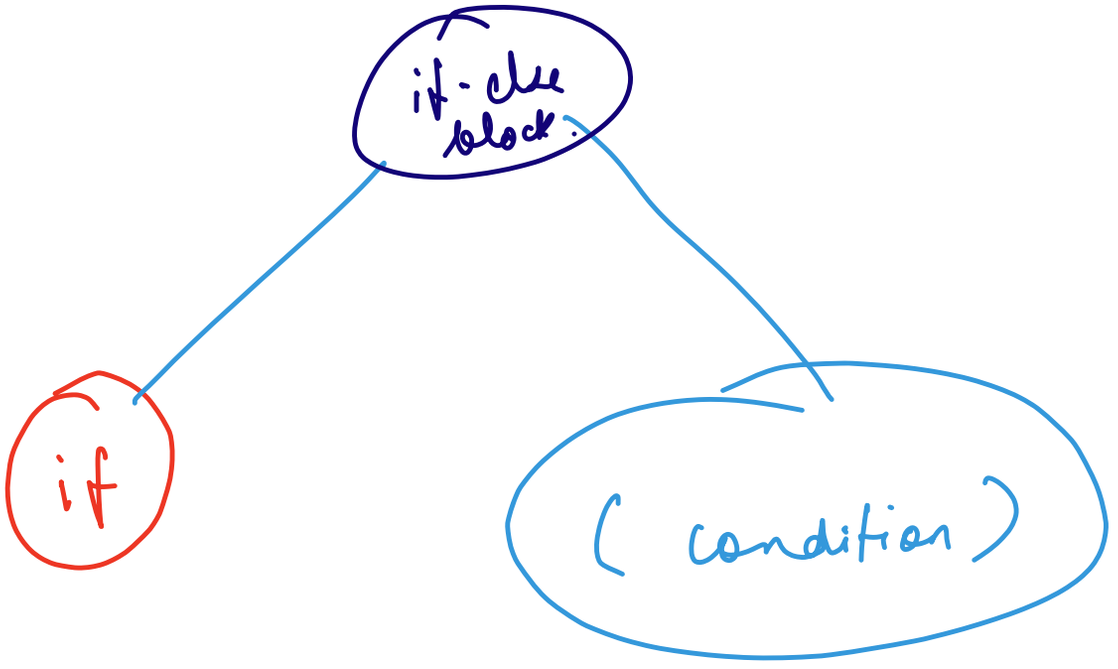
- k : No. of topics.
- V : Vocab size.
- M : No. of documents.
- N : No. of words in each document

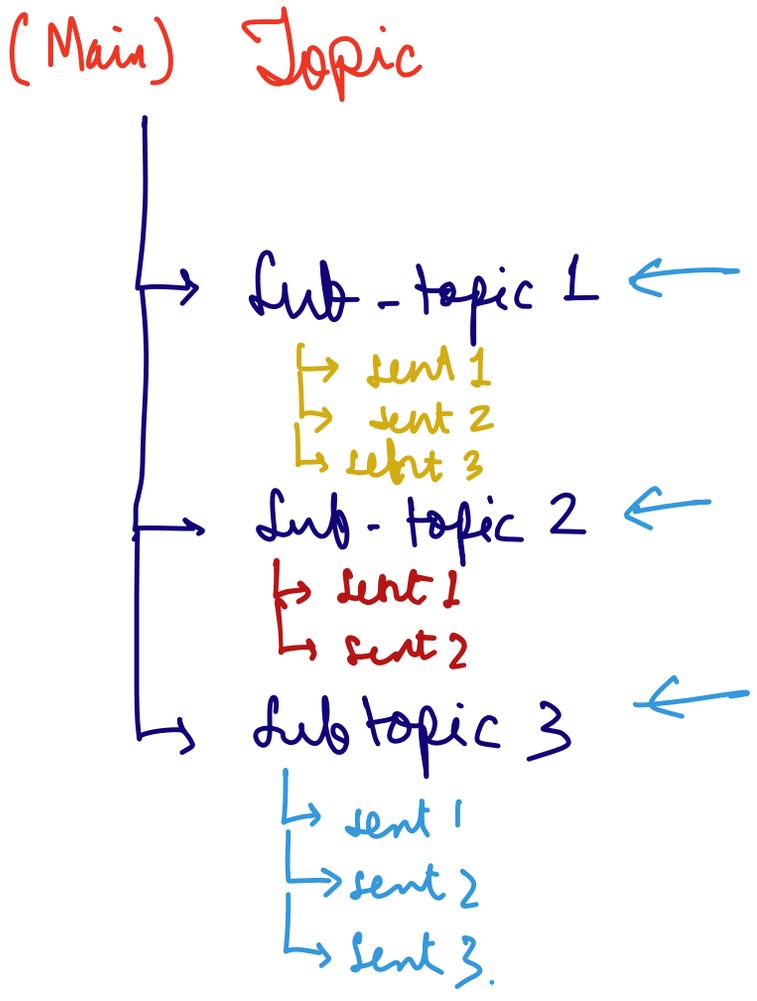


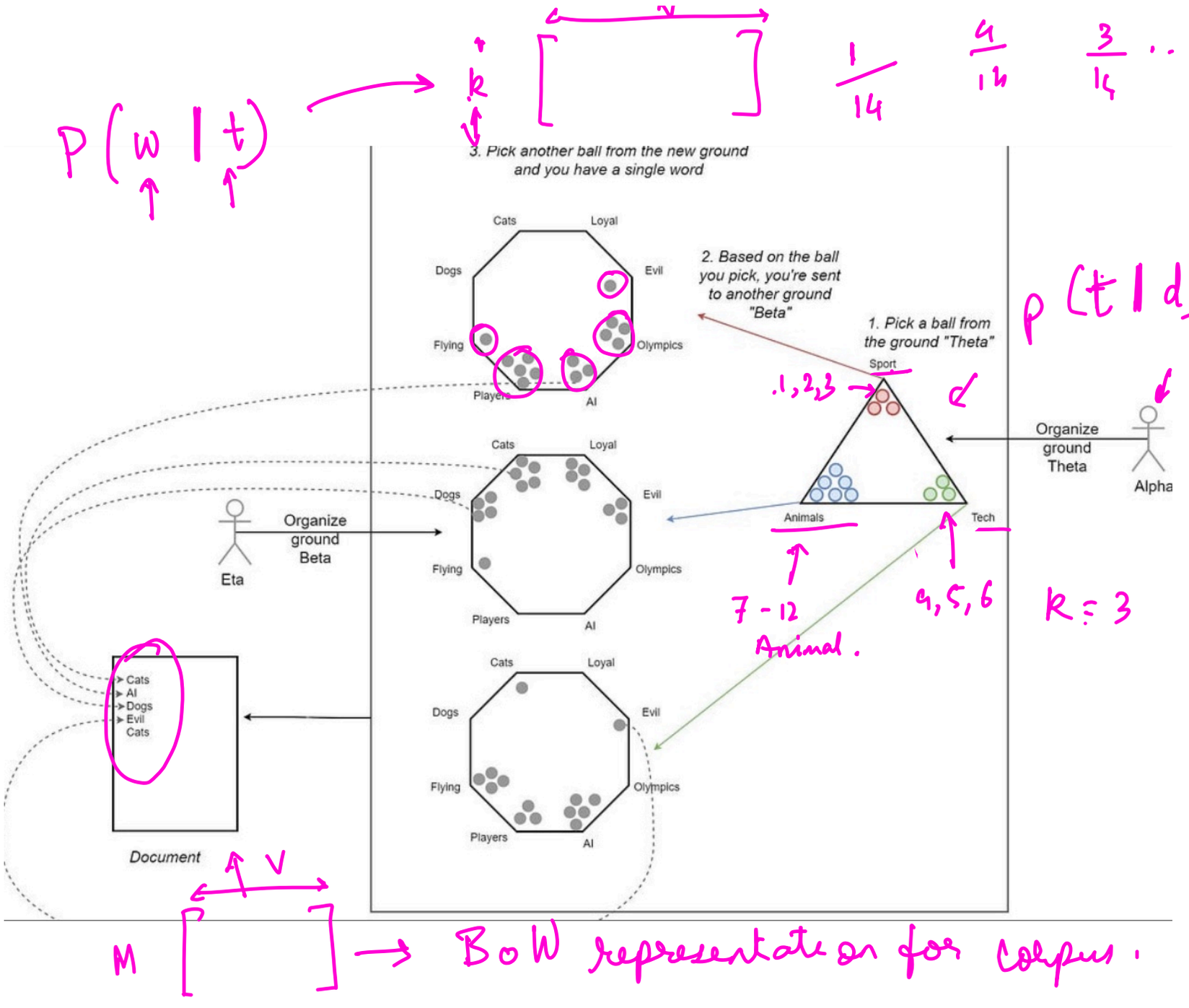
Our guess for the corpus BoW.

(1) Dirichlet probability distribution

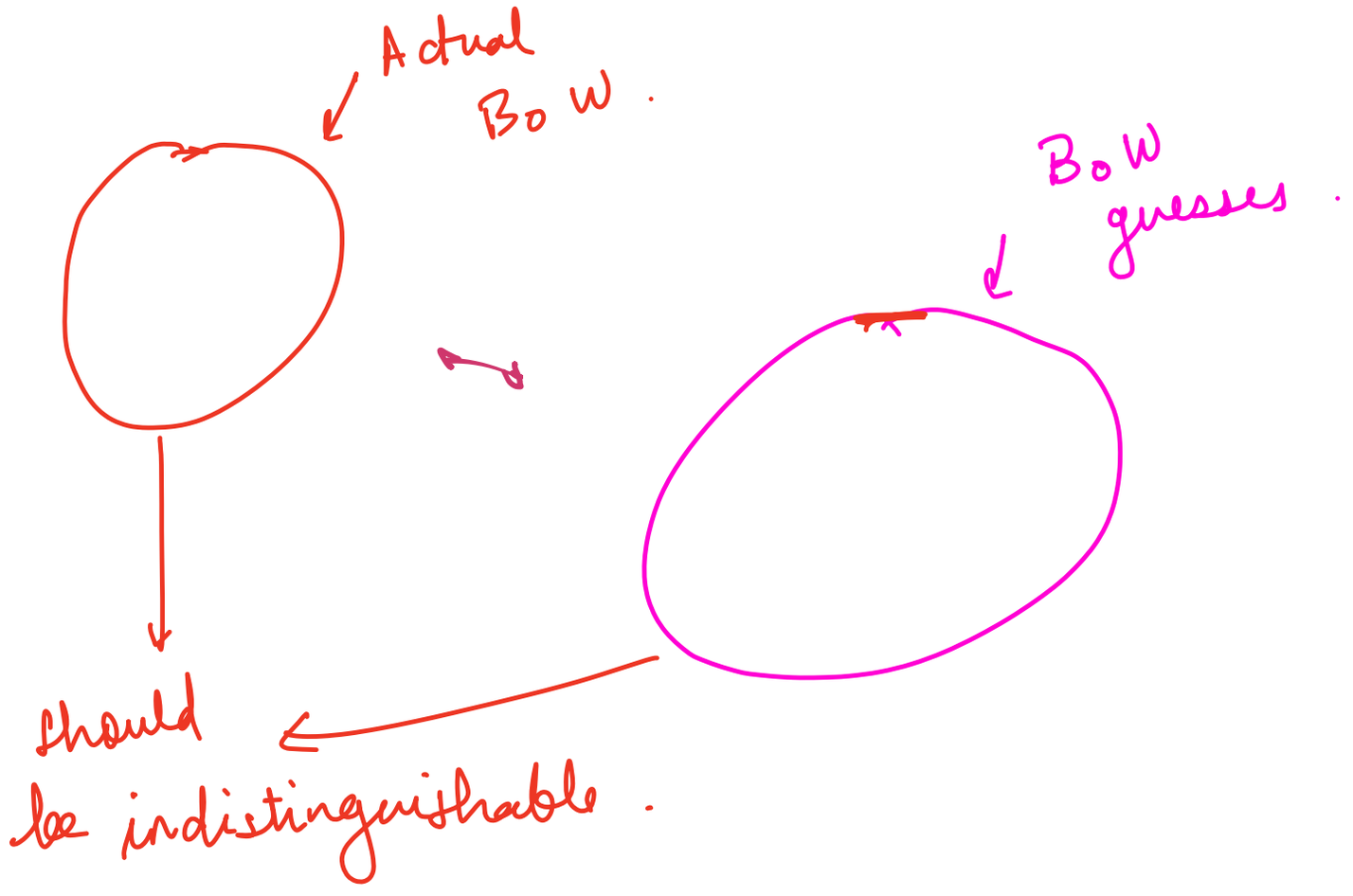
(2) A.P.D


if - else block
if
(condition)


(Main) Topic
Sub - topic 1
sent 1
sent 2
sent 3
Sub - topic 2
sent 1
sent 2
Sub topic 3
sent 1
sent 2
Sent 3.


3. Pick another ball from the new ground and you have a single word
2. Based on the ball you pick, you're sent to another ground "Beta"
1. Pick a ball from the ground "Theta"
Cats
Loyal
Dogs
Evil
Flying
Olympics
Players
AI
Sport
Animals
Tech
Organize ground Theta
Alpha
Organize ground Beta
Eta
Cats
AI
Dogs
Evil
Cats
Document

Now how to do topic modeling?

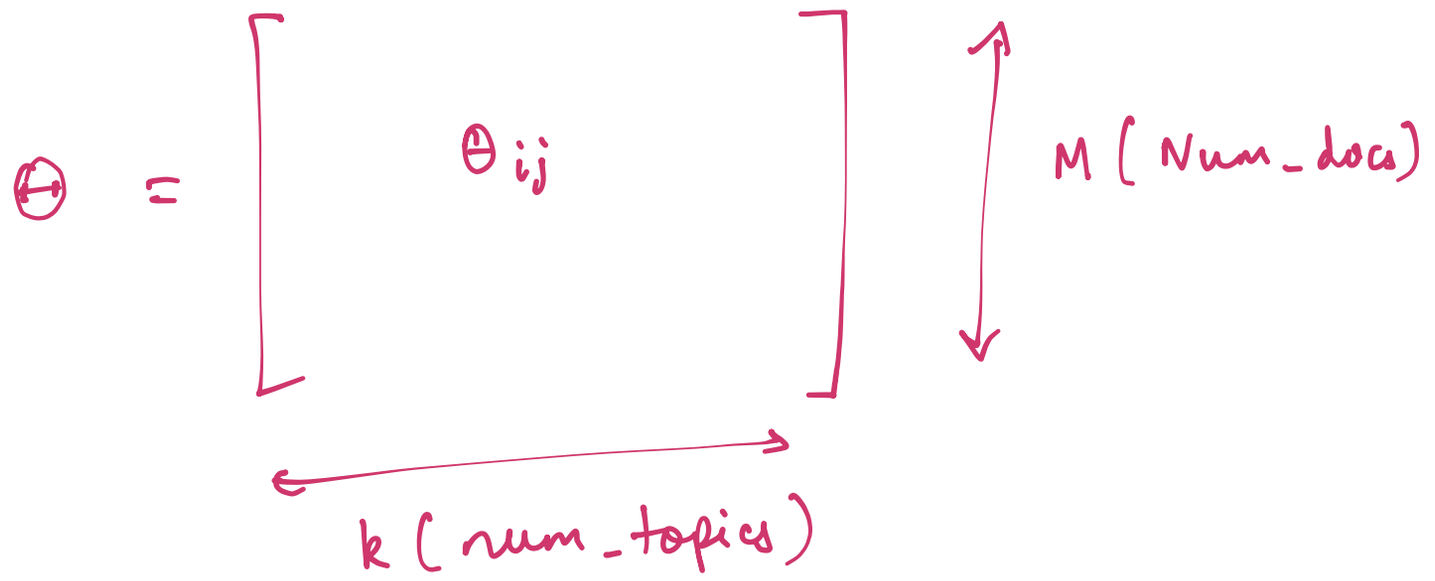


Loss function: KL - Divergence.

```
[(0,
  '0.020*"great" + 0.014*"good" + 0.012*"sound" + 0.012*"strings" + 0.011*"quality" + 0.011*"price"
+ 0.010*"would" + 0.009*"time" + 0.008*"use" + 0.007*"well"'),
```

```
Not much to write about here, but it does exactly what it's supposed to. filters out the pop
sounds. now my recordings are much more crisp. it is one of the lowest prices pop filters on amazon
so might as well buy it, they honestly work the same despite their pricing,
```

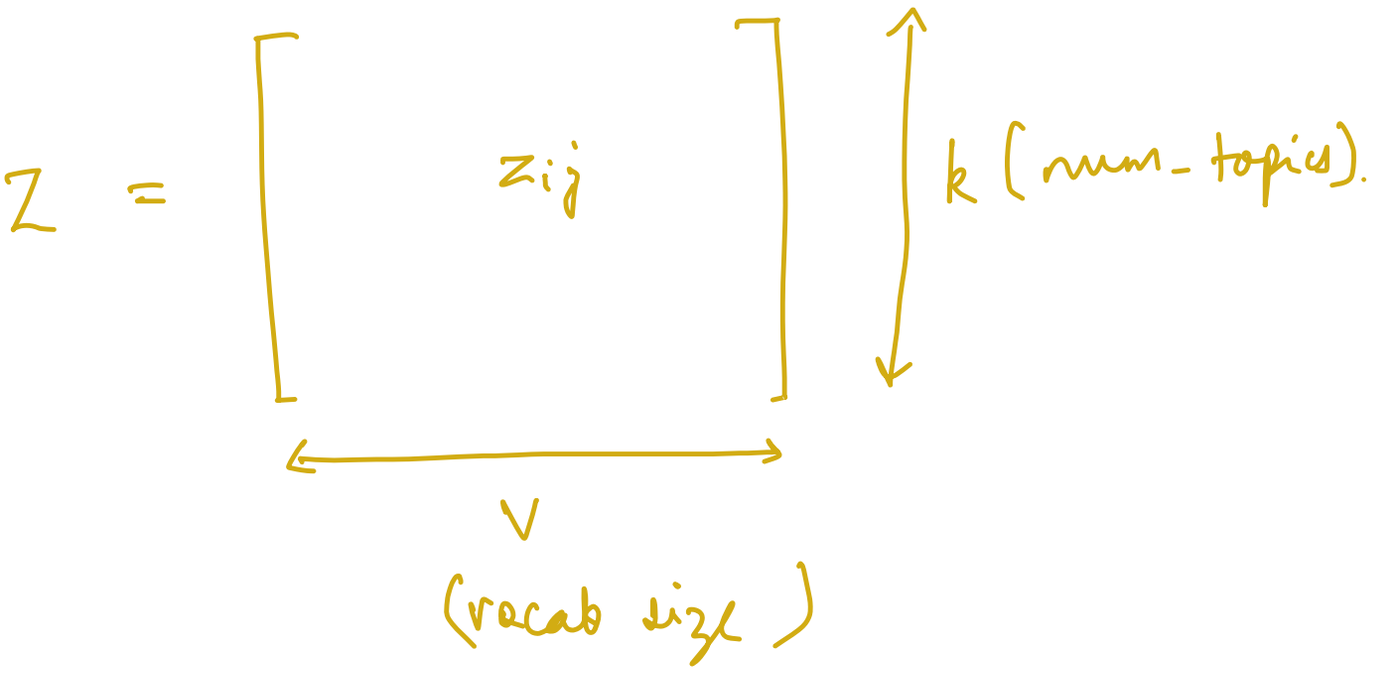
Matrix 1 :



Θ_{ij} → Probability that topic j is important for document M

$$P(t \mid d)$$

Matrix 2 :



z_{ij} → Probability that word j is relevant to topic i.

$P(w \mid t)$

$$\rightarrow P(w \mid d) = \frac{P(w \cap d)}{P(d)}$$

$$P(w \mid t)$$

$$P(w \cap d \cap t)$$

$$= P(w \cap d \mid t) \cdot P(t)$$

$$P(t \mid d)$$

$$P(w \mid t) = \frac{P(w \cap t)}{P(t)}$$

$$P(t \mid d) = \frac{P(t \cap d)}{P(d)}$$

$$P(w \mid t) \cdot P(d)$$

$$P(w \mid t)$$

J.P:

$$P(w, d \mid t)\, P(t)$$

$$= P(w \mid t) \cdot P(t)$$

$$P(d)$$