https://vvtesh.sarahah.com/

Information Retrieval

Venkatesh Vinayakarao

Term: Aug – Sep, 2019

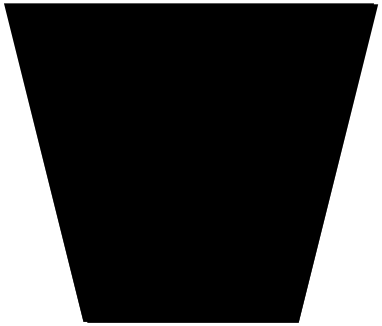
Chennai Mathematical Institute

So much of life, it seems to me, is determined by pure randomness.
– Sidney Poitier.

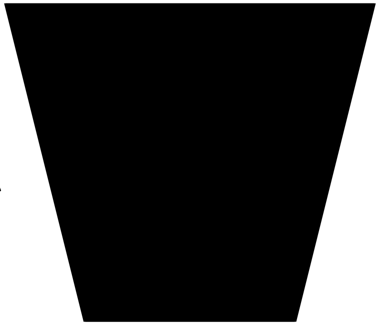
The Intuition

- Which bucket is most likely to lead to a

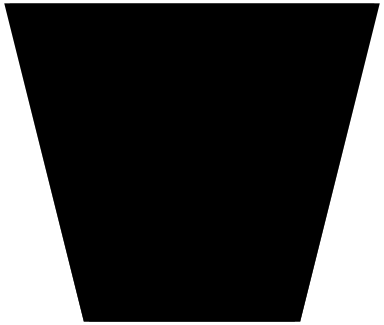
Black Ball?



OR



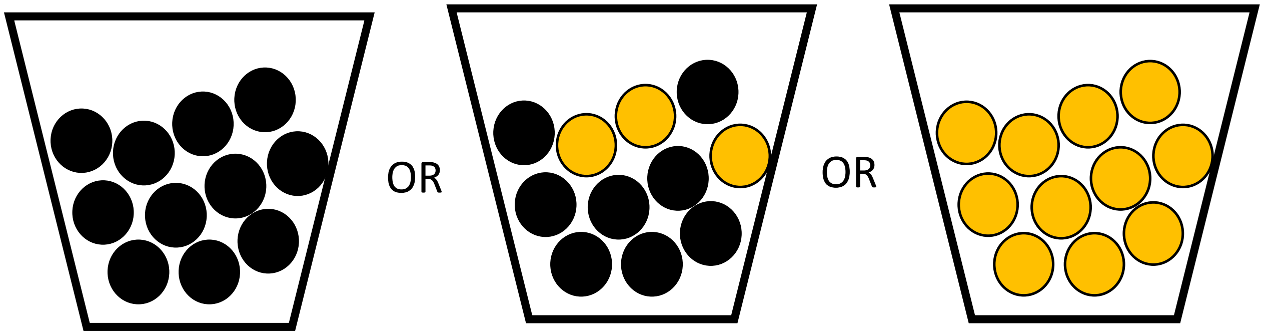
OR



The Intuition

- Which bucket is most likely to lead to a

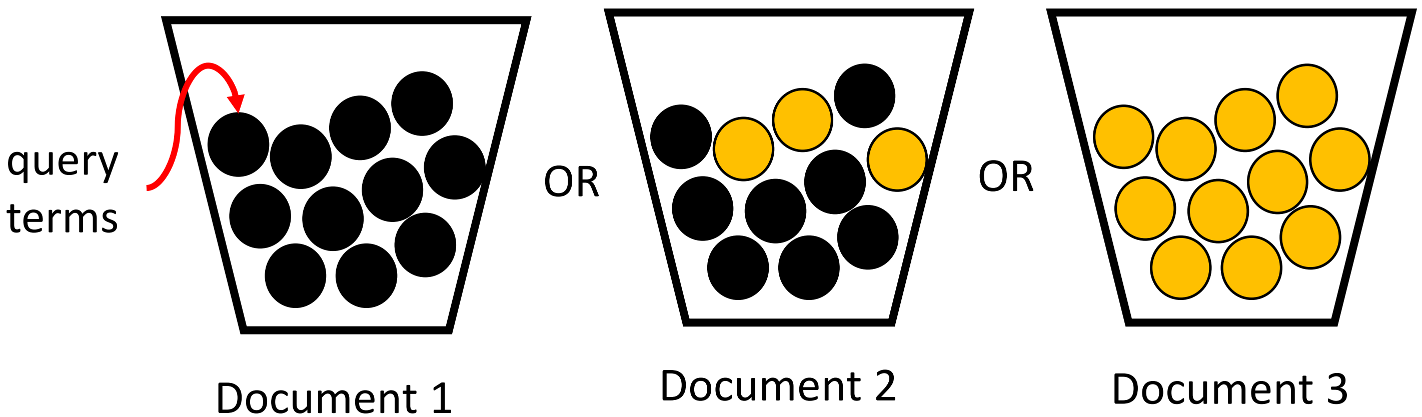
Black Ball?



The Intuition

- Which document is most likely to lead to a

query?



Can be modeled using
a discriminative model (P(D|R=1,Q)) or a generative model (P(Q|R=1,D))

A good query…

Words that are most likely to appear in a relevant document

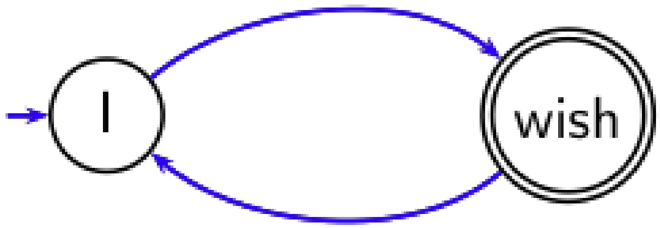
Another Way to Rank

Rank documents based on the probability of the model generating the query: $P(q|M_d)$

M_d is the model of the document which generates the query

A Model to Generate a String

- What strings can this model generate?



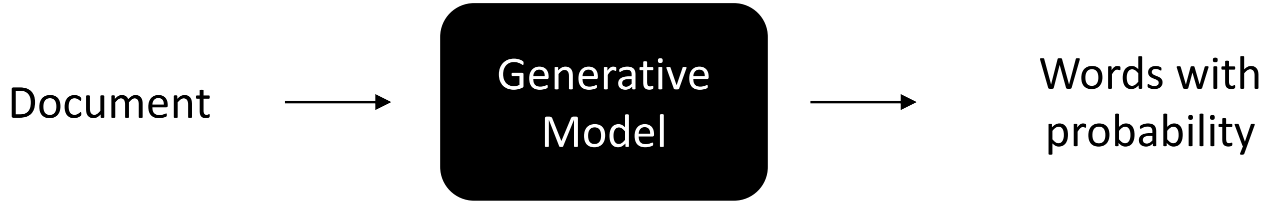
This is a Finite Automaton that can generate:

I wish

I wish I wish

I wish I wish I wish …

Key Idea



Example

- P(STOP|t) = 0.2 i.e., probability that our automaton stops after encountering any word is 0.2.
- You are given with the probabilities of words appearing in the query.
- What is probability of "frog said that toad likes frog" being the query?
  - Answer: (0.01 x 0.03 x 0.04 x 0.01 x 0.02 x 0.01) x (0.8^5 x 0.2) = 0.000000000001573

Word	Probability
the	0.2
a	0.1
frog	0.01
toad	0.01
said	0.03
likes	0.02
that	0.04

Note: It is much easier to take log and add instead of multiplying these numbers.

Comparing Models

- Say we have two models:

s	frog	said	that	toad	likes	that	dog
M_1	0.01	0.03	0.04	0.01	0.02	0.04	0.005
M_2	0.0002	0.03	0.04	0.0001	0.04	0.04	0.01

$P(s|M_1) = 0.000000000000048$
$P(s|M_2) = 0.000000000000000384$

- Model 1 seems to have higher probability of generating the string.
- Model 1 will match a document containing these terms better to this query.

A Simple Language Model

- Simple Model

$$P(t_1 t_2 t_3 t_4) = P(t_1) P(t_2|t_1) P(t_3|t_1 t_2) P(t_4|t_1 t_2 t_3).$$

- Even simpler – The Unigram Language Model

$$P_{\text{uni}}(t_1 t_2 t_3 t_4) = P(t_1) P(t_2) P(t_3) P(t_4)$$

- The Bigram Model

$$P_{\text{bi}}(t_1 t_2 t_3 t_4) = P(t_1) P(t_2|t_1) P(t_3|t_2) P(t_4|t_3)$$

Document as a Bag of Words

- We rank documents based on the probability of that document to generate the given query.

$$P(d|q) = P(q|d)P(d)/P(q).$$

- If documents are bag of words,

$$P(q|M_d) = K_q \prod_{t \in V} P(t|M_d)^{\mathrm{tf}_{t,d}}$$

- Assume p(d) is uniform. Hence can be dropped.
- P(q) and K_q does not affect ranking. Hence can be dropped.

Missing Terms

- What happens to P(d|q) if a query term does not appear in the document?

Clue: We approximate P(d|q) to document ranking by computing $\prod_{t \in V} P(t|M_d)^{\mathrm{tf}_{t,d}}$

Missing Terms & Smoothening

- Instead of zero, use $P(t|M_c)$
- M_c is the model over the entire collection.
- **Smoothen**: We can also mix document and collection probabilities using a linear interpolation co-efficient λ.

$$P(d|q) \propto P(d) \prod_{t \in q} ((1-\lambda)P(t|M_c) + \lambda P(t|M_d))$$

Wake Up Quiz

- Is the following same as the document length?
  - True or False?

$$\sum_{1 \leq i \leq M} tf(ti, d)$$

M is the term vocabulary size.
Function tf(t,d) is term
frequency of a term in a
document.

Example

- Suppose:
  - d1: Tada is a city between Chennai and Tirupati
  - d2: Tada has few restaurants but no good malls
- Use the smoothened MLE unigram language model to rank these documents for the query “Tada City”. Assume $\lambda = 0.5$.

Example

$$\prod_{t \in q} ((1-\lambda)P(t|M_c) + \lambda P(t|M_d))$$

- Suppose:
  - d1: Tada is a city between Chennai and Tirupati
  - d2: Tada has few restaurants but no good malls
- Use the smoothened MLE unigram language model to rank these documents for the query "Tada City". Assume *λ = 0.5.*
  - $P(q|d1) = [(1/8 + 2/16)/2] \cdot [(1/8 + 1/16)/2] = 1/8 \cdot 3/32 = 3/256$
  - $P(q|d2) = [(1/8 + 2/16)/2] \cdot [(0/8 + 1/16)/2] = 1/8 \cdot 1/32 = 1/256$
  - *Ranking: d1 > d2*

Thank You