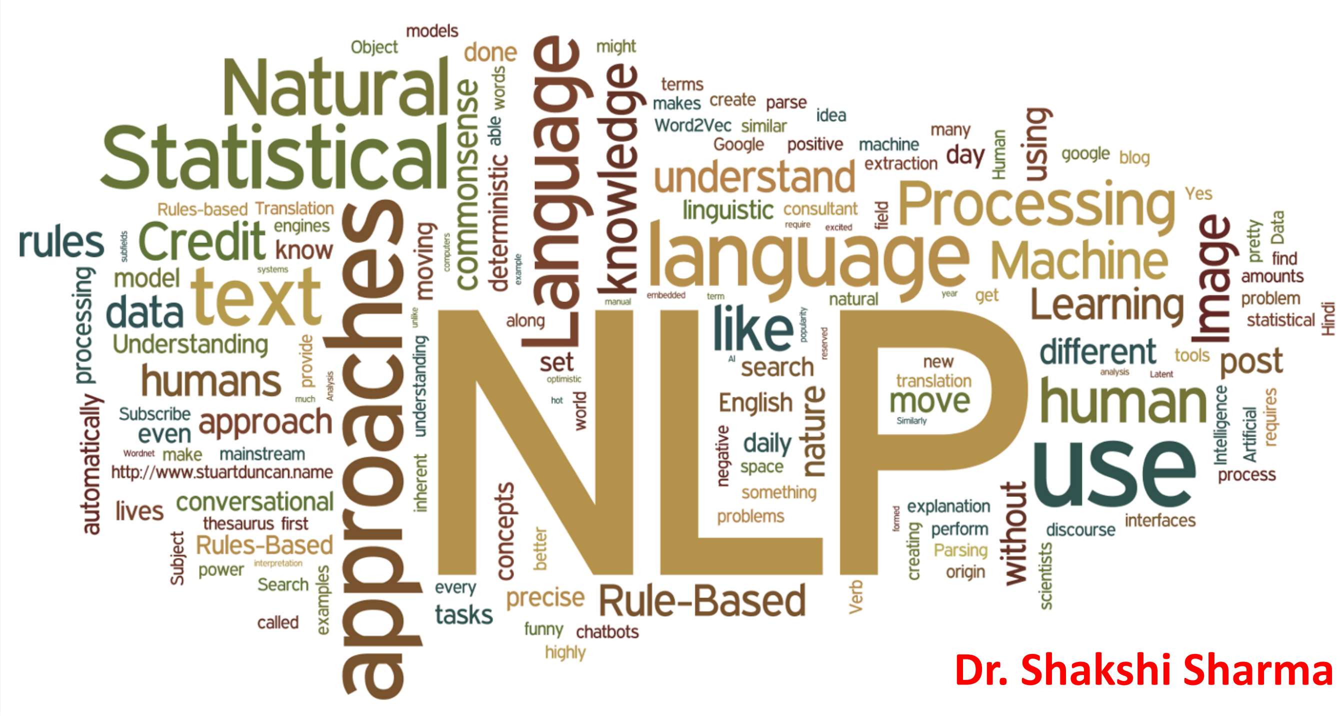
Natural Language Processing



Dr. Shakshi Sharma

Assistant Professor

School of Artificial Intelligence (SoAI), Bennett University

PhD|Junior Researcher|Estonia (Europe)

Research Intern|University of Sheffield, United Kingdom (UK)

Website: https://sites.google.com/view/shakshi-sharma/home

Topic Modeling

Organize the documents into a set of coherent topics

Find relationships between these topics

Understand how different documents talk about the same topic

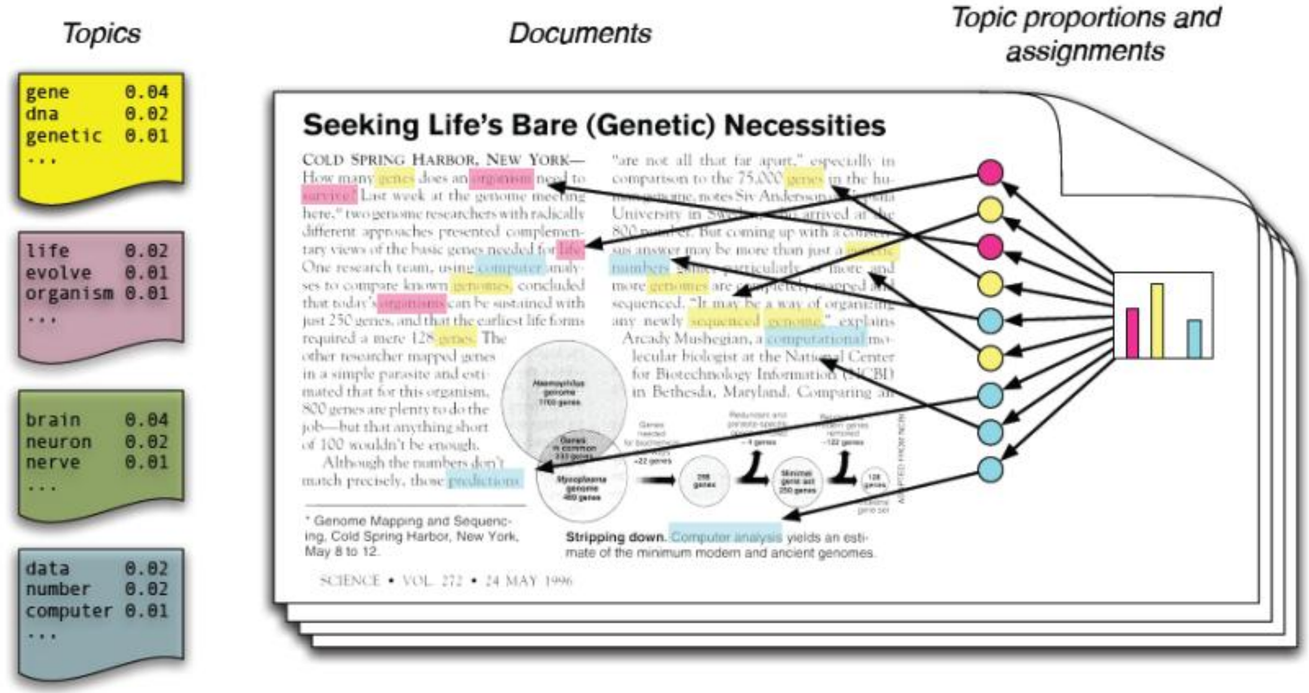
Track the evolution of topics over time

Topic Modeling

A method of (unsupervised) discovery of latent or hidden structure in a corpus

- Applied primarily to text corpora
- Provides a modeling toolbox
- Has prompted the exploration of a variety of new inference methods to accommodate large-scale datasets

Latent Dirichlet Allocation (LDA)



- The researcher picks a number of topics, K .
- Each *topic* (k) is a distribution over words
- Each *document* (d) is a mixture of corpus-wide topics

Dirichlet Distribution from Latent Dirichlet Allocation

- Dirichlet is a **probability distribution** used in statistics and machine learning, especially in **LDA (Latent Dirichlet Allocation)** for topic modeling.

Dirichlet Distribution – A Real-World Analogy

Now, you have a **bowl**, and you must decide how much of each flavor to add. This decision follows a **Dirichlet distribution**!

Dirichlet Distribution – A Real-World Analogy

Now, you have a **bowl**, and you must decide how much of each flavor to add. This decision follows a **Dirichlet distribution**!

Scenario 1: Low Dirichlet Value (e.g., 0.1)
You prefer mostly one flavor.
You scoop **90% chocolate, 5% vanilla, 5% strawberry**.
Your bowl is **dominated by chocolate**.

In LDA terms:
A document is mostly about **one topic**, like "Sports" or "Technology."
Most words in the document belong to a **single** topic.

Dirichlet Distribution – A Real-World Analogy

Now, you have a **bowl**, and you must decide how much of each flavor to add. This decision follows a **Dirichlet distribution**!

Scenario 2: High Dirichlet Value (e.g., 10)
You like all flavors equally.

- You take **33% chocolate, 33% vanilla, 34% strawberry**.
- Your bowl has a **balanced mix of all flavors**.

In LDA terms:

- A document contains **many topics in equal proportions**.
- Each topic contributes a small but **balanced** portion to the document.

.

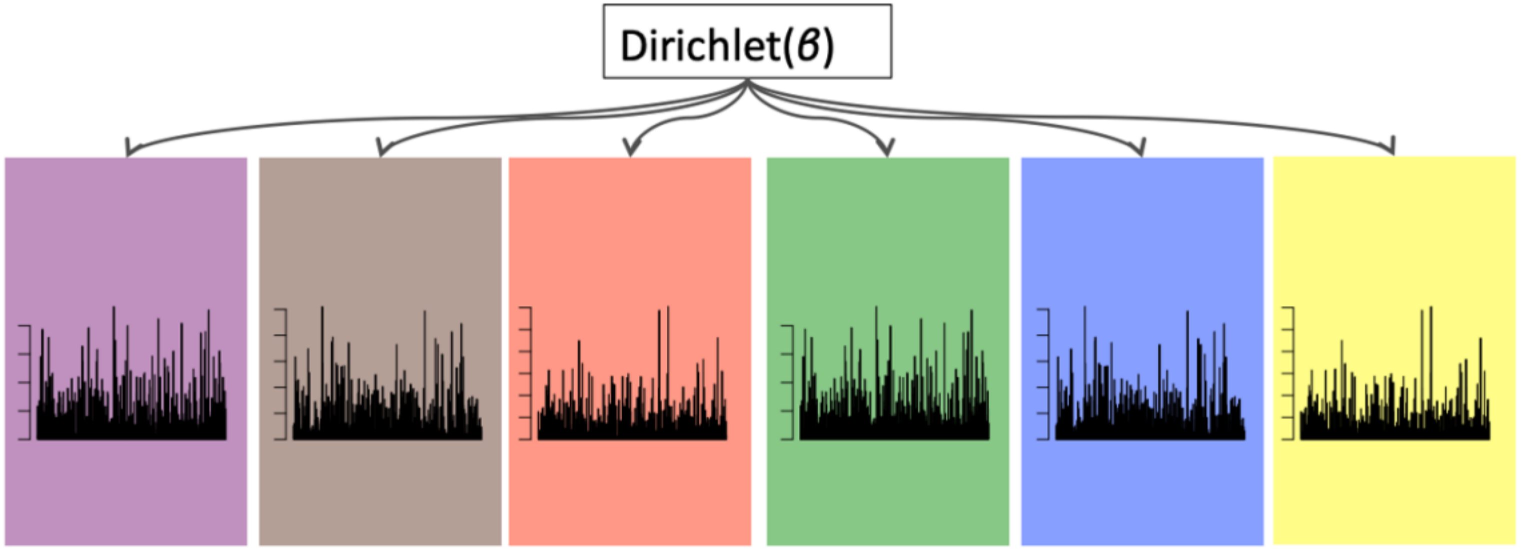
How This Relates to LDA Topic Modeling

1. Ice cream flavors = Topics (Sports, Technology, Politics, etc.)

2. Your bowl = A document

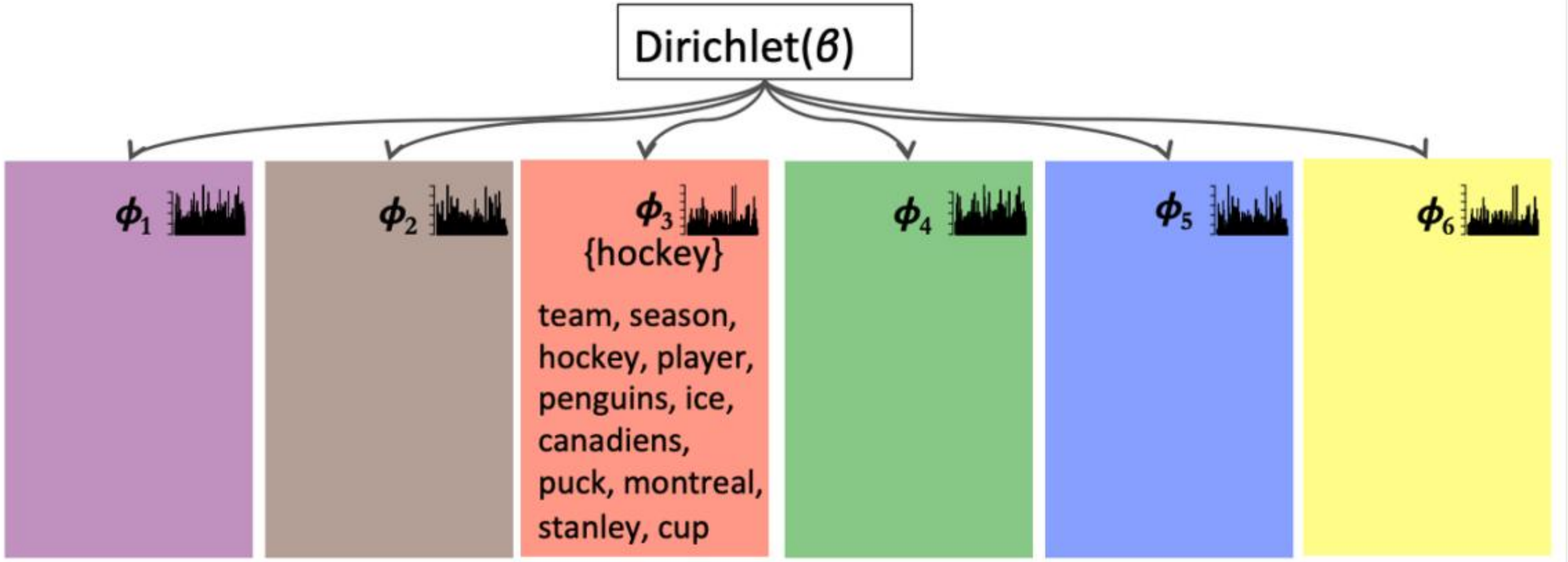
3. The amount of each flavor you choose = Topic distribution in the document

4. Dirichlet value controls how mixed or dominant the topics are



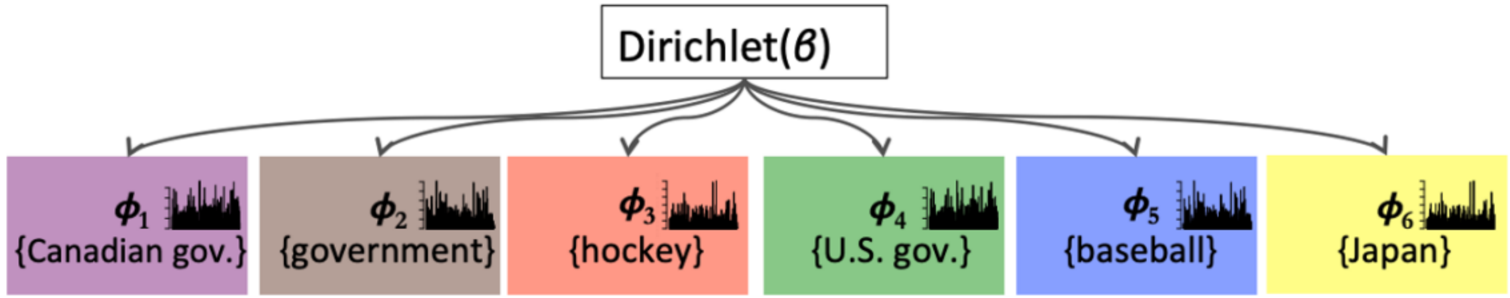
The **generative story** begins with only a **Dirichlet prior** over the topics

Each topic is defined as a **Multinomial distribution** over the vocabulary, parameterized by ϕ_k



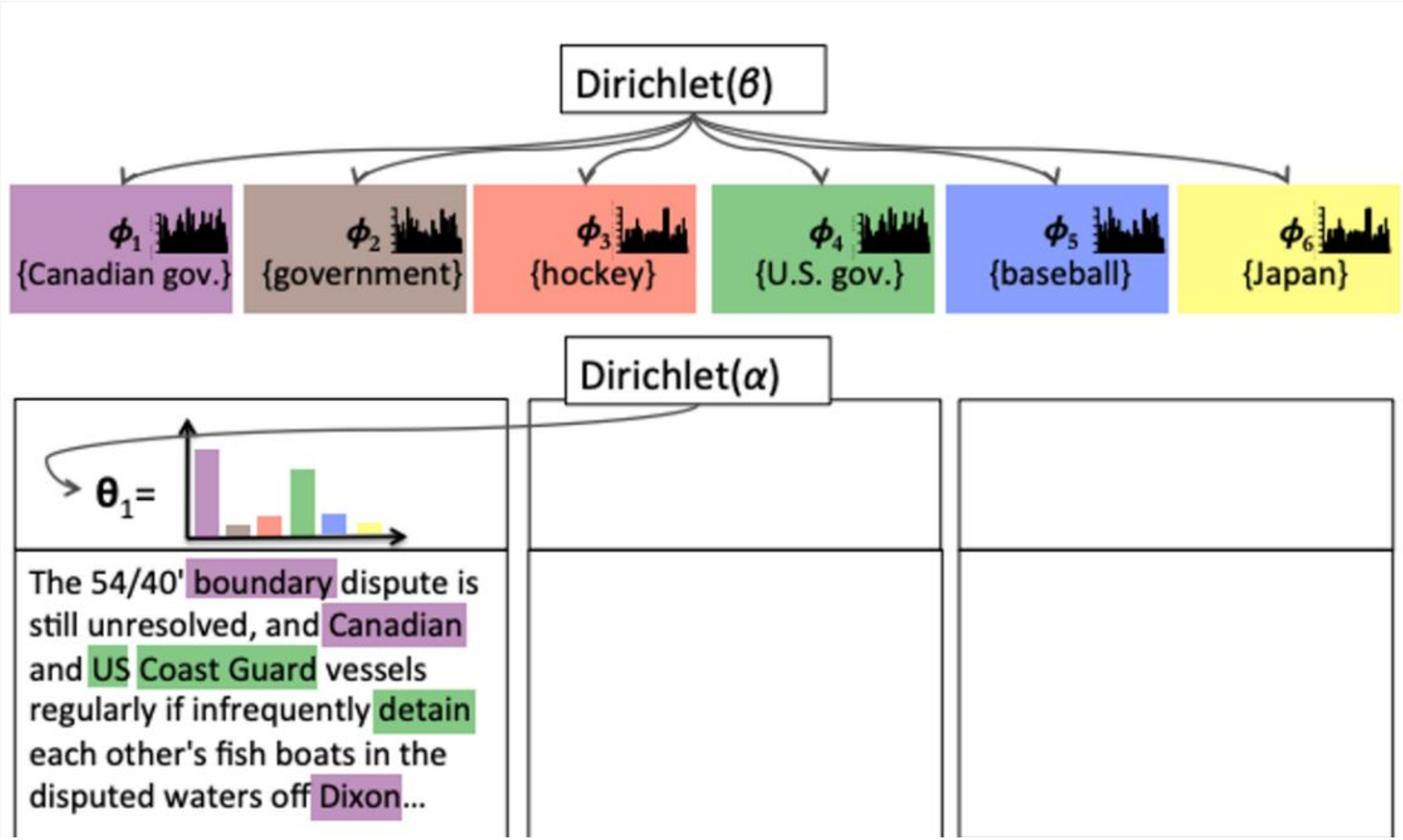
A topic is visualized as its **high probability words.**

A pedagogical **label** is used to identify the topic.

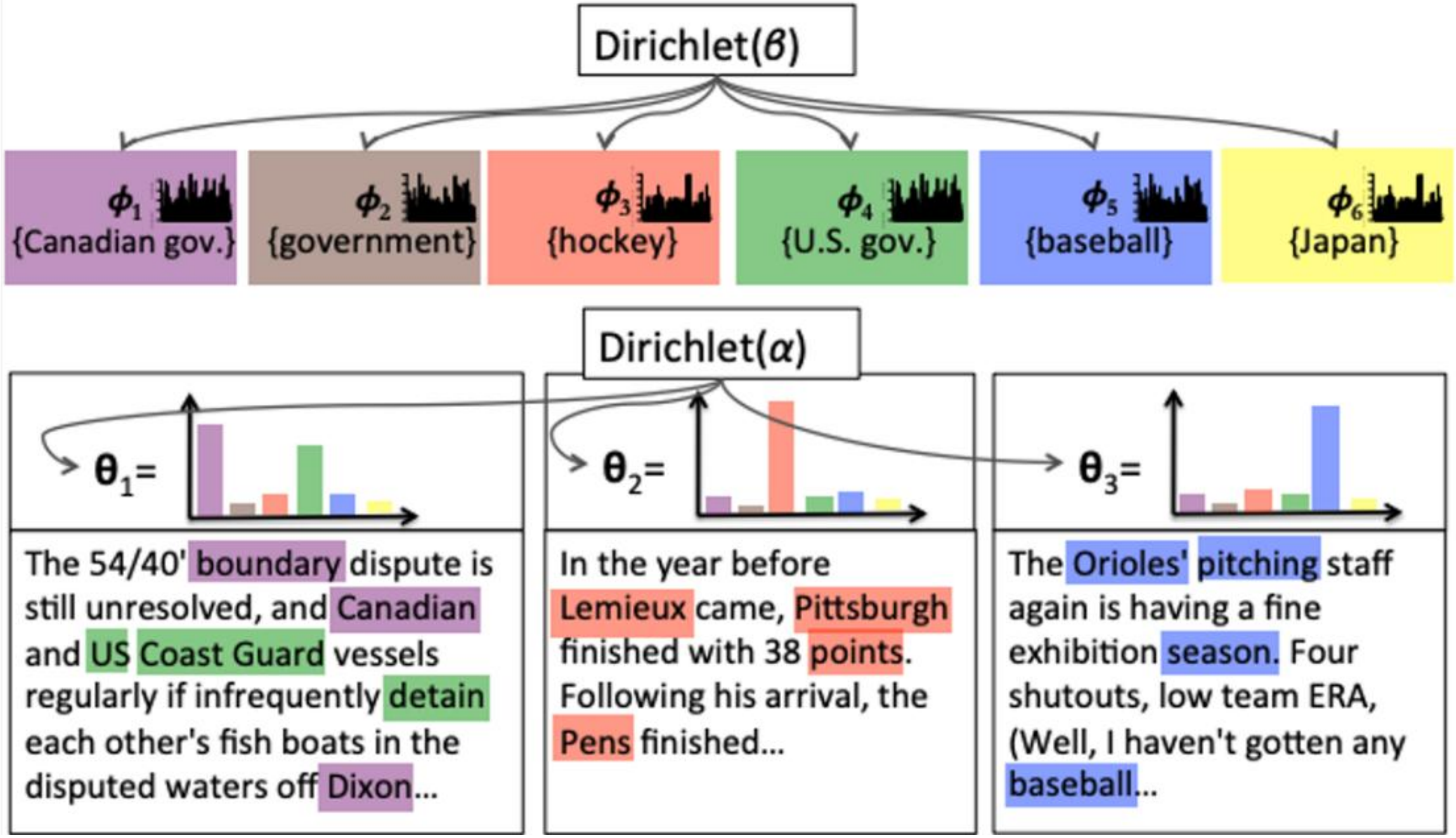


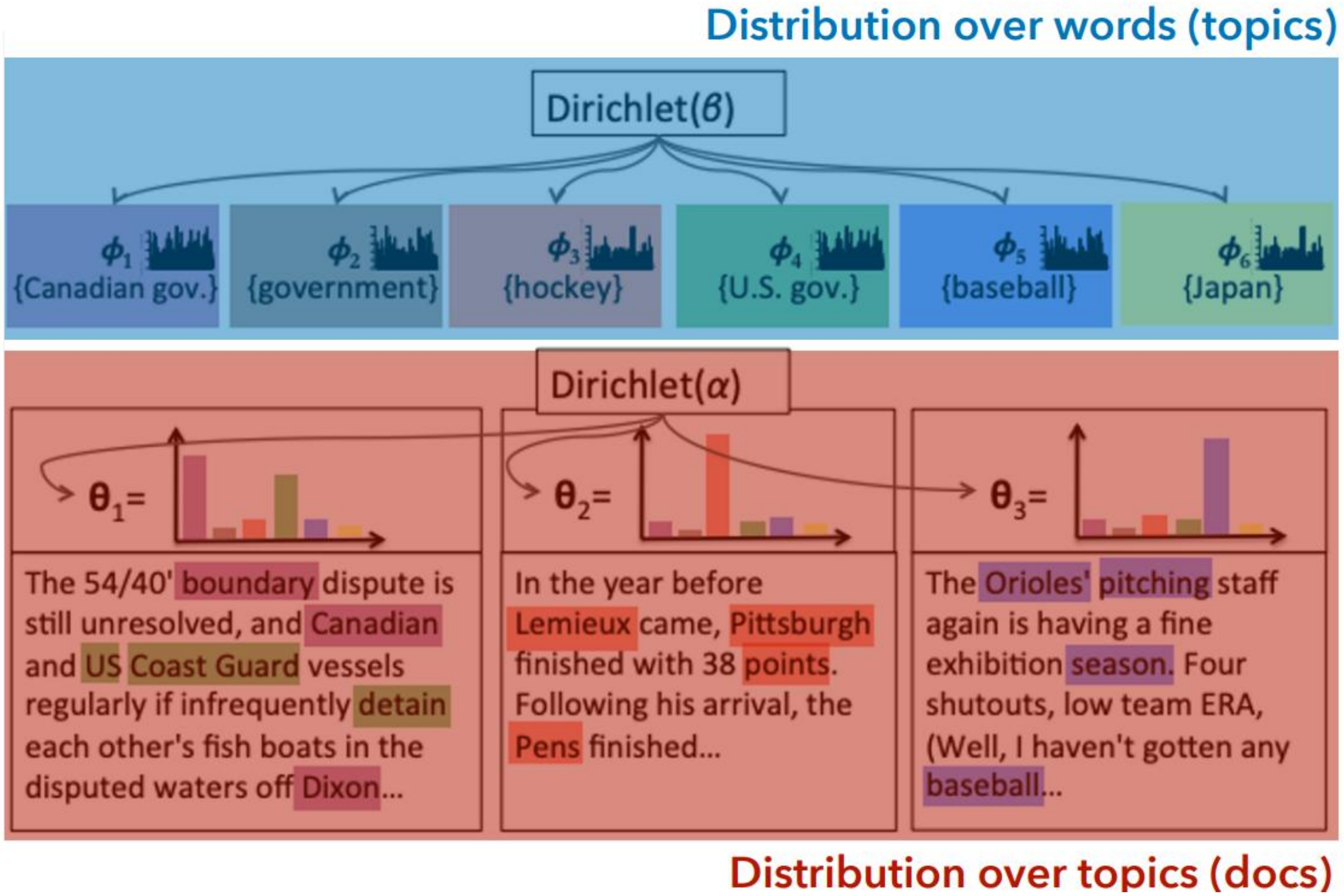
A topic is visualized as its **high probability words.**

A pedagogical **label** is used to identify the topic.



Dirichlet(β)
ϕ1 {Canadian gov.}
ϕ2 {government}
ϕ3 {hockey}
ϕ4 {U.S. gov.}
ϕ5 {baseball}
ϕ6 {Japan}
Dirichlet(α)
θ1=
The 54/40' boundary dispute is still unresolved, and Canadian and US Coast Guard vessels regularly if infrequently detain each other's fish boats in the disputed waters off Dixon...


Dirichlet(β)
ϕ1 {Canadian gov.}
ϕ2 {government}
ϕ3 {hockey}
ϕ4 {U.S. gov.}
ϕ5 {baseball}
ϕ6 {Japan}
Dirichlet(α)
θ1=
θ2=
θ3=
The 54/40' boundary dispute is still unresolved, and Canadian and US Coast Guard vessels regularly if infrequently detain each other's fish boats in the disputed waters off Dixon...
In the year before Lemieux came, Pittsburgh finished with 38 points. Following his arrival, the Pens finished...
The Orioles' pitching staff again is having a fine exhibition season. Four shutouts, low team ERA, (Well, I haven't gotten any baseball...



Distribution over words (topics)
Dirichlet(β)
φ1 {Canadian gov.}
φ2 {government}
φ3 {hockey}
φ4 {U.S. gov.}
φ5 {baseball}
φ6 {Japan}
Dirichlet(α)
θ1=
θ2=
θ3=
The 54/40' boundary dispute is still unresolved, and Canadian and US Coast Guard vessels regularly if infrequently detain each other's fish boats in the disputed waters off Dixon...
In the year before Lemieux came, Pittsburgh finished with 38 points. Following his arrival, the Pens finished...
The Orioles' pitching staff again is having a fine exhibition season. Four shutouts, low team ERA, (Well, I haven't gotten any baseball...
Distribution over topics (docs)

Step 1: Understanding the Dataset

Suppose we have the following set of documents:

1. **Doc 1**: "I love playing football and watching sports."
2. **Doc 2**: "The game of cricket is popular in many countries."
3. **Doc 3**: "Artificial intelligence and machine learning are transforming technology."
4. **Doc 4**: "Data science involves statistics and programming."
5. **Doc 5**: "Many athletes train hard to win championships."

Step 2: Preprocessing the Text

Before applying LDA, we need to clean and process the text.

1. **Tokenization**: Split sentences into words.
   - Example for Doc 1: ["I", "love", "playing", "football", "and", "watching", "sports"]

2. **Lowercasing**: Convert all words to lowercase.
   - Example: ["i", "love", "playing", "football", "and", "watching", "sports"]

3. **Stopword Removal**: Remove common words like "I", "and", "the".
   - Example: ["love", "playing", "football", "watching", "sports"]

4. **Lemmatization**: Convert words to their base form.
   - Example: "playing" → "play", "watching" → "watch"

Step 2: Preprocessing the Text

After processing, our cleaned documents might look like:

1. **Doc 1**: ["love", "play", "football", "watch", "sports"]
2. **Doc 2**: ["game", "cricket", "popular", "many", "country"]
3. **Doc 3**: ["artificial", "intelligence", "machine", "learning", "technology"]
4. **Doc 4**: ["data", "science", "involve", "statistics", "programming"]
5. **Doc 5**: ["athlete", "train", "hard", "win", "championship"]

Step 3: Creating the Document-Term Matrix

- We now represent our documents in a matrix format, where each row represents a document, and each column represents a unique word.

	love	play	football	watch	sports	game	cricket	popular	...	train	win	championship
D1	1	1	1	1	1	0	0	0	...	0	0	0
D2	0	0	0	0	0	1	1	1	...	0	0	0
D3	0	0	0	0	0	0	0	0	...	0	0	0
D4	0	0	0	0	0	0	0	0	...	0	0	0
D5	0	0	0	0	0	0	0	0	...	1	1	1

Step 4: Applying LDA Topic Modeling

LDA **assumes** that:

- Each document is a mixture of topics.
- Each topic is a mixture of words.

- **(A) Initialize Random Topic Assignments**
- **Each word in a document is randomly assigned to one of K topics.**
- Let's say we choose **K = 2 topics** ("Sports" and "Technology").

Example:

- "Football" → **Topic 1 (Sports)**
- "Artificial" → **Topic 2 (Technology)**
- "Train" → **Topic 1 (Sports)**

Step 4: Applying LDA Topic Modeling

(B) Iterative Process Using Gibbs Sampling

- LDA uses **Gibbs Sampling** (a type of Markov Chain Monte Carlo method) to refine topic assignments **over many iterations**.
- For each word in a document, LDA updates its topic assignment based on:

1. **How common the word is in each topic.**
    1. If "football" appears frequently in **Topic 1**, it is likely to stay in **Topic 1**.
2. **How prevalent the topic is in the document.**
    1. If most words in **Doc 1** are about sports, new words are more likely to be assigned to the **Sports topic**.

This step repeats thousands of times until topics stabilize!

Step 5: Extracting Topics and Results

After enough iterations, the model identifies **meaningful topics**.

(A) Topic-Word Distribution
LDA produces a **probability distribution of words for each topic**.

Topic	Top Words
Topic 1 (Sports)	football, player, train, match, goal
Topic 2 (Technology)	artificial, intelligence, technology, data, machine

Step 5: Extracting Topics and Results

(B) Document-Topic Distribution
LDA also assigns **a probability distribution of topics** for each document.

Document	**Topic 1 (Sports)**	**Topic 2 (Technology)**
D1 ("Football players train hard")	**90%**	10%
D2 ("Artificial Intelligence is transforming technology")	5%	**95%**

So, Doc 1 and Doc 2 belong mainly to Topic 1 (**Sports**),
while Doc 3 and Doc 4 belong to Topic 2 (**Technology**).

Step 6: Interpreting the Results

Based on LDA results:

- **If a new document contains words like "football" or "cricket"**, it is likely about **sports**.
- **If a new document contains words like "AI" or "programming"**, it is likely about **technology**.

Summary of LDA Process

1. **Preprocess the text** (tokenization, stopword removal, lemmatization).
2. **Create a Document-Term Matrix** (word frequencies).
3. **Randomly assign words to topics** (initialize topic assignments).
4. **Use Gibbs Sampling** to iteratively refine topic distributions.
5. **Extract topics and document-topic probabilities.**

Evaluating Topic Modeling

Manual Inspection / Human judgement

- Top ranked words

Intrinsic Evaluation

- Coherence score
- Intruder test

Extrinsic Evaluation

- Downstream application

Evaluating LDA Models: Perplexity and Topic Coherence

Perplexity

Measures how well the model predicts the data. Lower perplexity is better.

Topic Coherence

Measures the interpretability of the topics. Higher coherence is preferred.

Applications of Topic Modeling

- **Topic Classification:** Automatically classify new documents into topics.
- **Document Clustering:** Group similar articles together.
- **Keyword Extraction:** Identify key themes in large datasets.
- **Recommendation Systems:** Suggest articles based on topic similarity.

Limitations of LDA

1. Requires **Predefined** Number of Topics

2. Struggles with **Short Texts**
   - LDA performs poorly on short documents (e.g., tweets, single-sentence reviews) because it relies on word co-occurrence patterns.

3 . Struggles with **Overlapping Topics**
   - In reality, topics often overlap, but LDA assumes each document is a mixture of separate, well-defined topics.

Alternatives to LDA

To overcome these limitations, modern alternatives include:

1. **BERTopic** (uses Transformers + clustering for better topics)
2. **Neural Topic Models** (e.g., ProdLDA, ETM)
3. **Non-Negative Matrix Factorization (NMF)** (simpler and sometimes more effective)
4. **Word Embedding-Based Models** (e.g., LDA2Vec)

References

- https://courses.cs.washington.edu/courses/cse447/23wi/assets/slides/15-SequenceLabeling-UndergradNLP-2023wi.pdf
- https://web.stanford.edu/class/cs224c/slides/s5_topic_modeling.pdf