Data Science Bootcamp, 11th January 2017

N-gram Language Models
Vector Space Model

Kriste Krstovski

Columbia University
Data Science Institute

In This Lab Session

- **Natural Language Processing (NLP)**
  - Language Modeling
  - Machine Translation
  - Information Extraction
  - Text Summarization
  - Dialogue Systems
  - and many more…

- **Information Retrieval (IR)**
  - Retrieval Models
  - Learning to Rank
  - Image and Multimodal Search
  - Web Search
  - Question Answering
  - and many more…

In This Lab Session

- N-gram Language Models
- Vector Space Model

In This Lab Session

- **N-gram Language Models**
- Vector Space Model

What is a Language Model?

- One of the most important concepts in NLP
- Help us assign probabilities to sequence of words (e.g. sentences, phrases, tweets, etc.)
- Applications in many CS areas: speech recognition, optical character recognition (OCR), machine translation, etc.
- Given a sentence S with a set of w_i words, $i = 1, 2, ..., n$, language models formally define the probability of the sentence as the probability of having the particular sequence of words:

$$p(S) = p(w_1, w_2, w_3, ..., w_n)$$

What is a Language Model?

- Predict the probability of a specific word given the previous words in the sentence:

$$p(w|w_{-1}, w_{-2}..w_{-k})$$

- Probability of a sentence is computed using the chain rule:

$$p(S) = p(w_1, w_2, w_3, ..., w_n) = \prod_{i}^{n} p(w_i|w_1, w_2, ..., w_{i-1})$$

What is a Language Model?

- Probability of a sentence is computed using the chain rule:

$$p(S) = p(w_1, w_2, w_3, ..., w_n) = \prod_{i}^{n} p(w_i | w_1, w_2, ..., w_{i-1})$$

- Markov assumption to simplify the computation of the above probability:

$$\prod_{i}^{n} p(w_i | w_1, w_2, ..., w_{i-1}) \approx \prod_{i}^{n} p(w_i | w_{i-k}, w_{i-(k-1)}, ..., w_{i-1})$$

N-gram Models

- Unigram:

$$p(S) \approx \prod_{i}^{n} p(w_i)$$

- Bigram:

$$p(S) \approx \prod_{i}^{n} p(w_i | w_{i-1})$$

- Trigram:

$$p(S) \approx \prod_{i}^{n} p(w_i | w_{i-1}, w_{i-2})$$

Computing N-gram Models using ML estimates

- Unigram:

$$p(w_i) = \frac{count(w_i)}{\sum_{i=1}^{v} count(w_i)}$$

- Bigram:

$$p(w_i|w_{i-1}) = \frac{count(w_i|w_{i-1})}{count(w_i)}$$

- Trigram:

$$p(w_i|w_{i-1}, w_{i-2}) = \frac{count(w_i|w_{i-1}, w_{i-2})}{count(w_{i-1}, w_{i-2})}$$

Example

- S= "I would rather do data science bootcamp"
- Unigram:

$$p(S) \approx \prod_{i}^{n} p(w_i) = p(i) * p(would) * (p(rather) * p(do) * p(data) * p(science) * p(bootcamp)$$

- Bigram:

$$p(S) \approx \prod_{i}^{n} p(w_i|w_{i-1}) = p(i|none) * p(would|i) * p(rather|would) * p(do|rather)$$

$$*p(data|do) * p(science|data) * p(bootcamp|science)$$

Example

- S= "I would rather do data science bootcamp"
- Trigram:

$$p(S) \approx \prod_{i}^{n} p(w_i|w_{i-1}, w_{i-2}) = p(i|none, none) * p(would|none, i) * p(rather|i, would)$$

$$*p(do|would, rather) * p(data|rather, do)$$

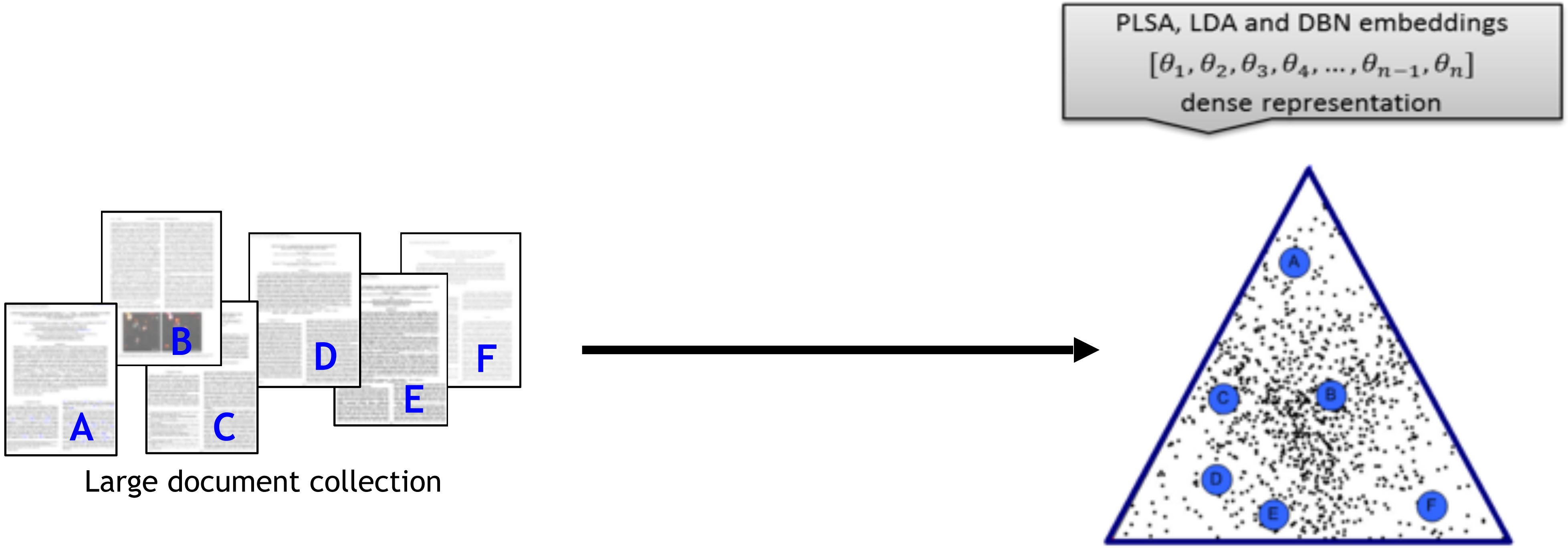
$$*p(science|do, data) * p(bootcamp|data, science)$$

In This Lab Session

- N-gram Language Models
- **Vector Space Model**

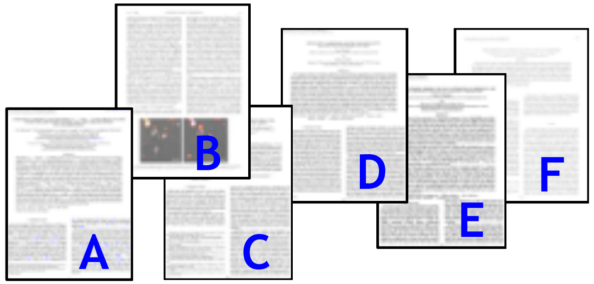
Document Similarity

- How do we represent documents in a shared space?



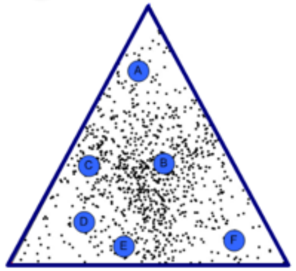
Document Similarity

- How do we do it?

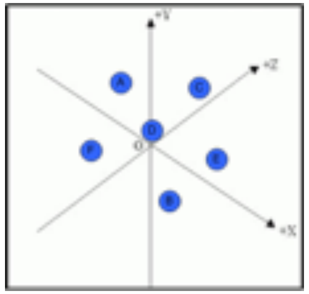


Large document collection

PLSA, LDA and DBN embeddings
$[\theta_1, \theta_2, \theta_3, \theta_4, \ldots, \theta_{n-1}, \theta_n]$
dense representation

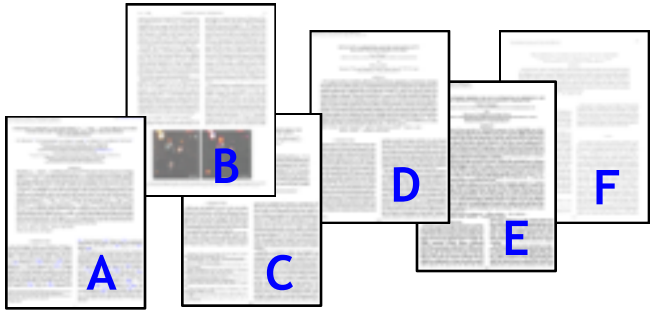


Vector space model
$[tf-idf_1, 0, 0, tf-idf_4, \ldots, 0, 0]$
sparse representation



Vector Space Model

- tf-idf weighting:



A	t_1	t_2	t_3	t_4	t_5	$\cdots$	$\cdots$	$\cdots$	$\cdots$	t_n
B	t_1	t_2	t_3	t_4	t_5	$\cdots$	$\cdots$	$\cdots$	$\cdots$	t_n
$\vdots$						$\vdots$				
F	t_1	t_2	t_3	t_4	t_5	$\cdots$	$\cdots$	$\cdots$	$\cdots$	t_n

Vector Space Model

- tf-idf weighting:

$$tf_{ik} * idf_k \qquad tf_{ik} = \frac{f_{ik}}{\sum_{j=1}^{t} tf_{ij}} \qquad idf_k = \log \frac{N}{n_k}$$

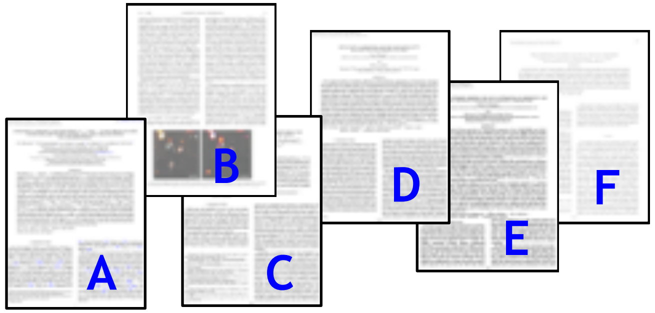
- Example:

B	t_1	t_2	t_3	t_4	t_5	$\cdots$	$\cdots$	$\cdots$	$\cdots$	t_n

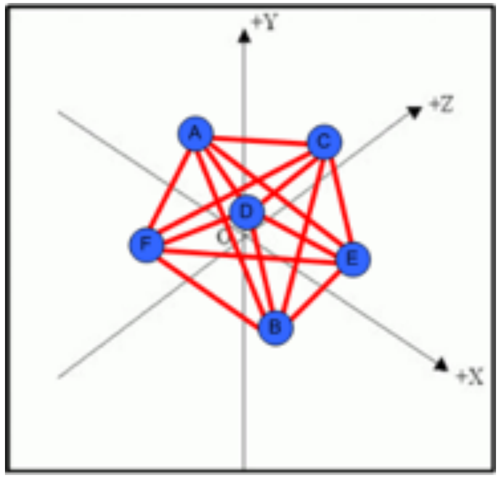
$$tf_{B3} = \frac{f_{B3}}{\sum_{j=1}^{|B|} tf_{Bj}} \qquad idf_3 = \log \frac{N}{n_3}$$

Vector Space Model

- Compute Document Similarity using distance metrics: Euclidean, Cosine, etc.



Large document collection



In This Lab Session

- Compute unigram, bigram and trigram LMs on books
- Use vector space model to represent books and run a query