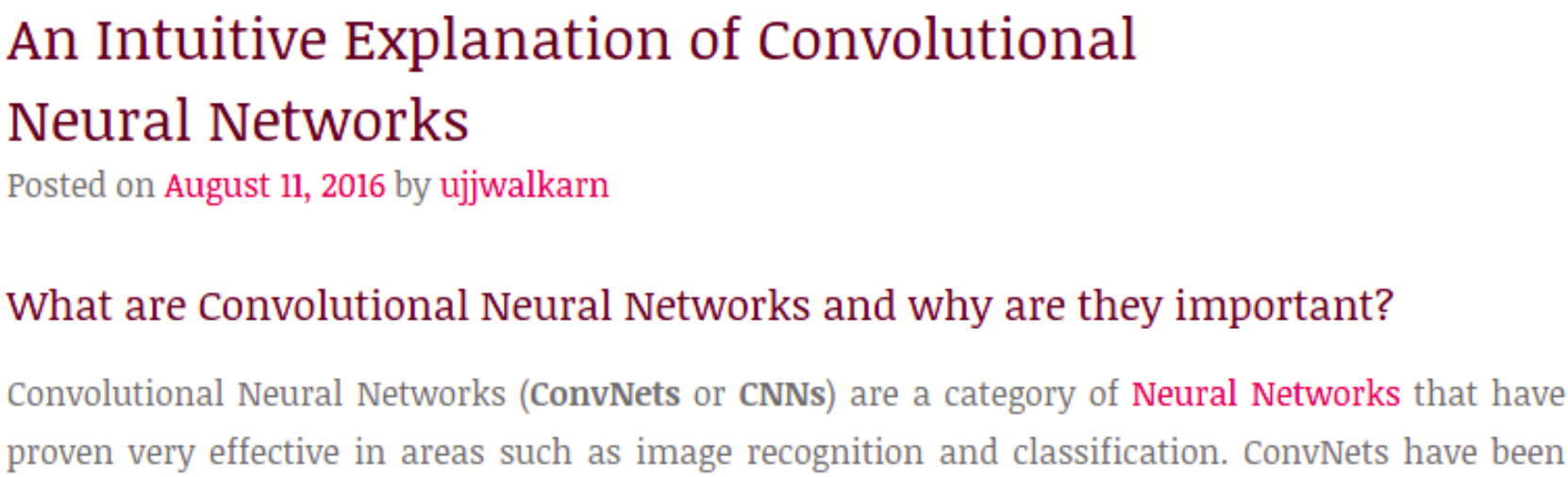
# Information Retrieval (CSD510)

Parametric and Zone Indexing

January 29, 2024

# Documents with Structure

- So far, we discussed unstructured text.
- Many documents in reality have a structure.
- They are composed of
  - Fields
  - Zones



An Intuitive Explanation of Convolutional Neural Networks

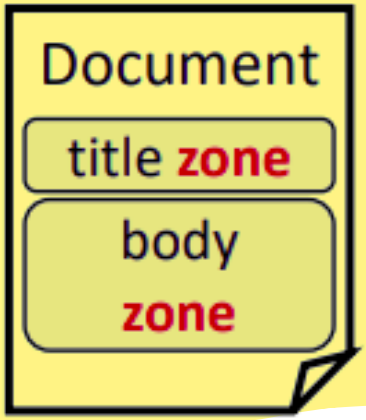
Posted on August 11, 2016 by ujjwalkarn

What are Convolutional Neural Networks and why are they important?

Convolutional Neural Networks (**ConvNets** or **CNNs**) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. ConvNets have been

# Zones and Fields

- A document is associated with fields and zones.

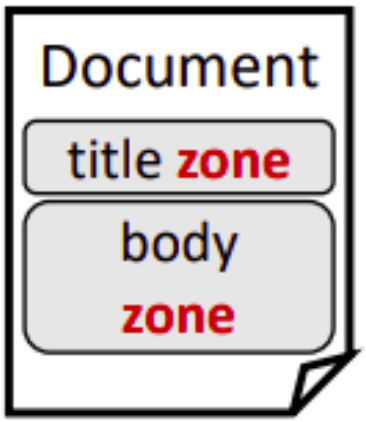



**METADATA Fields**
**filename**
**author**
**title**
**date of creation**

- **Sample Query:** find documents authored by William Shakespeare in 1601 containing the phrase "you brutus"
- **Sample Query**: find documents with merchant in the title and william in the author list and the phrase gentle rain in the body

# Zones and Fields

- Zones are arbitrary free text.
- Fields may take relatively small set of values...(may call for range query (year between 1600 and 1700) support)




**METADATA Fields**
**filename**
**author**
**title**
**date of creation**

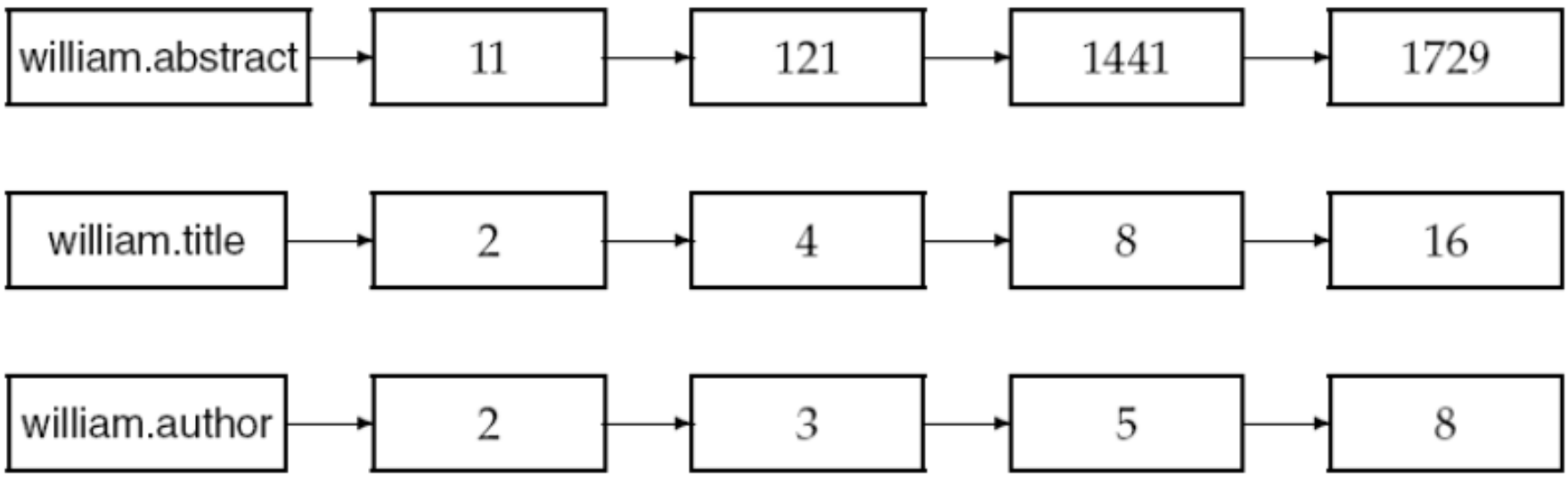
**How to index zones and fields?**

# Indexing fields

- Parameteric indexing
- One parameteric index for each field.
- May call for range query (year between 1600 and 1700) support.

# Indexing Zones and Fields

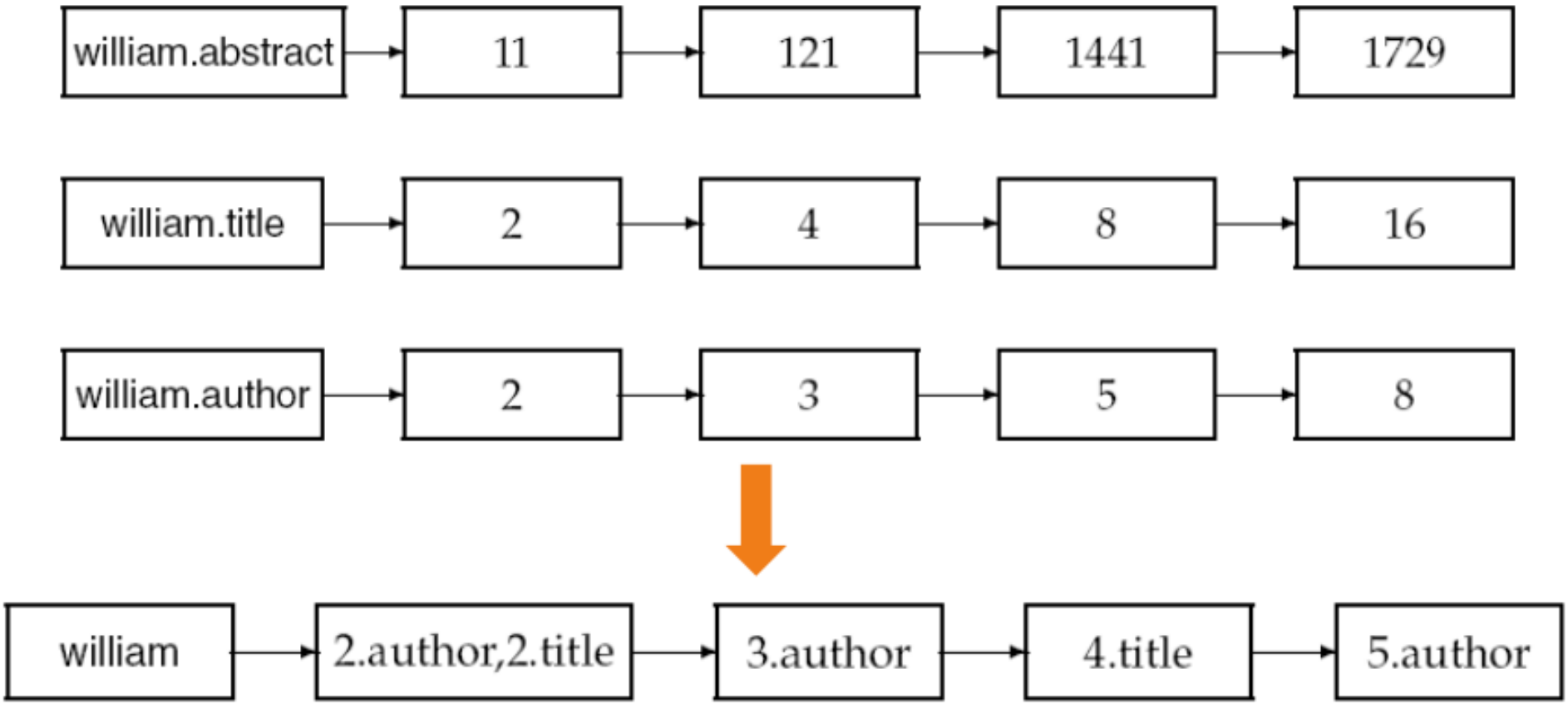
- Create separate Index for each field and each zone.
  - Standard Inverted Index +
  - Parametric Indexes (one for each field) +
  - Zone Indexes (one for each zone)




Zone Index Example

# We can do better...

- Encode zones in postings.



# Weighted Zones

- Not all zones are equally important!
- Consider a collection where documents have three zones ($l = 3$):
  1. author (least important)
  2. title (more important)
  3. body (most important)
- We can associate a weight, $g_i$ to each zone

1. author ($g_1 = 0.2$)
2. title ($g_2 = 0.3$)
3. body ($g_3 = 0.5$)

$$\sum_{i=1}^{l} g_i = 1$$

$$g_i \in [0,1]$$

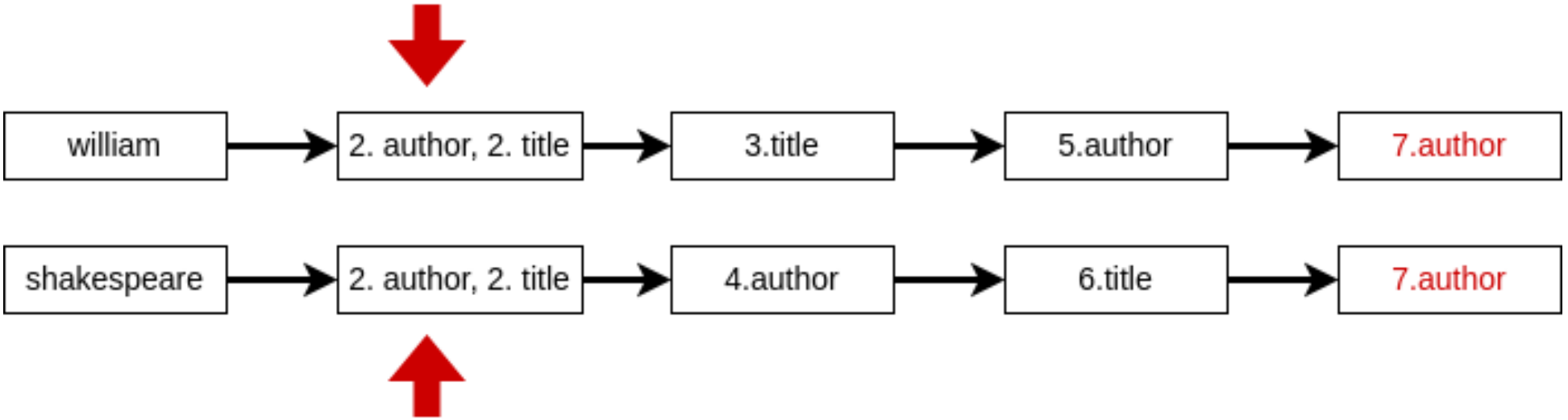
# Weighted Zone Scoring

- If all query terms appear in $i^{th}$ zone,
  - we say $s_i = 1$
- Then, we score the document as $\sum_{i=1}^{l} g_i s_i$

# Weighted Zone Scoring on Inverted Index

A Match Found? Add $g_i$ to an array **score[docID]**.

This array is often called **Accumulator**.



# Machine Learned Relevance

- Use human-annotated training examples
- Consider:
  - Only two zones exist: title and body.
  - $S_T(d, q)$= 1 if query term exists in title.
  - $S_B(d, q)$= 1 if query term exists in body

| Example | DocID | Query | $s_T$ | $s_B$ | Judgment |
|---|---|---|---|---|---|
| $\Phi_1$ | 37 | linux | 1 | 1 | Relevant |
| $\Phi_2$ | 37 | penguin | 0 | 1 | Non-relevant |
| $\Phi_3$ | 238 | system | 0 | 1 | Relevant |
| $\Phi_4$ | 238 | penguin | 0 | 0 | Non-relevant |
| $\Phi_5$ | 1741 | kernel | 1 | 1 | Relevant |
| $\Phi_6$ | 2094 | driver | 0 | 1 | Relevant |
| $\Phi_7$ | 3191 | driver | 1 | 0 | Non-relevant |

# Learning Weights (without ML)

| Query | DocID | User Judgment |
|---|---|---|
| linux | 37 | 1 |
| system | 238 | 1 |
| kernel | 1741 | 1 |
| driver | 2094 | 1 |

Judgment of 1 implies the document is relevant.

| Query | DocID | In Title? ($S_T$) | In Body? ($S_B$) | Our Score |
|---|---|---|---|---|
| linux | 37 | 1 | 1 | ? |
| system | 238 | 0 | 1 | ? |
| kernel | 1741 | 1 | 1 | ? |
| driver | 2094 | 0 | 1 | ? |

Assume zone weights: title ($g_1 = 0.3$) and body ($g_2 = 1 - 0.3 = 0.7$)

# Learning Weights

| Query | DocID | User Judgment |
|---|---|---|
| linux | 37 | 1 |
| system | 238 | 1 |
| kernel | 1741 | 1 |
| driver | 2094 | 1 |

| Query | DocID | In Title? ($S_T$) | In Body? ($S_B$) | Our Score |
|---|---|---|---|---|
| linux | 37 | 1 | 1 | 1 |
| system | 238 | 0 | 1 | 0.7 |
| kernel | 1741 | 1 | 1 | 1 |
| driver | 2094 | 0 | 1 | 0.7 |

# Learning Weights

If g is the title weight, how to quantify the error?

| Query | DocID | User Judgment |
|---|---|---|
| linux | 37 | 1 |
| system | 238 | 1 |
| kernel | 1741 | 1 |
| driver | 2094 | 1 |

| $s_T$ | $s_B$ | Score |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 - g |
| 1 | 0 | g |
| 1 | 1 | 1 |

| Query | DocID | In Title? ($S_T$) | In Body? ($S_B$) | Our Score |
|---|---|---|---|---|
| linux | 37 | 1 | 1 | 1 |
| system | 238 | 0 | 1 | 1 – g |
| kernel | 1741 | 1 | 1 | 1 |
| driver | 2094 | 0 | 1 | 1 - g |

# Learning Weights

$$score = g.s_T + (1 - g).s_B$$

Then the error,

$$\epsilon = (\text{relevance} - score)^2$$

Our objective is to find g such that we minimize the total error,

$$\sum_{all\ documents} \epsilon$$

# For the 01 Case...

How to quantify the error?

| Query | DocID | User Judgment |
|---|---|---|
| linux | 37 | 1 |
| system | 238 | 1 |
| kernel | 1741 | 1 |
| driver | 2094 | 1 |

| $s_T$ | $s_B$ | Score |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 - g |
| 1 | 0 | g |
| 1 | 1 | 1 |

| Query | DocID | In Title? ($S_T$) | In Body? ($S_B$) | Our Score |
|---|---|---|---|---|
| linux | 37 | 1 | 1 | 1 |
| system | 238 | 0 | 1 | 1 – g |
| kernel | 1741 | 1 | 1 | 1 |
| driver | 2094 | 0 | 1 | 1 - g |

*What if we have non-relevant judgments also?

# For 01 case...

$$\text{Error} = [1 - (1 - g)]^2 n_{01r} + [0 - (1 - g)]^2 n_{01n}$$

# Total Error

$$(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1 - g)^2 + n_{00r} + n_{11n}$$

# Total Error

$$(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1 - g)^2 + n_{00r} + n_{11n}$$

# Total Error

$(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1 - g)^2 + n_{00r} + n_{11n}$

Differentiate w.r.t g and equate to zero

# Total Error

$(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1 - g)^2 + n_{00r} + n_{11n}$

Differentiate w.r.t g and equate to zero

$$g = \frac{n_{01r}+n_{10n}}{n_{10r}+n_{10n}+n_{01r}+n_{01n}}$$