COEN 169

Statistical Language Modeling

Yi Fang

Department of Computer Engineering

Santa Clara University

Statistical language modeling

Introduction to statistical language modeling

Language modeling for information retrieval

Query-likelihood Retrieval Model

Smoothing

Document priors

What is a language model?

"A statistical language model assigns a probability to a sequence of words by means of a probability distribution"

--Wikipedia

What is a language model?

- To understand what a language model is, we have to understand what a probability distribution is
- To understand what a probability distribution is, we have to understand what a discrete random variable is

What is a discrete random variable?

- Let A denote a discrete random variable
- A is a discrete random variable if:
  - A describes an event with a finite number of possible outcomes (this property makes the random variable discrete)
  - A describes an event whose outcome has some degree of uncertainty (this property makes the variable random)

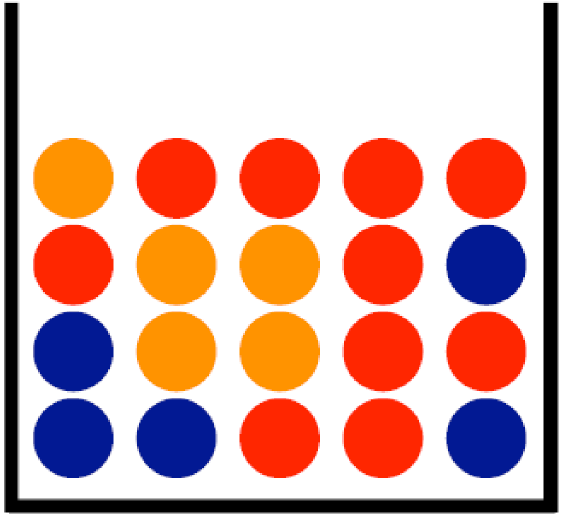
Discrete Random Variables

examples

- A = it will rain tomorrow
- A = the coin-flip will show heads
- A = you will win the lottery in your lifetime
- A = the 2023 US president will be female
- A = you have the flu
- A = flip a dice

What is a probability distribution?

- A probability distribution gives the probability of each possible outcome of a random variable
- P(RED) = probability that you will reach into this bag and pull out a red ball
- P(BLUE) = probability that you will reach into this bag and pull out a blue ball
- P(ORANGE) = probability that you will reach into this bag and pull out an orange ball



What is a probability distribution?

- For it to be a probability distribution, two conditions must be satisfied:
  - the probability assigned to each possible outcome must be between 0 and 1 (inclusive)
  - the sum of probabilities across outcome must be 1

$$0 \leq P(\text{RED}) \leq 1$$

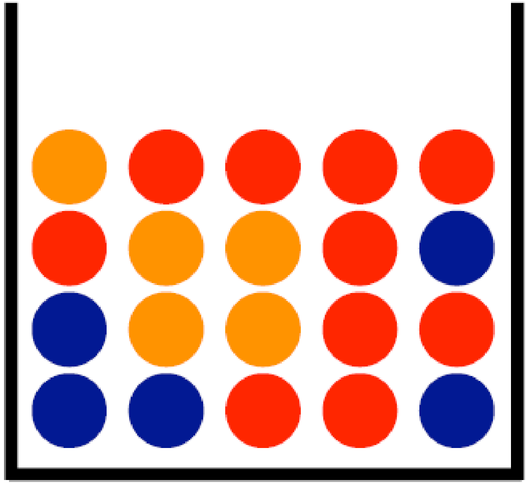
$$0 \leq P(\text{BLUE}) \leq 1$$

$$0 \leq P(\text{ORANGE}) \leq 1$$

$$P(\text{RED}) + P(\text{BLUE}) + P(\text{ORANGE}) = 1$$

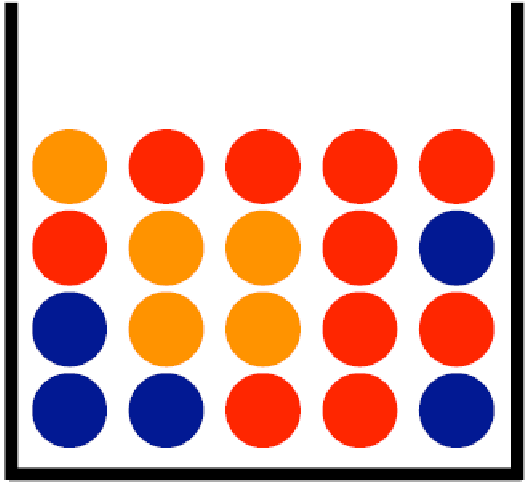
Estimating a Probability Distribution

- Let’s estimate these probabilities based on what we know about the contents of the bag
- P(RED) = ?
- P(BLUE) = ?
- P(ORANGE) = ?



Estimating a Probability Distribution

- Let's estimate these probabilities based on what we know about the contents of the bag
- P(RED) = 10/20 = 0.5
- P(BLUE) = 5/20 = 0.25
- P(ORANGE) = 5/20 = 0.25
- P(RED) + P(BLUE) + P(ORANGE) = 1.0



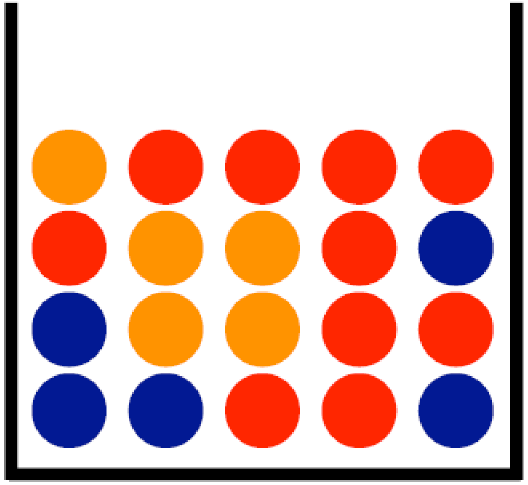
What can we do with a probability distribution?

- We can assign probabilities to different outcomes
- I reach into the bag and pull out an orange ball. What is the probability of that happening?
- I reach into the bag and pull out two balls: one red, one blue. What is the probability of that happening?
- What about three orange balls?

P(RED) = 0.5

P(BLUE) = 0.25

P(ORANGE) = 0.25



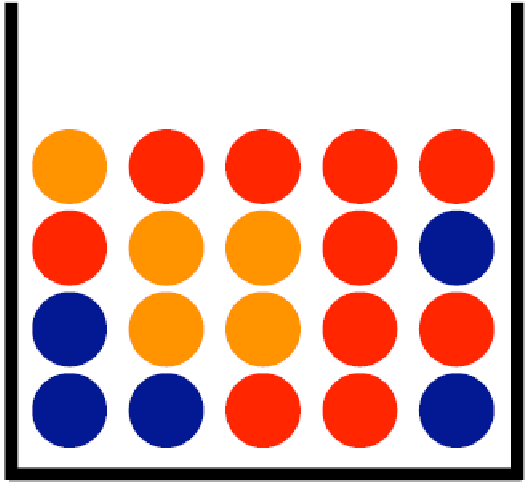
What can we do with a probability distribution?

- If we assume that each outcome is independent of previous outcomes, then the probability of a sequence of outcomes is calculated by multiplying the individual probabilities
- Note: we're assuming that when you take out a ball, you put it back in the bag before taking another one out

P(RED) = 0.5

P(BLUE) = 0.25

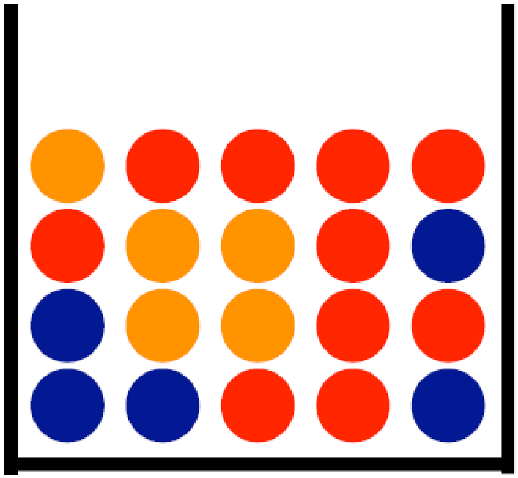
P(ORANGE) = 0.25



What can we do with a probability distribution?

- P(🔵) = 0.25
- P(🔴) = 0.5
- P(🟠🟠🟠) = 0.25 x 0.25 x 0.25
- P(🟠🔵🟠) = 0.25 x 0.25 x 0.25
- P(🟠🔴🟠) = 0.25 x 0.50 x 0.25
- P(🟠🔴🟠🔴) = 0.25 x 0.50 x 0.25 x 0.50

P(RED) = 0.5
P(BLUE) = 0.25
P(ORANGE) = 0.25

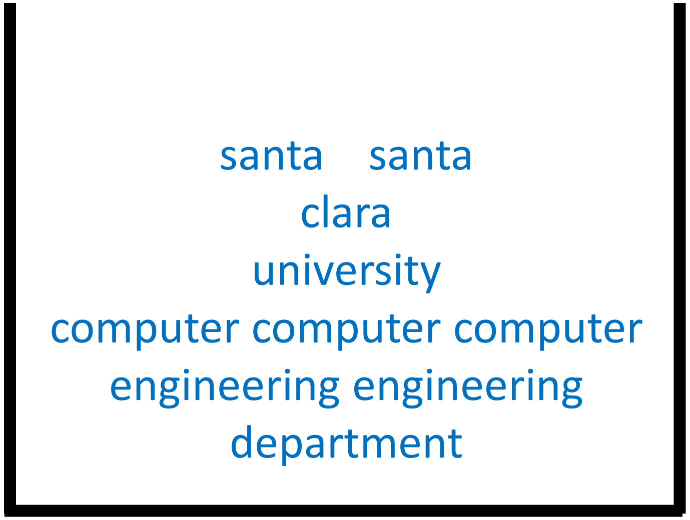


Language modeling

- We want to assign a probability to a particular sequence of words
- We want to use a “bag of something” to do it (similar to our bag of colored balls)
- What should be contained in the bag? Sequences of words? Individual words?

Unigram Language Model

- Defines a probability distribution over individual words
  - P(santa) = 2/10
  - P(clara) = 1/10
  - P(university) = 1/10
  - P(computer) = 3/10
  - P(engineering) = 2/10
  - P(department) = 1/10



santa santa
clara
university
computer computer computer
engineering engineering
department

Unigram Language Model

- It is called a unigram language model because we estimate (and predict) the likelihood of each word independent of any other word
- Assumes that words are independent!
  - The probability of seeing “clara” is the same, even if the preceding word is “santa”
- Other language models take context into account
- Those work better for applications like speech recognition or automatic language translation
- Unigram models work well for information retrieval

Unigram Language Model

- Sequences of words can be assigned a probability by multiplying their individual probabilities:

P(santa clara university) =

P(santa) x P(clara) x P(university) =

(2/10) x (1/10) x (1/10) = 0.002

P(computer engineering department) =

P(computer) x P(engineering) x P(department) =

(3/10) x (2/10) x (1/10) = 0.006

Unigram Language Model

- There are two important steps in language modeling
  - estimation: observing text and estimating the probability of each word
  - prediction: using the language model to assign a probability to a span of text

Unigram Language Model

- Any span of text can be used to estimate a language model
- And, given a language model, we can assign a probability to any span of text
  - a word
  - a sentence
  - a document
  - a corpus
  - the entire web

Unigram Language Model Estimation

- General estimation approach:
  - tokenize/split the text into terms
  - count the total number of term occurrences (*N*)
  - count the number of occurrences of each term (tf_t)
  - assign term *t* a probability equal to

$$P_t = \frac{tf_t}{N}$$

IMDB Corpus

language model estimation (top 20 terms)

term	tf	N	P(term)	term	tf	N	P(term)
the	1586358	36989629	0.0429	year	250151	36989629	0.0068
a	854437	36989629	0.0231	he	242508	36989629	0.0066
and	822091	36989629	0.0222	movie	241551	36989629	0.0065
to	804137	36989629	0.0217	her	240448	36989629	0.0065
of	657059	36989629	0.0178	artist	236286	36989629	0.0064
in	472059	36989629	0.0128	character	234754	36989629	0.0063
is	395968	36989629	0.0107	cast	234202	36989629	0.0063
i	390282	36989629	0.0106	plot	234189	36989629	0.0063
his	328877	36989629	0.0089	for	207319	36989629	0.0056
with	253153	36989629	0.0068	that	197723	36989629	0.0053

IMDB Corpus

language model estimation (top 20 terms)

term	tf	N	P(term)	term	tf	N	P(term)
the	1586358	36989629	0.0429	year	250151	36989629	0.0068
a	854437	36989629	0.0231	he	242508	36989629	0.0066
and	822091	36989629	0.0222	movie	241551	36989629	0.0065
to	804137	36989629	0.0217	her	240448	36989629	0.0065
of	657059	36989629	0.0178	artist	236286	36989629	0.0064
in	472059	36989629	0.0128	character	234754	36989629	0.0063
is	395968	36989629	0.0107	cast	234202	36989629	0.0063
i	390282	36989629	0.0106	plot	234189	36989629	0.0063
his	328877	36989629	0.0089	for	207319	36989629	0.0056
with	253153	36989629	0.0068	that	197723	36989629	0.0053

- What is the probability associated with “artist of the year”?

IMDB Corpus

language model estimation (top 20 terms)

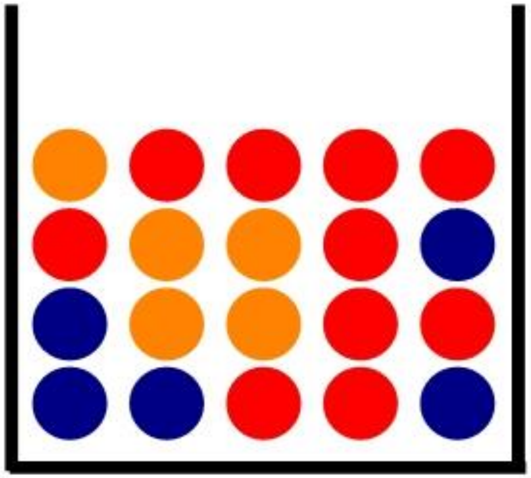
term	tf	N	P(term)	term	tf	N	P(term)
the	1586358	36989629	0.0429	year	250151	36989629	0.0068
a	854437	36989629	0.0231	he	242508	36989629	0.0066
and	822091	36989629	0.0222	movie	241551	36989629	0.0065
to	804137	36989629	0.0217	her	240448	36989629	0.0065
of	657059	36989629	0.0178	artist	236286	36989629	0.0064
in	472059	36989629	0.0128	character	234754	36989629	0.0063
is	395968	36989629	0.0107	cast	234202	36989629	0.0063
i	390282	36989629	0.0106	plot	234189	36989629	0.0063
his	328877	36989629	0.0089	for	207319	36989629	0.0056
with	253153	36989629	0.0068	that	197723	36989629	0.0053

- What is more probable: "artist of the year" or "movie of the year?"

Language Models

- A language model is a probability distribution defined over a particular vocabulary

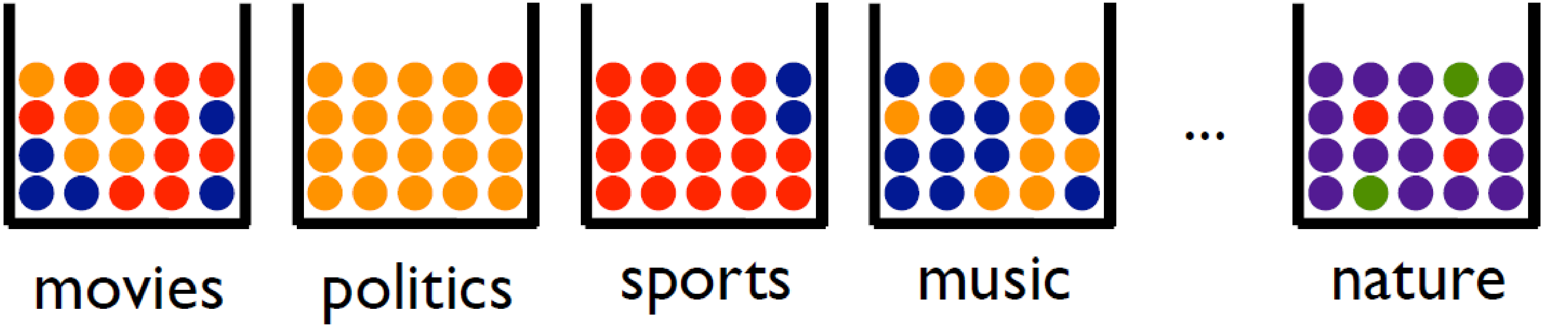
- In this analogy, each color represents a vocabulary term and each ball represents a term occurrence in the text used to estimate the language model



P(RED) = 0.5
P(BLUE) = 0.25
P(ORANGE) = 0.25

Topic Models

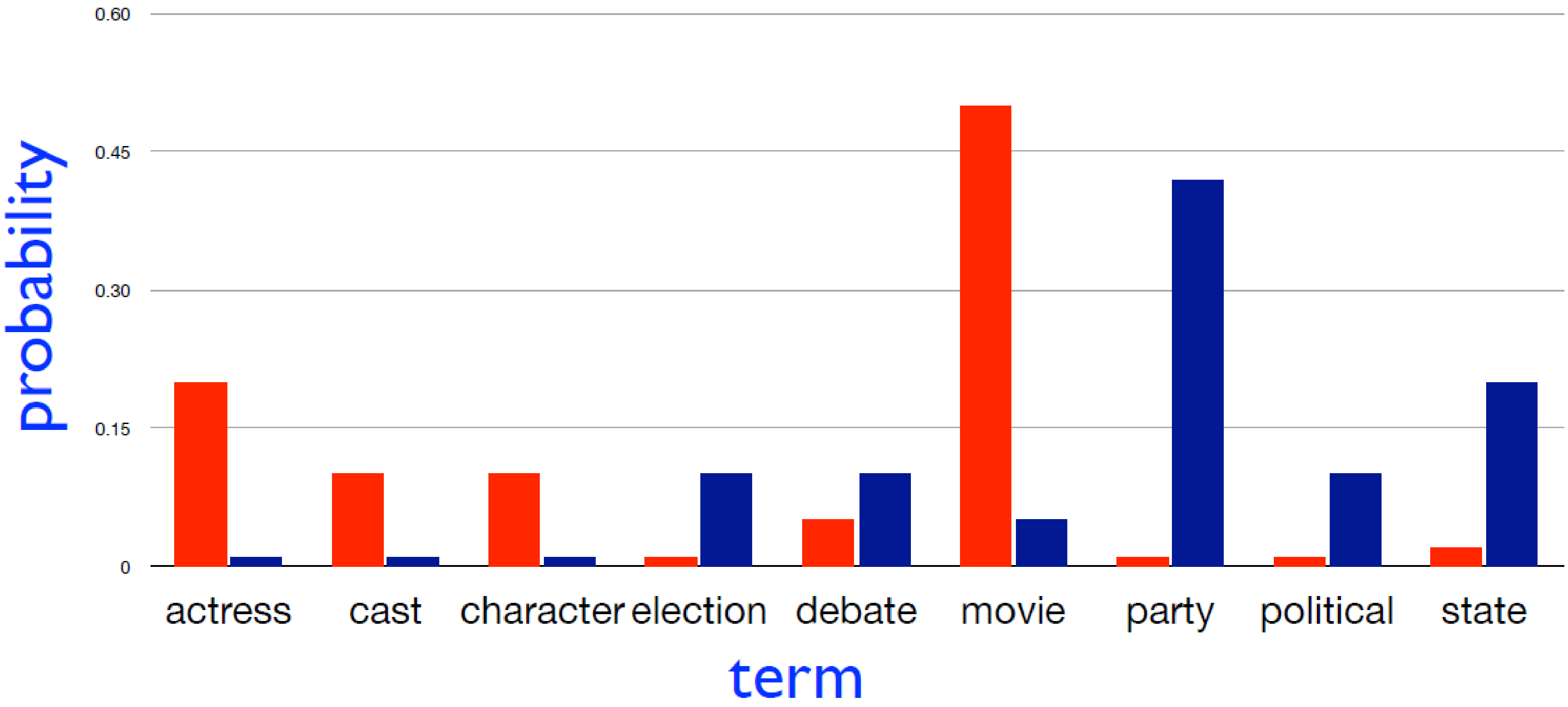
- We can think of a topic as being defined by a language model
- A high-probability of seeing certain words and a low-probability of seeing others



movies	politics	sports	music	nature
P(RED) = 0.5	P(RED) = 0.05	P(RED) = 0.90	P(RED) = 0.00	P(RED) = 0.10
P(BLUE) = 0.25	P(BLUE) = 0.00	P(BLUE) = 0.10	P(BLUE) = 0.50	P(BLUE) = 0.80
P(ORANGE) = 0.25	P(ORANGE) = 0.95	P(ORANGE) = 0.00	P(ORANGE) = 0.50	P(ORANGE) = 0.10

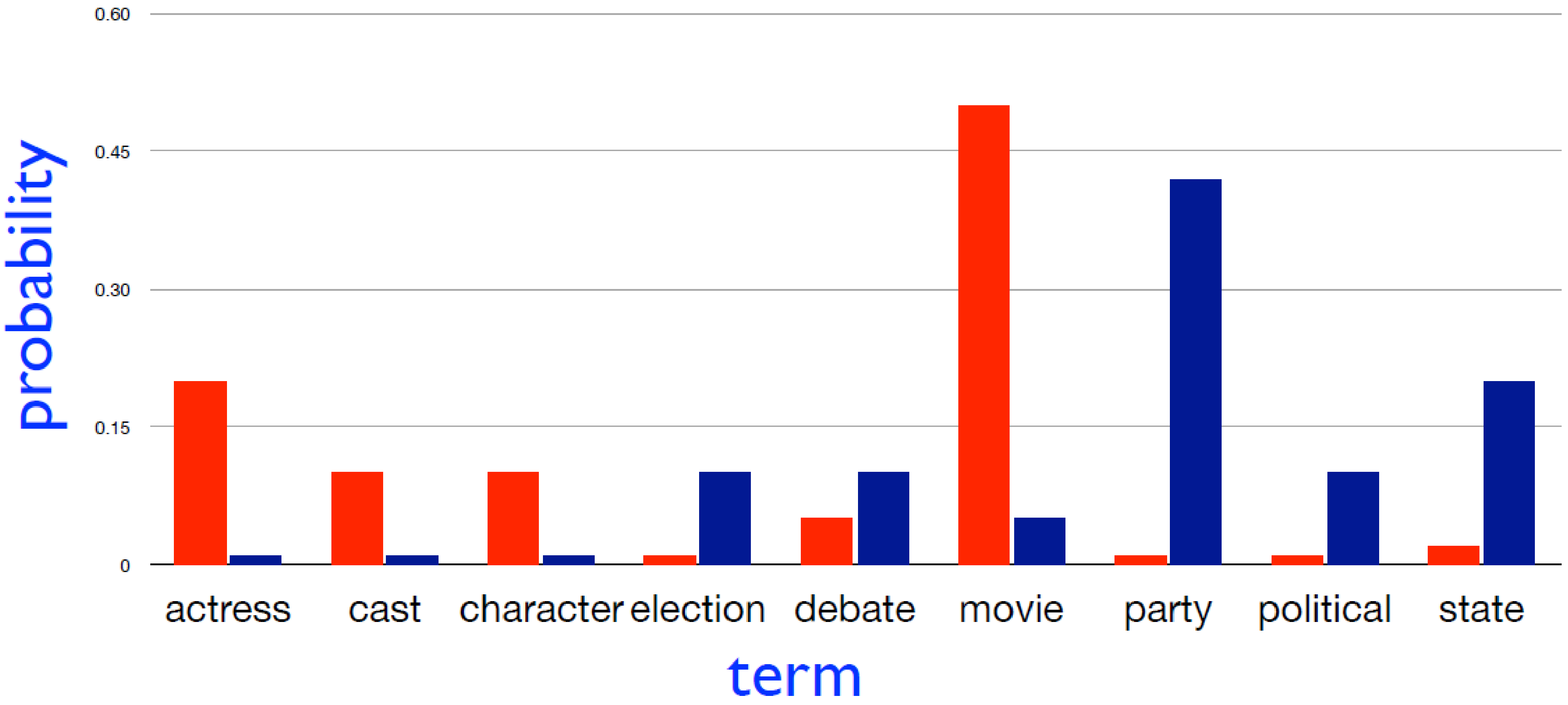
Topic Models

??? vs. ???



Topic Models

movie vs. politics



Topical Relevance

- Many factors affect whether a document satisfies a particular user's information need

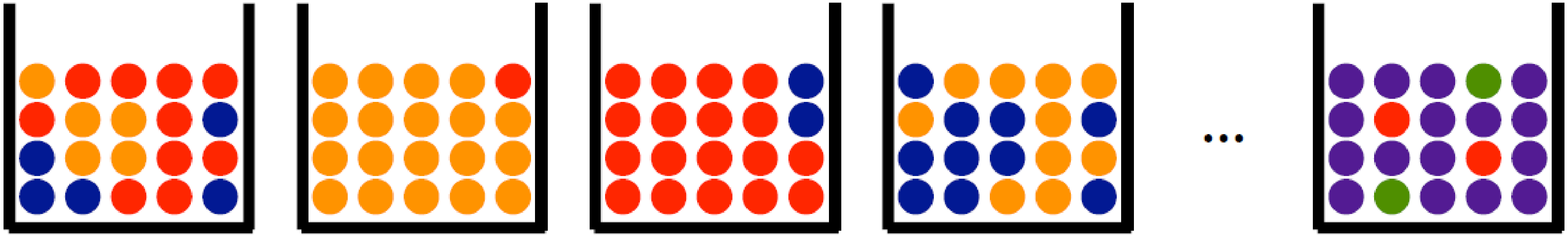
- Topicality, novelty, freshness, authority, formatting, reading level, assumed level of expertise, etc.

- Topical relevance: the document is on the same topic as the query

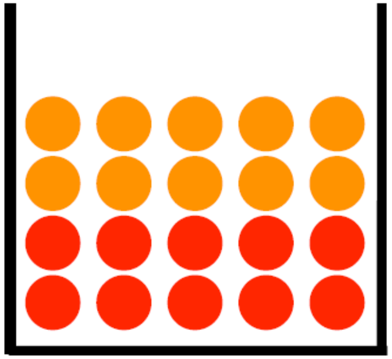
- Remember, our goal right now is to predict topical relevance

Document Language Models

- The topic (or topics) discussed in a particular document can be captured by its language model



movies	politics	sports	music	nature
P(RED) = 0.5	P(RED) = 0.05	P(RED) = 0.90	P(RED) = 0.00	P(RED) = 0.10
P(BLUE) = 0.25	P(BLUE) = 0.00	P(BLUE) = 0.10	P(BLUE) = 0.50	P(BLUE) = 0.80
P(ORANGE) = 0.25	P(ORANGE) = 0.95	P(ORANGE) = 0.00	P(ORANGE) = 0.50	P(ORANGE) = 0.10



What is this document about?

Document Language Models

- Estimating a document’s language model:
  1. tokenize/split the document text into terms
  2. count the number of times each term occurs ($tf_{t,D}$)
  3. count the total number of term occurrences (N_D)
  4. assign term *t* a probability equal to:

$$\frac{tf_{t,D}}{N_D}$$

Document Language Models

- The language model estimated from document *D* is sometimes denoted as:

$$\theta_D$$

- The probability given to term *t* by the language model estimated from document *D* is sometimes denoted as:

$$P(t|D) = P(t|\theta_D) = \frac{tf_{t,D}}{N_D}$$

Document Language Models

- Movie: Rocky (1976)

- Plot:

  Rocky Balboa is a struggling boxer trying to make the big time. Working in a meat factory in Philadelphia for a pittance, he also earns extra cash as a debt collector. When heavyweight champion Apollo Creed visits Philadelphia, his managers want to set up an exhibition match between Creed and a struggling boxer, touting the fight as a chance for a "nobody" to become a "somebody". The match is supposed to be easily won by Creed, but someone forgot to tell Rocky, who sees this as his only shot at the big time. Rocky Balboa is a small-time boxer who lives in an apartment in Philadelphia, Pennsylvania, and his career has so far not gotten off the canvas. Rocky earns a living by collecting debts for a loan shark named Gazzo, but Gazzo doesn't think Rocky has the viciousness it takes to beat up deadbeats. Rocky still boxes every once in a while to keep his boxing skills sharp, and his ex-trainer, Mickey, believes he could've made it to the top if he was willing to work for it. Rocky, goes to a pet store that sells pet supplies, and this is where he meets a young woman named Adrian, who is extremely shy, with no ability to talk to men. Rocky befriends her. Adrain later surprised Rocky with a dog from the pet shop that Rocky had befriended. Adrian's brother Paulie, who works for a meat packing company, is thrilled that someone has become interested in Adrian, and Adrian spends Thanksgiving with Rocky. Later, they go to Rocky's apartment, where Adrian explains that she has never been in a man's apartment before. Rocky sets her mind at ease, and they become lovers. Current world heavyweight boxing champion Apollo Creed comes up with the idea of giving an unknown a shot at the title. Apollo checks out the Philadelphia boxing scene, and chooses Rocky. Fight promoter Jergens gets things in gear, and Rocky starts training with Mickey. After a lot of training, Rocky is ready for the match, and he wants to prove that he can go the distance with Apollo. The 'Italian Stallion', Rocky Balboa, is an aspiring boxer in downtown Philadelphia. His one chance to make a better life for himself is through his boxing and Adrian, a girl who works in the local pet store. Through a publicity stunt, Rocky is set up to fight Apollo Creed, the current heavyweight champion who is already set to win. But Rocky really needs to triumph, against all the odds...

Document Language Models

language model estimation (top 20 terms)

term	$tf_{t,D}$	N_D	*P(term|D)*	*term*	$tf_{t,D}$	N_D	*P(term|D)*
a	22	420	0.05238	creed	5	420	0.01190
rocky	19	420	0.04524	philadelphia	5	420	0.01190
to	18	420	0.04286	has	4	420	0.00952
the	17	420	0.04048	pet	4	420	0.00952
is	11	420	0.02619	boxing	4	420	0.00952
and	10	420	0.02381	up	4	420	0.00952
in	10	420	0.02381	an	4	420	0.00952
for	7	420	0.01667	boxer	4	420	0.00952
his	7	420	0.01667	s	3	420	0.00714
he	6	420	0.01429	balboa	3	420	0.00714

Document Language Models

- Suppose we have a document *D*, with language model θ_D
- We can use this language model to determine the probability of a particular sequence of text such as a query
- How? We multiple the probability associated with each term in the query!

Document Language Models

language model estimation (top 20 terms)

term	$tf_{t,D}$	N_D	$P(term \mid D)$	*term*	$tf_{t,D}$	N_D	$P(term \mid D)$
a	22	420	0.05238	creed	5	420	0.01190
rocky	19	420	0.04524	philadelphia	5	420	0.01190
to	18	420	0.04286	has	4	420	0.00952
the	17	420	0.04048	pet	4	420	0.00952
is	11	420	0.02619	boxing	4	420	0.00952
and	10	420	0.02381	up	4	420	0.00952
in	10	420	0.02381	an	4	420	0.00952
for	7	420	0.01667	boxer	4	420	0.00952
his	7	420	0.01667	s	3	420	0.00714
he	6	420	0.01429	balboa	3	420	0.00714

- What is the probability given by this language model to the sequence of text “rocky is a boxer”?

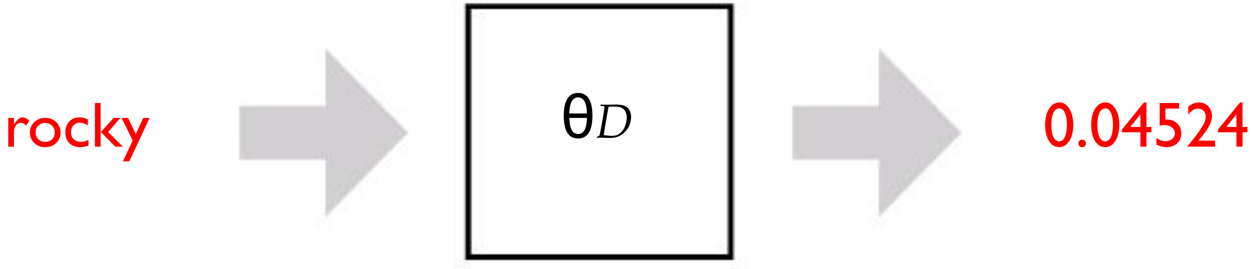
Query-Likelihood Retrieval Model

- Objective: rank documents based on the probability that they are on the same topic as the query
- Solution:
  - Score each document (denoted by *D*) according to the probability given by its language model to the query (denoted by *Q*)
  - Rank documents in descending order of score

$$score(Q, D) = P(Q|\theta_D) = \prod_{i=1}^{n} P(q_i|\theta_D)$$

Query-Likelihood Retrieval Model

- Every document in the collection is associated with a language model
- Let θ_D denote the language model associated with document *D*
- You can think of θ_D as a “black-box”: given a word, it outputs a probability



rocky

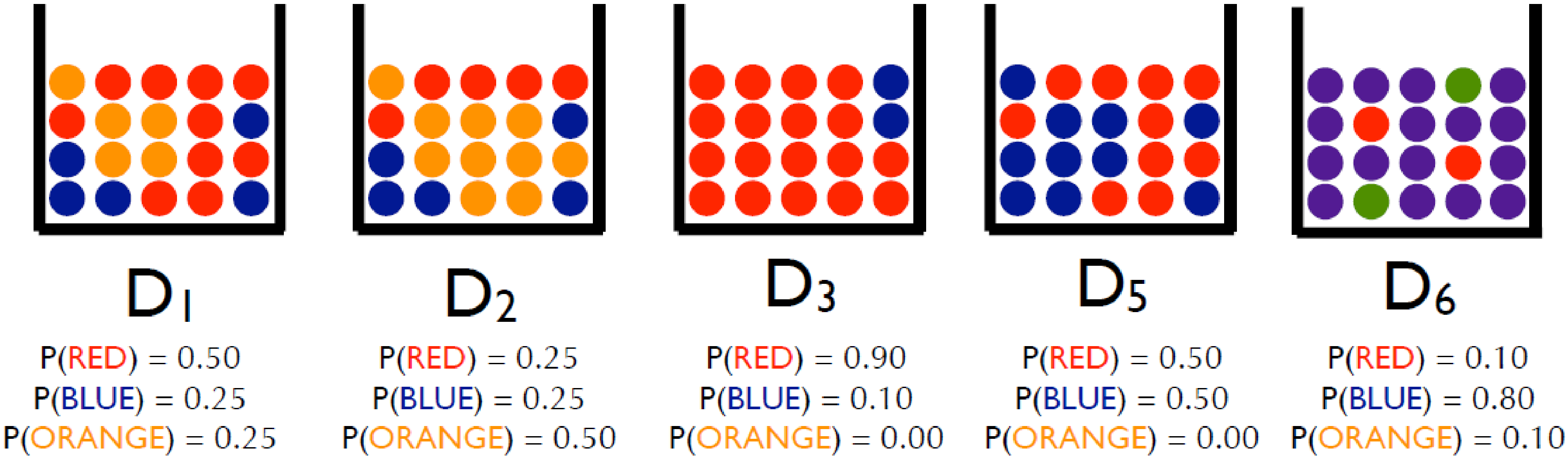
θ_D

0.04524

- Let $P(t|\theta_D)$ denote the probability given by θ_D to term t

Query-Likelihood Model

back to our analogy

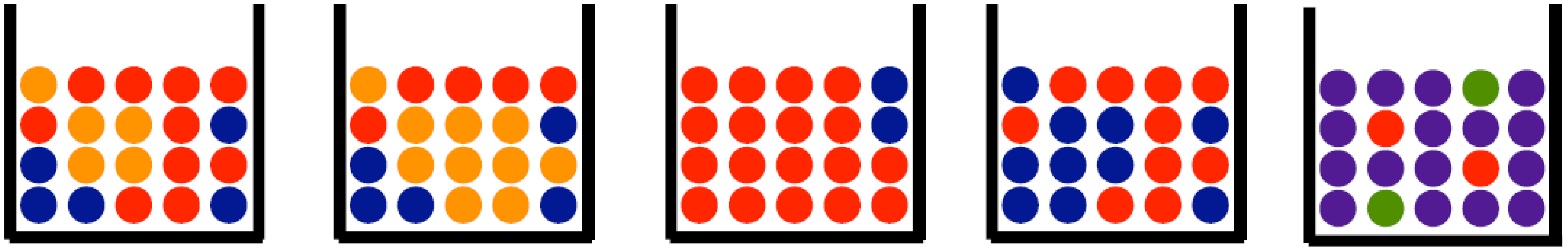


D1	D2	D3	D5	D6
P(RED) = 0.50	P(RED) = 0.25	P(RED) = 0.90	P(RED) = 0.50	P(RED) = 0.10
P(BLUE) = 0.25	P(BLUE) = 0.25	P(BLUE) = 0.10	P(BLUE) = 0.50	P(BLUE) = 0.80
P(ORANGE) = 0.25	P(ORANGE) = 0.50	P(ORANGE) = 0.00	P(ORANGE) = 0.00	P(ORANGE) = 0.10

- Each document is scored according to the probability that it “generated” the query

Query-Likelihood Model

back to our analogy

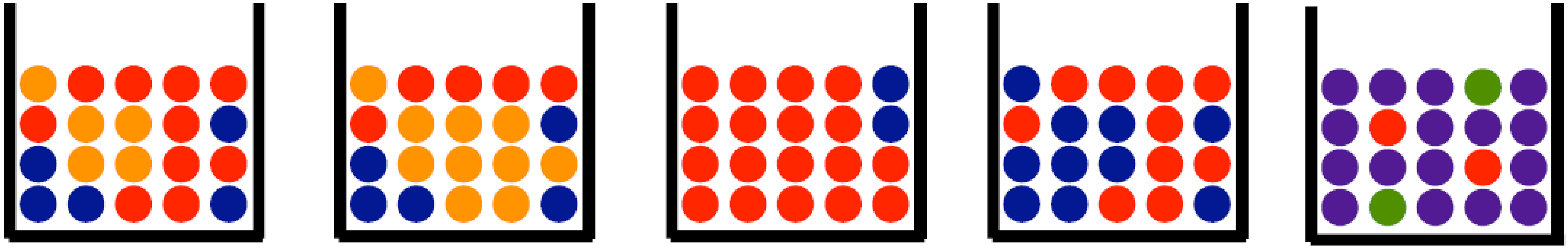


D1	D2	D3	D5	D6
P(RED) = 0.50	P(RED) = 0.25	P(RED) = 0.90	P(RED) = 0.50	P(RED) = 0.10
P(BLUE) = 0.25	P(BLUE) = 0.25	P(BLUE) = 0.10	P(BLUE) = 0.50	P(BLUE) = 0.80
P(ORANGE) = 0.25	P(ORANGE) = 0.50	P(ORANGE) = 0.00	P(ORANGE) = 0.00	P(ORANGE) = 0.10

- Query = ●●●
- Which would be the top-ranked document and what would be its score?

Query-Likelihood Model

back to our analogy

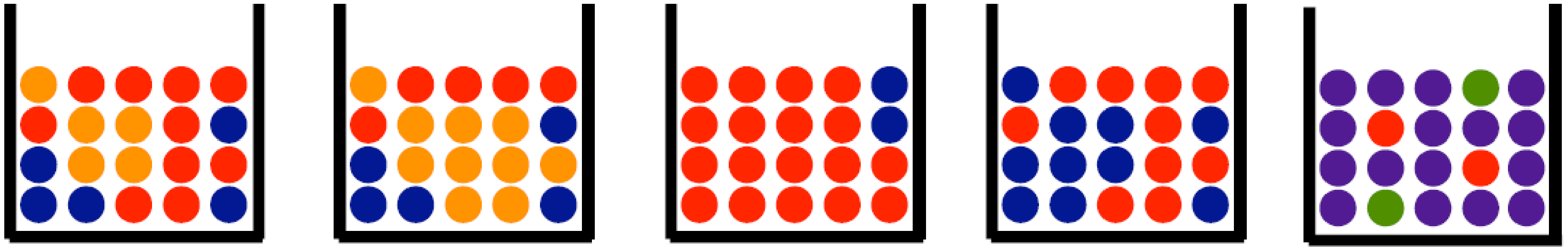


D_1	D_2	D_3	D_5	D_6
P(RED) = 0.50	P(RED) = 0.25	P(RED) = 0.90	P(RED) = 0.50	P(RED) = 0.10
P(BLUE) = 0.25	P(BLUE) = 0.25	P(BLUE) = 0.10	P(BLUE) = 0.50	P(BLUE) = 0.80
P(ORANGE) = 0.25	P(ORANGE) = 0.50	P(ORANGE) = 0.00	P(ORANGE) = 0.00	P(ORANGE) = 0.10

- Query = ● ●
- Which would be the top-ranked document and what would be its score?

Query-Likelihood Model

back to our analogy

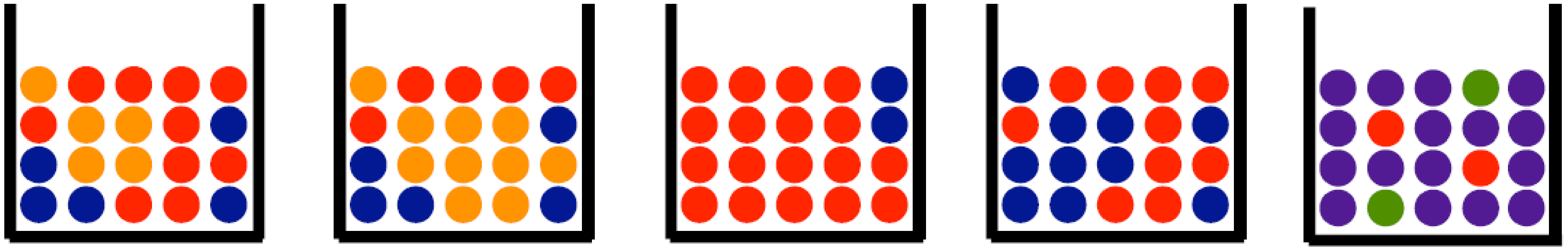


D_1	D_2	D_3	D_5	D_6
P(RED) = 0.50	P(RED) = 0.25	P(RED) = 0.90	P(RED) = 0.50	P(RED) = 0.10
P(BLUE) = 0.25	P(BLUE) = 0.25	P(BLUE) = 0.10	P(BLUE) = 0.50	P(BLUE) = 0.80
P(ORANGE) = 0.25	P(ORANGE) = 0.50	P(ORANGE) = 0.00	P(ORANGE) = 0.00	P(ORANGE) = 0.10

- Query = 🔴 🔵 🔴 🔵 🔴 🔵 🔴 🔵 🔴 🔵
- Which would be the top-ranked document and what would be its score?

Query-Likelihood Model

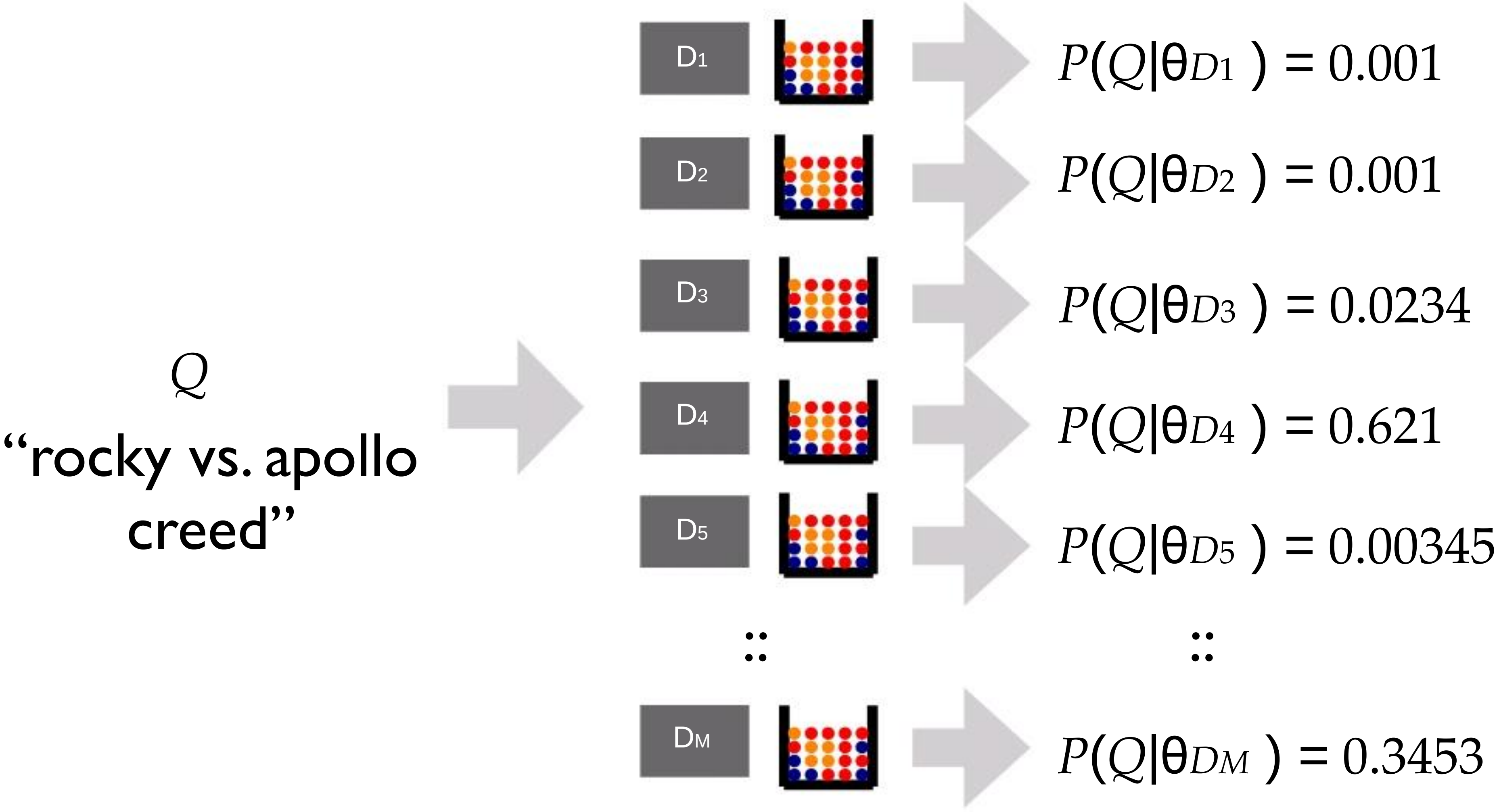
back to our analogy



D1	D2	D3	D5	D6
P(RED) = 0.50	P(RED) = 0.25	P(RED) = 0.90	P(RED) = 0.50	P(RED) = 0.10
P(BLUE) = 0.25	P(BLUE) = 0.25	P(BLUE) = 0.10	P(BLUE) = 0.50	P(BLUE) = 0.80
P(ORANGE) = 0.25	P(ORANGE) = 0.50	P(ORANGE) = 0.00	P(ORANGE) = 0.00	P(ORANGE) = 0.10

- Query = 🔴 🔵 🔴 🔵 🔴 🔵 🔴 🔵 🔴 🟡
- Which would be the top-ranked document and what would be its score?

Query-Likelihood Retrieval Model



Q

“rocky vs. apollo creed”

$P(Q|\theta_{D1}) = 0.001$

$P(Q|\theta_{D2}) = 0.001$

$P(Q|\theta_{D3}) = 0.0234$

$P(Q|\theta_{D4}) = 0.621$

$P(Q|\theta_{D5}) = 0.00345$

::

$P(Q|\theta_{DM}) = 0.3453$

Query-Likelihood Retrieval Model

$$score(Q, D) = P(Q|\theta_D) = \prod_{i=1}^{n} P(q_i|\theta_D)$$

$score$(rocky vs apollo creed, D_5) =

$$P(rocky|\theta_{D_5}) \times P(vs|\theta_{D_5}) \times P(apollo|\theta_{D_5}) \times P(creed|\theta_{D_5})$$

Query-Likelihood Retrieval Model

$$score(Q, D) = P(Q|\theta_D) = \prod_{i=1}^{n} P(q_i|\theta_D)$$

- There is one major issue with this scoring function
- What is it?

Query-Likelihood Retrieval Model

- A document with a single missing query-term will receive a score of zero

Outline

Introduction to language modeling

Language modeling for information retrieval

Query-likelihood Retrieval Model

Smoothing