Week 1

06 July 2018 22:20

Lecture 1.1 (NL Content Analysis):

What is NLP?
Why is it difficult for computers?
Sematic Analysis
- Part of speech tagging
- Entity extraction - partial understanding of sentence
- Word sense disintegration - how words have different meanings in different contexts
- Sentiment analysis - whether sentence is positive or negative - useful in review analysis

Inference
Speech Act analysis

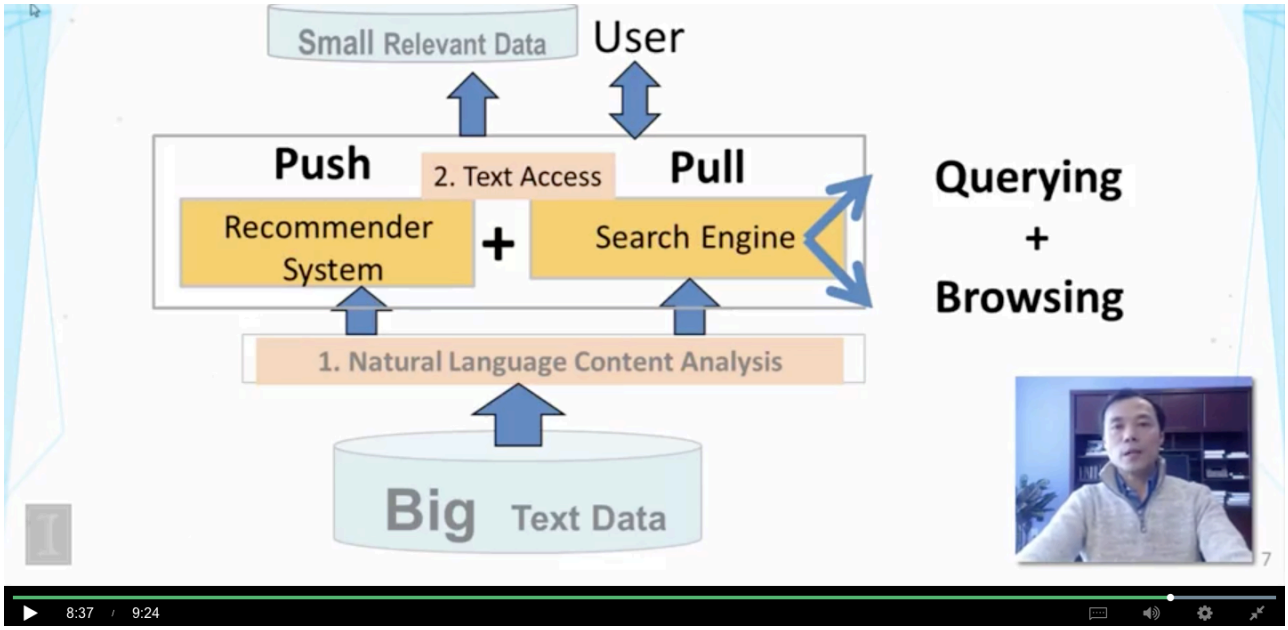
NLP for Text Retrieval - General, robust & efficient - shallow NLP techniques are used
Bag of words representation - used by most search engines

————————————————————

Lecture 1.2 (Text Access):

Text-Information systems
Modes of text access:
- Push (recommender)
- vs.
- Pull (search engines)
0. Querying vs. Browsing

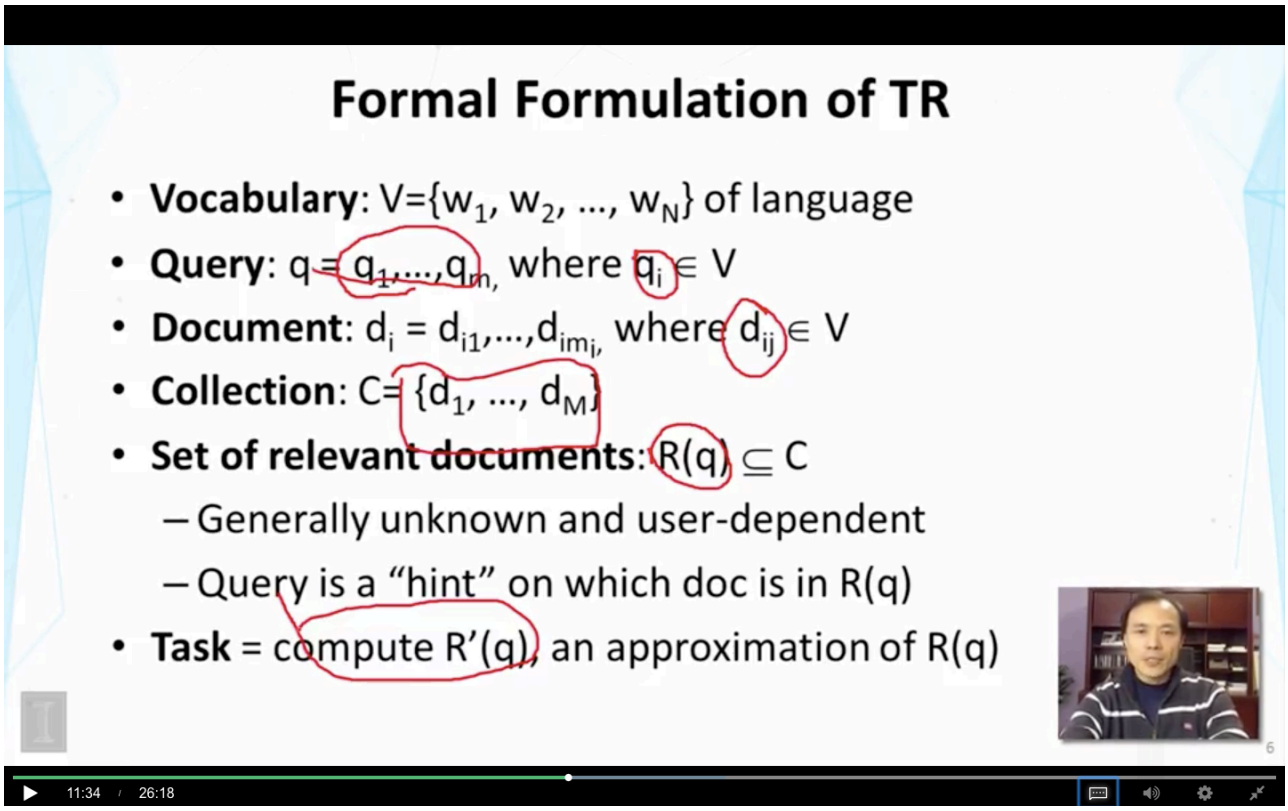
Summary text access (picture):



———————————————————————

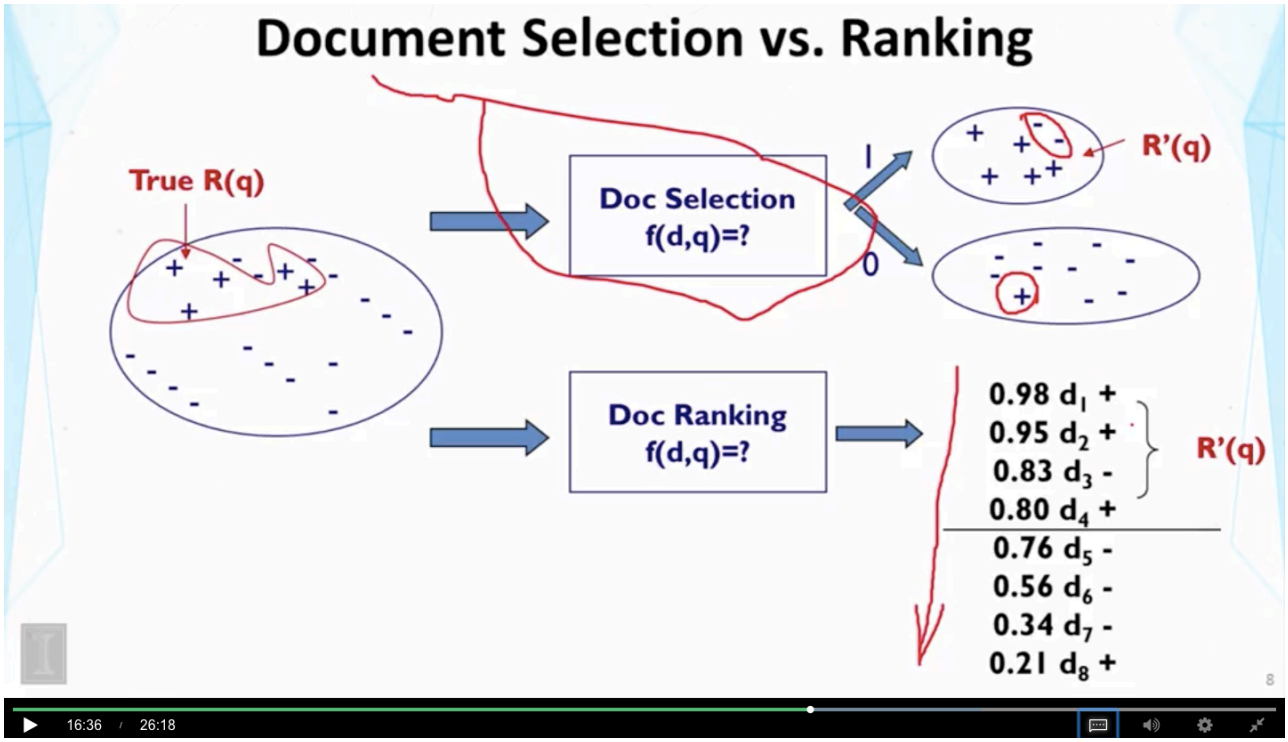
Lecture 1.3 (Text Retrieval problem):

Formal Formulation of text retrieval:

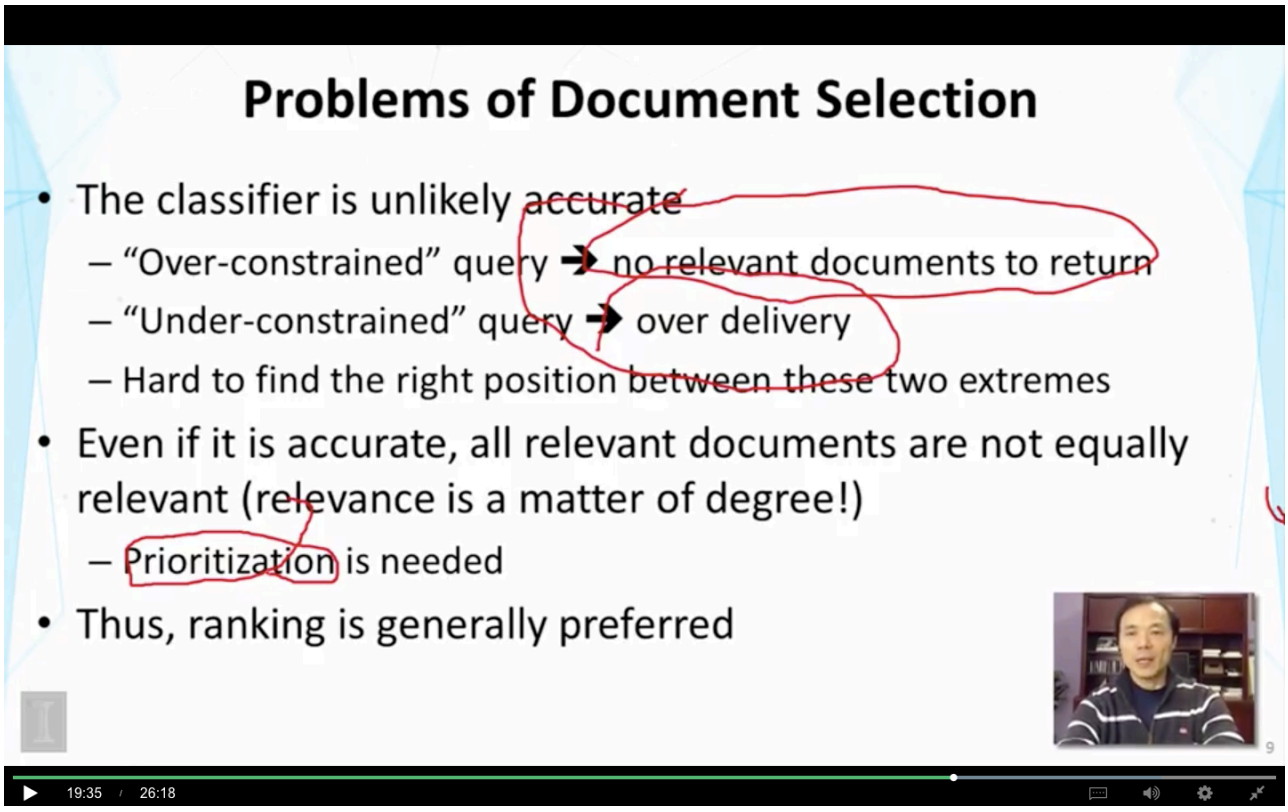


Methods of TR:

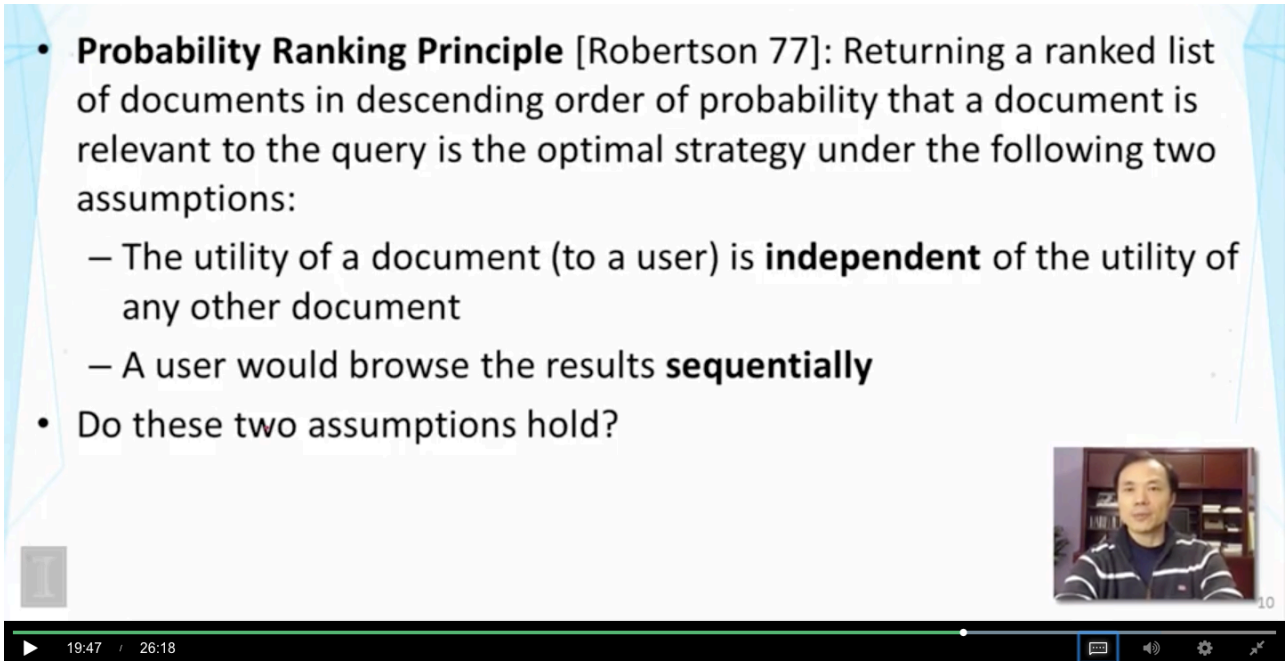
1. Document selection - binary classifier (document shown or not)
2. Document ranking - assign probability of relevance of the document



Document ranking is better suited to user queries. Why?
Problems with document selection:



Why Ranking? Theoretically justify ranking method using probability ranking principle.



However, none of the assumptions are actually true.

a. In first assumption, there is no use showing those documents which are identical or similar to the ones the user has read, meaning that utility of documents depend on other documents.
b. In second assumption, it might be the case that one document might not be helpful to the user but, when three documents are put together, they make sense to the user and satiate user's query.

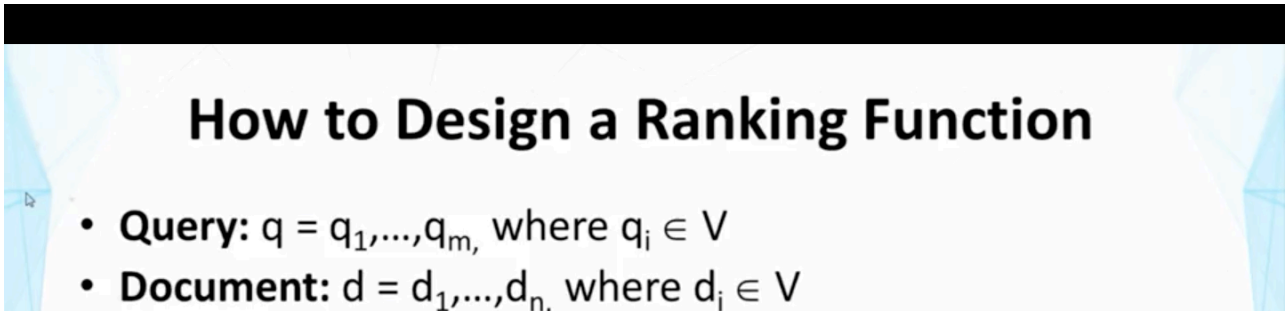
These assumptions aren't necessarily true, still a lot of research work is based on these assumptions. We can address this problem by post-processing rank list to remove not useful / redundant documents.

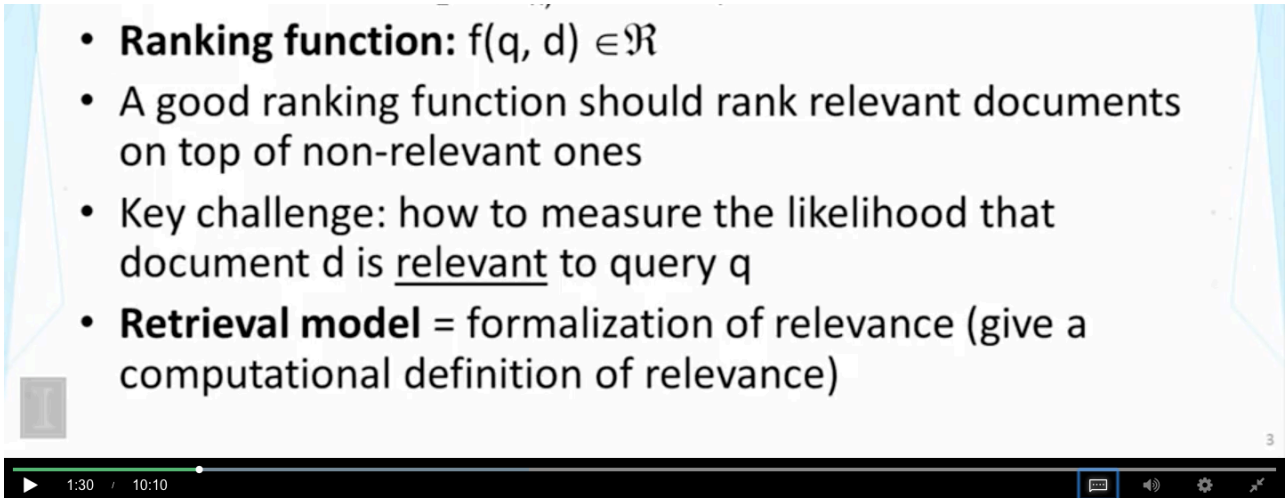
The main challenge is to design the ranking function, what value is to be assigned to the query-document pair.

————————————————————

Lecture 1.4 (Text Retrieval Methods):

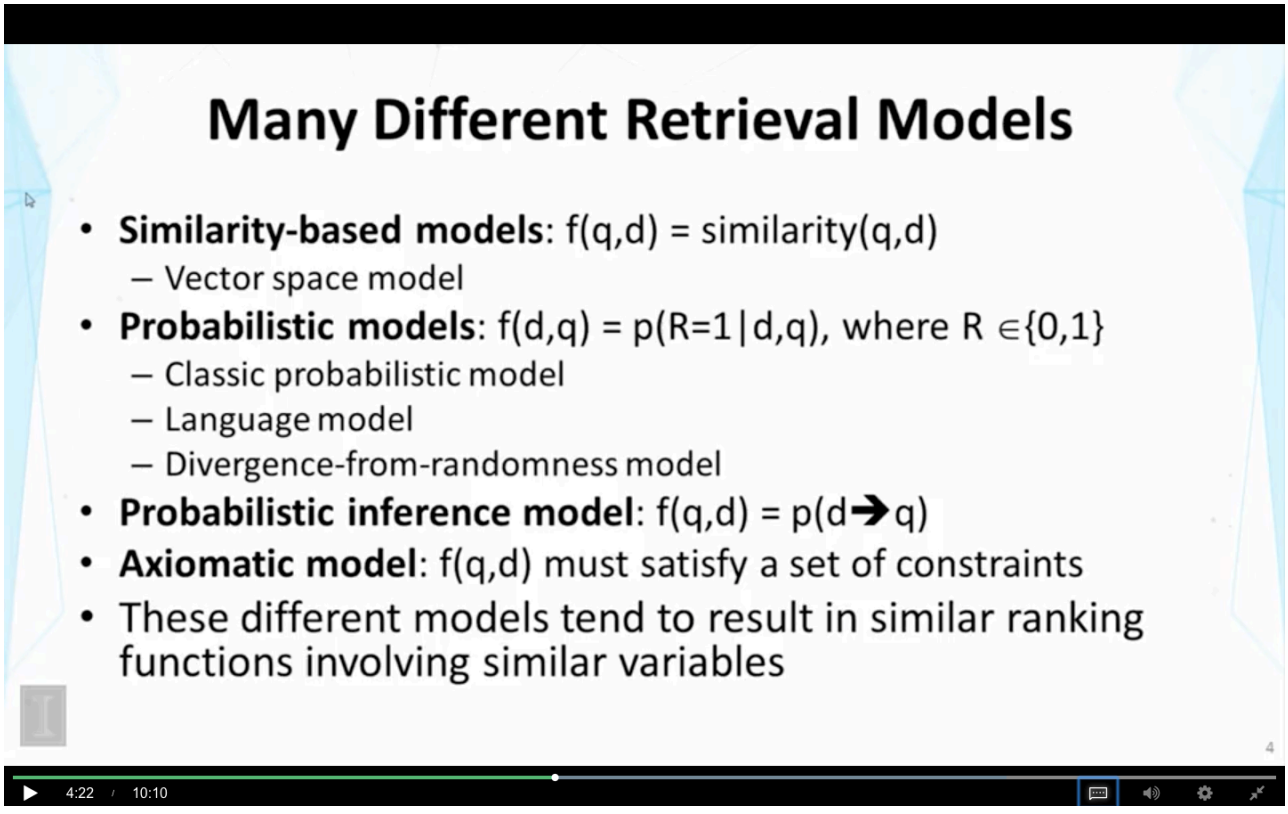
How to design a Ranking function:





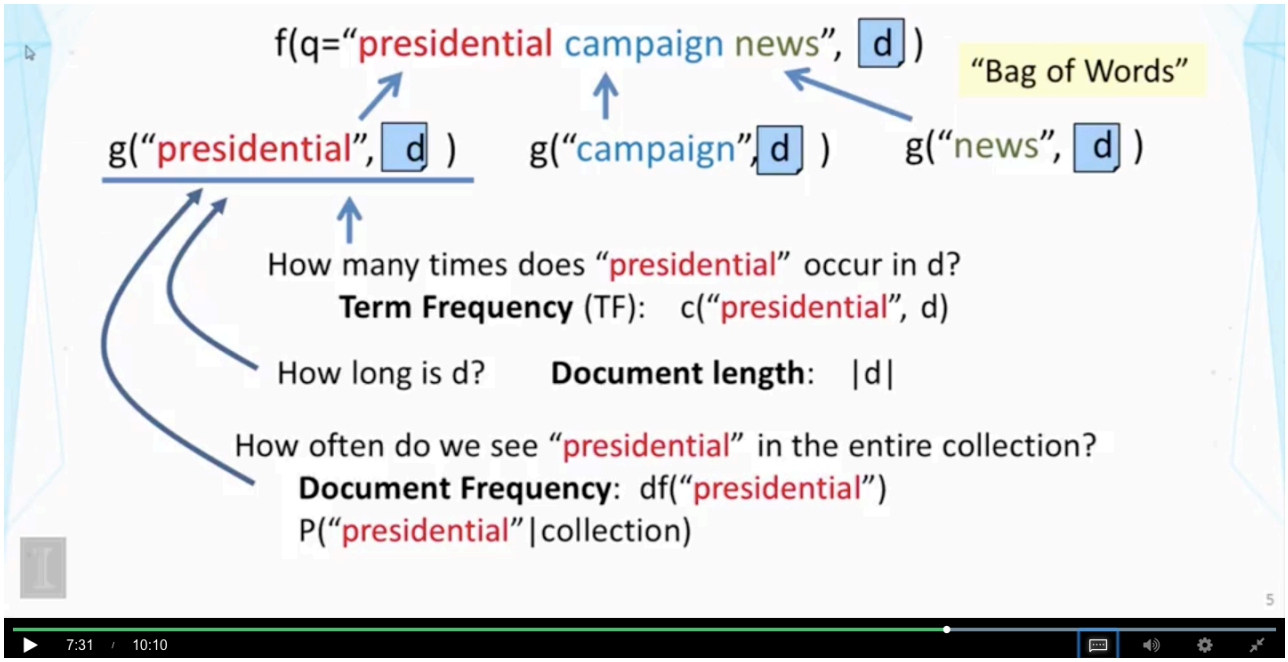
Ranking function - measure the likelihood that document 'd' is relevant to query 'q'
Retrieval model - provides a computational definition of relevance

Types of retrieval models:



State-of-the-Art Text Retrieval models:
Rely on:

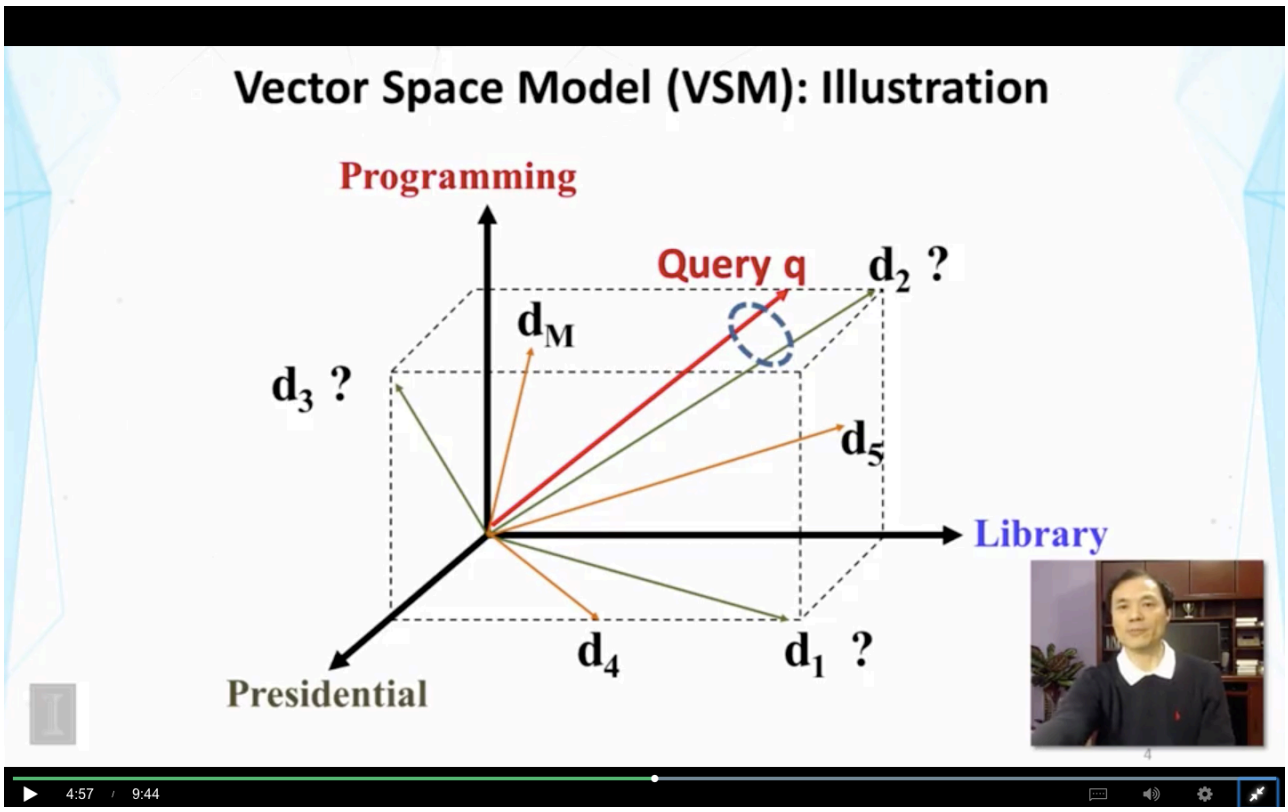
1. Bag of words representation
2. Term Frequency (TF) & Document Frequency (DF) of words
3. Document length



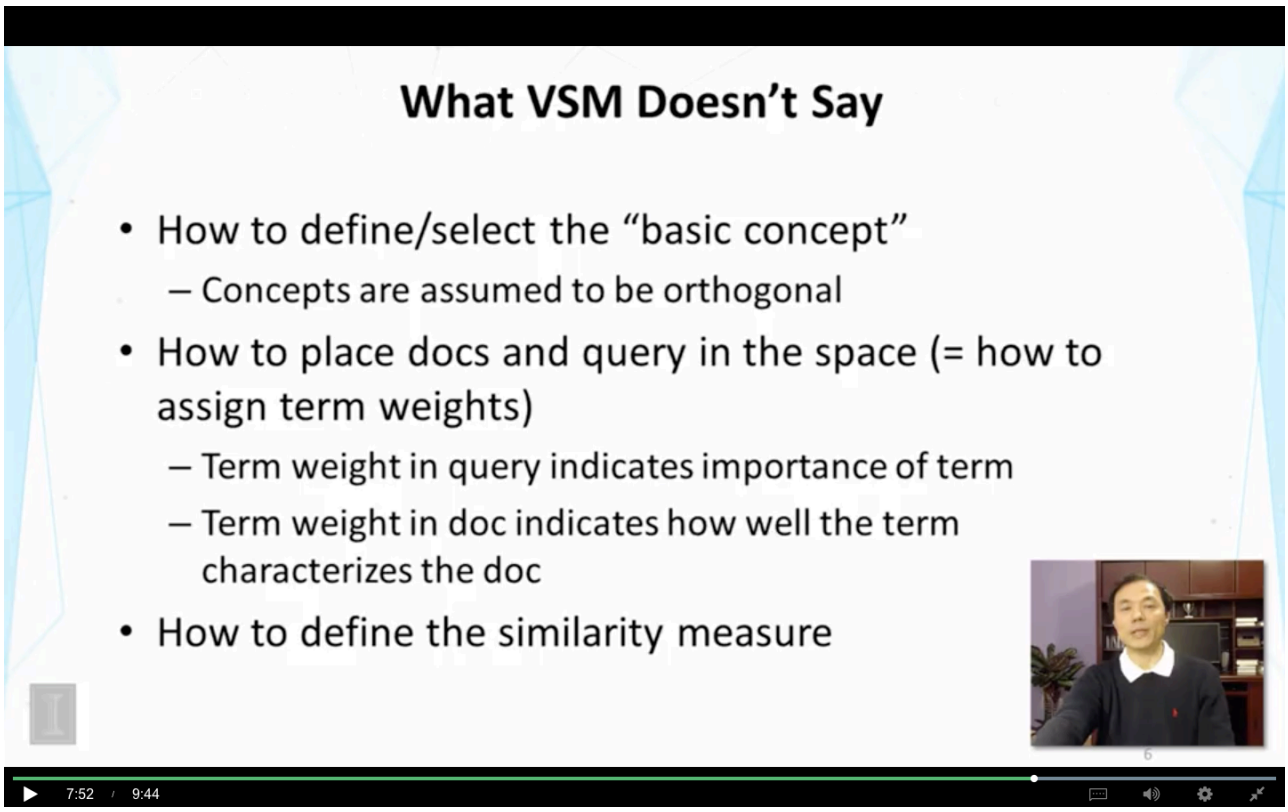
———————————————————————

Lecture 1.5 (Vector Space Model):

VSM is a framework
What is VSM ?



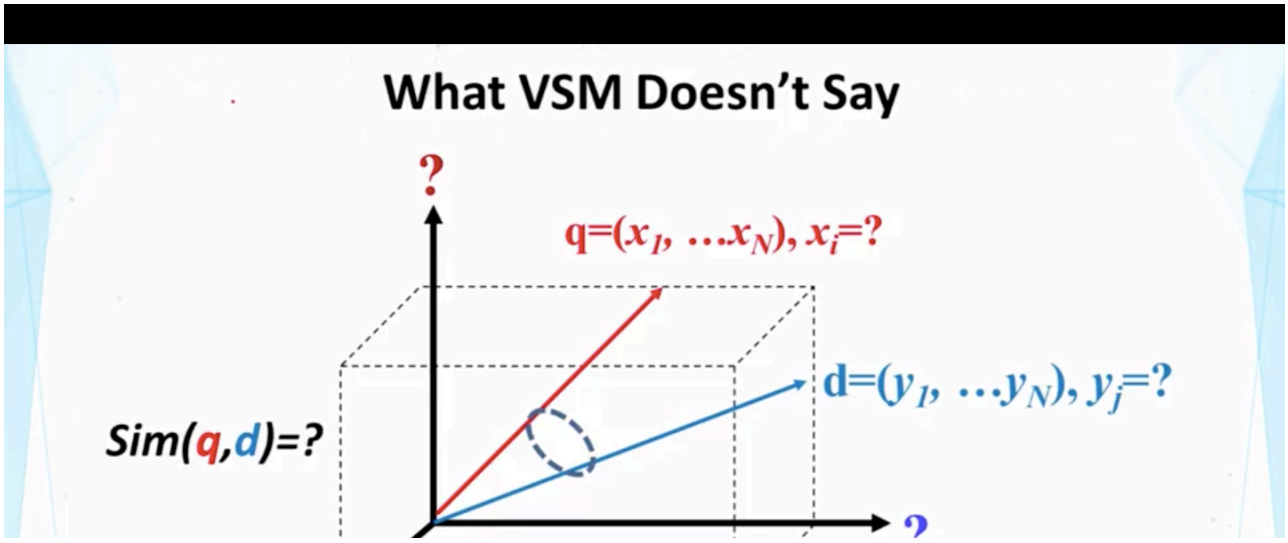
This basic idea of VSM leaves out lots of important mentions due to which this idea cannot be programmed yet.



Lecture 1.6 (VSM - simplest instantiation):

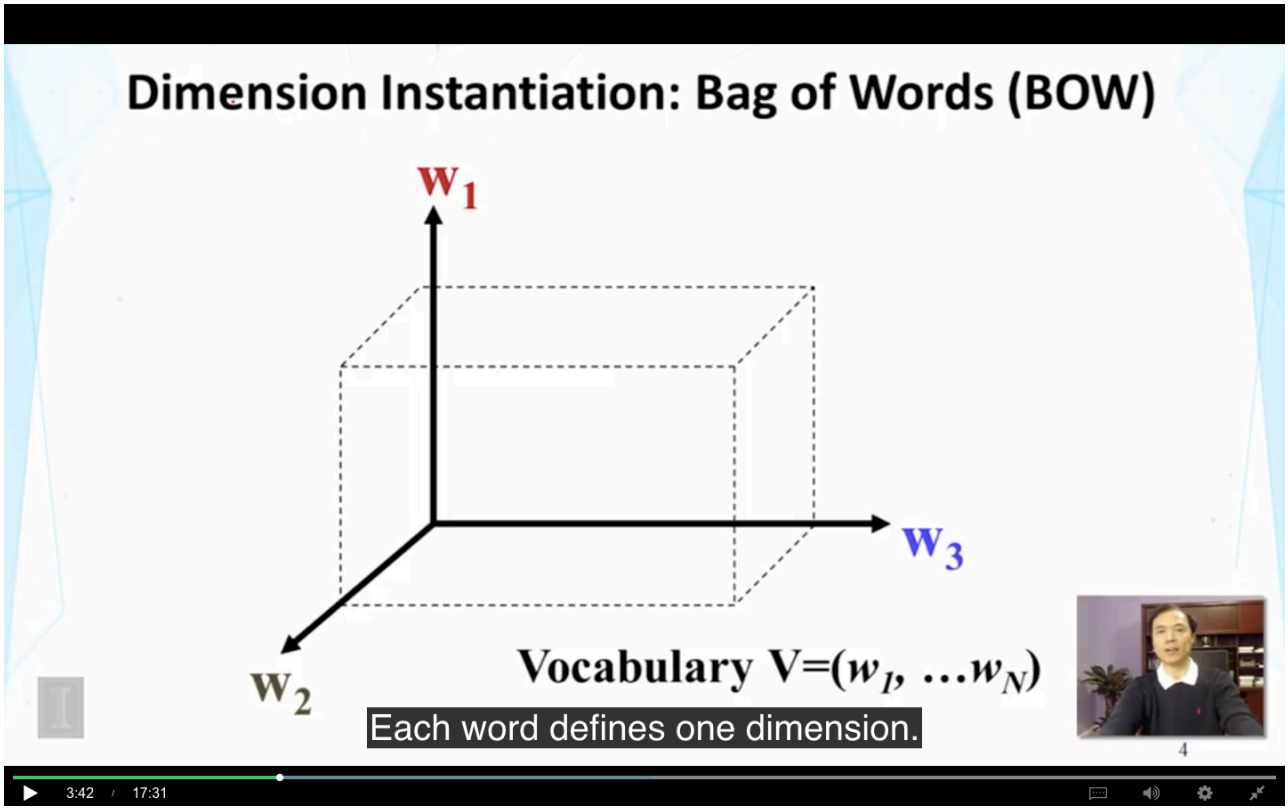
What VSM doesn't say:

1. How to represent dimensions of the model (Dimension instantiation)
2. How to represent the vectors in the space, what weights do we assign to the terms
3. How do we measure similarity between query vector 'q' and document vector 'd'

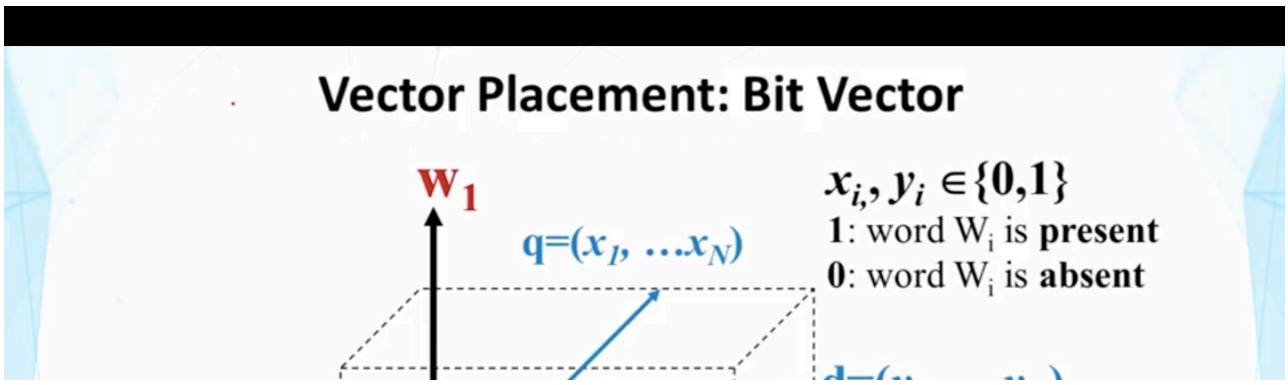


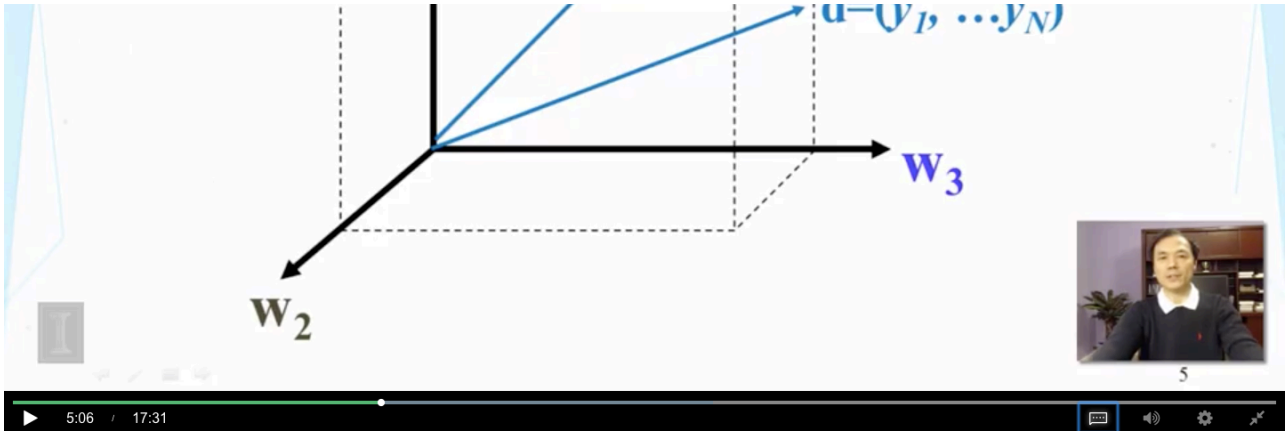
So, let's define these terms:

1. Dimension instantiation

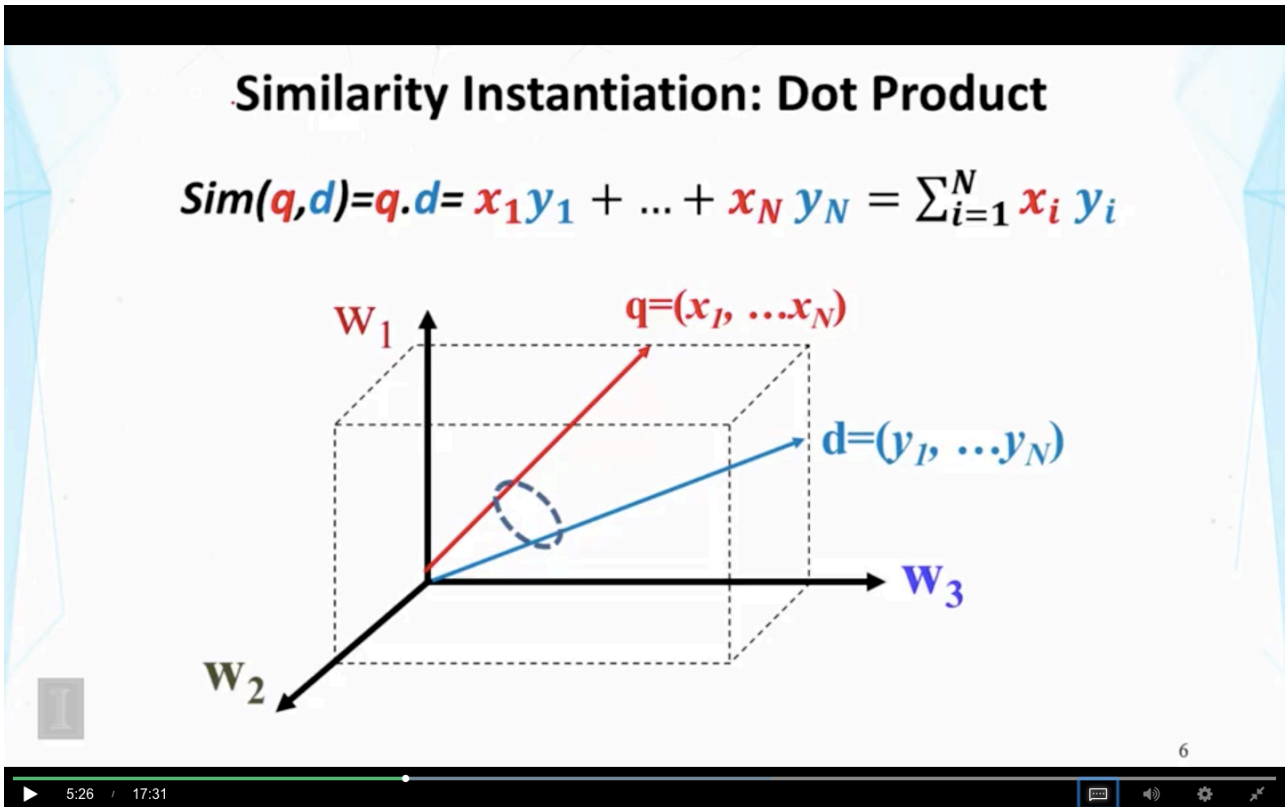


2. Vector placement: Bit vector used here; We can also use the **term frequency** (no. of occurrences) or **document frequency** (no. of documents in which these words occur) as a weight to assign these vectors. These might provide more information as to how the query and document contain these dimensions (words in vocabulary).

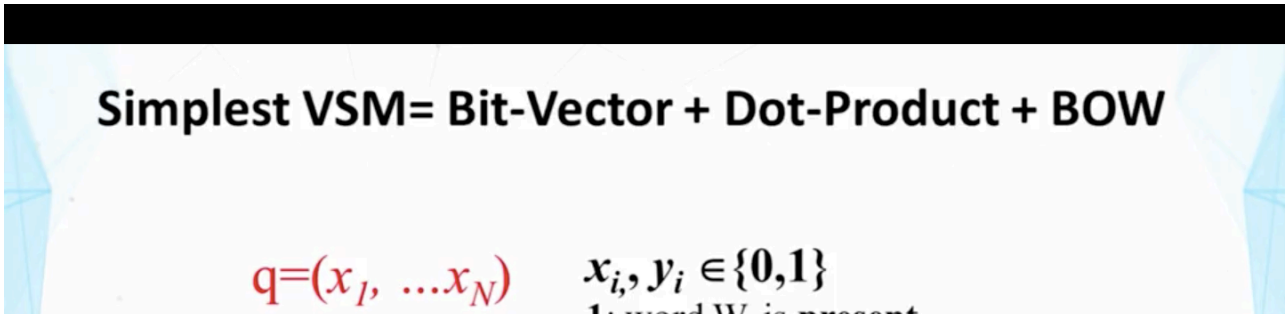


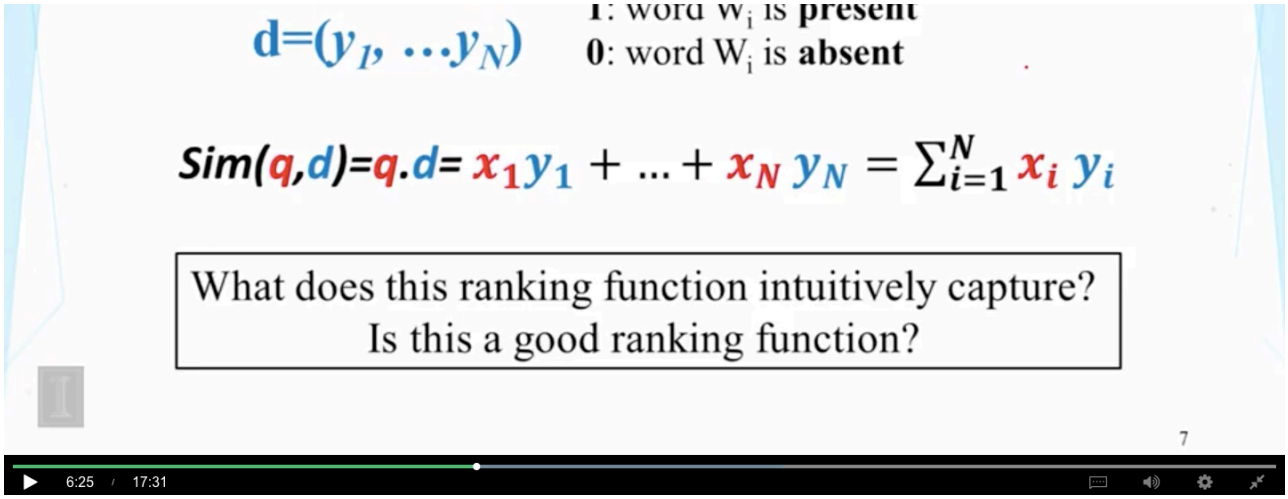


3. Similarity instantiation: Dot product used here - we can also use distance measures (Euclidean, etc.) to measure distance b/w 2 vectors. More distance, less similar are query and document vectors.

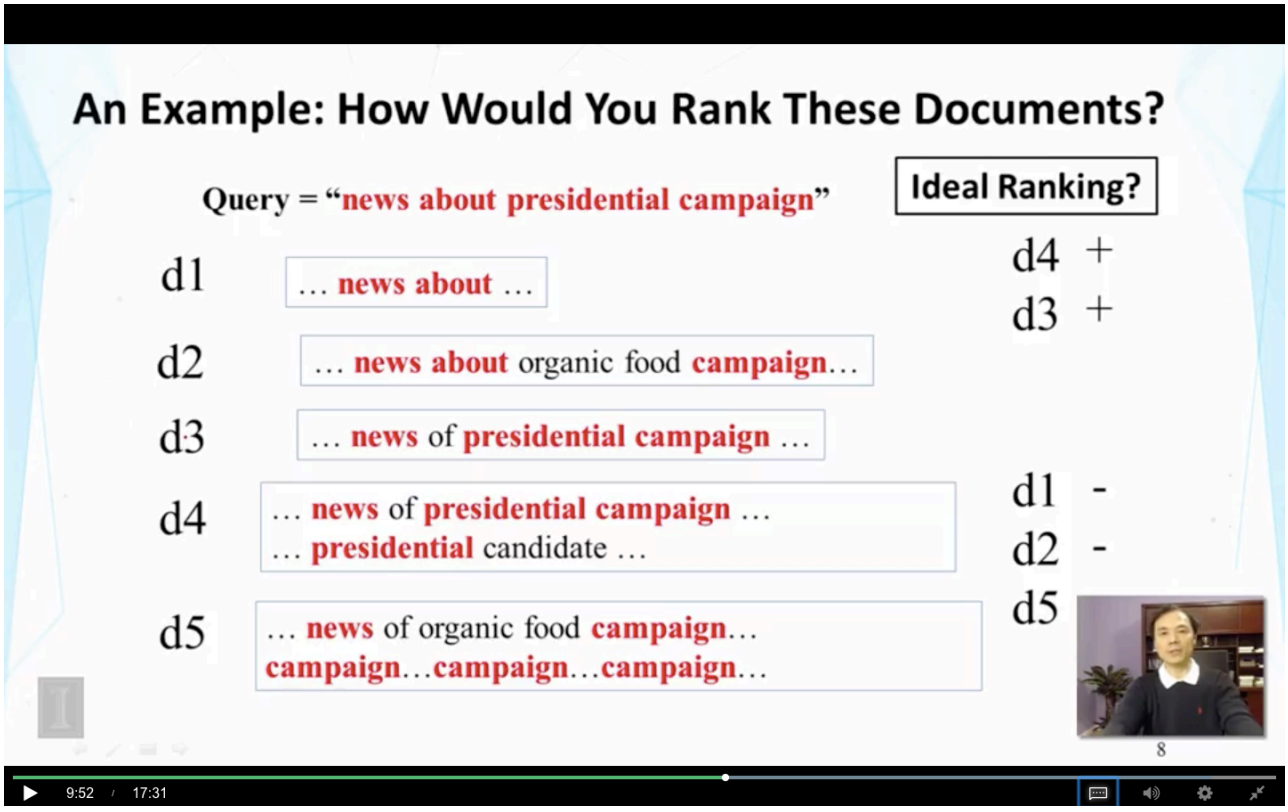


Simplest VSM instantiation:

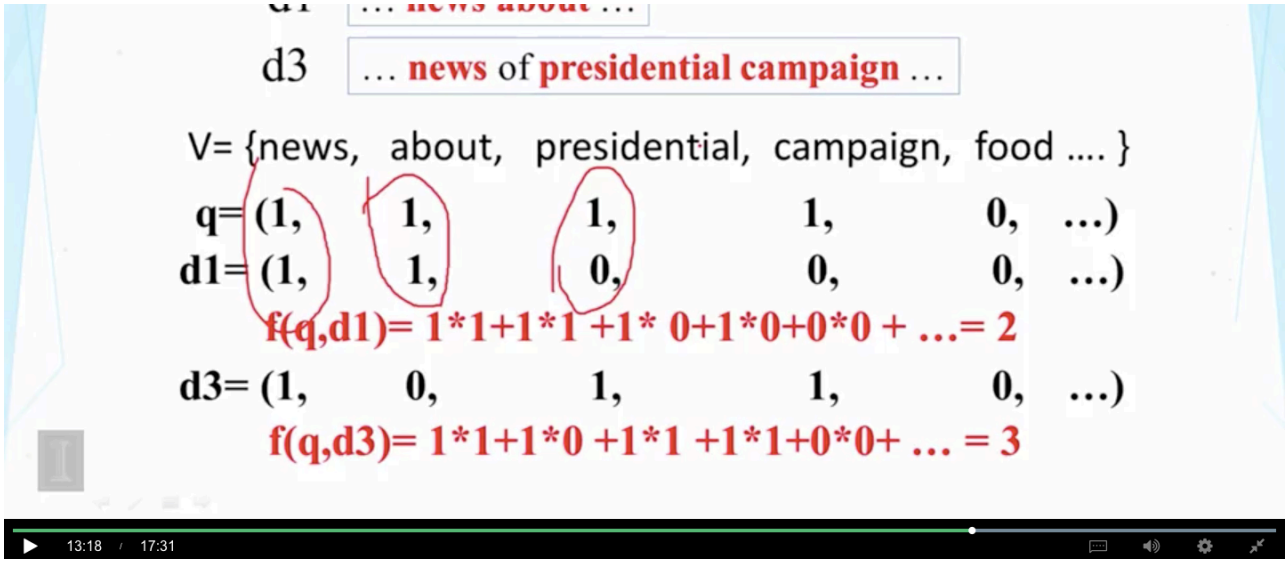




How do we say whether this ranking function is good or not? What does the value of this ranking function represent?
Example:

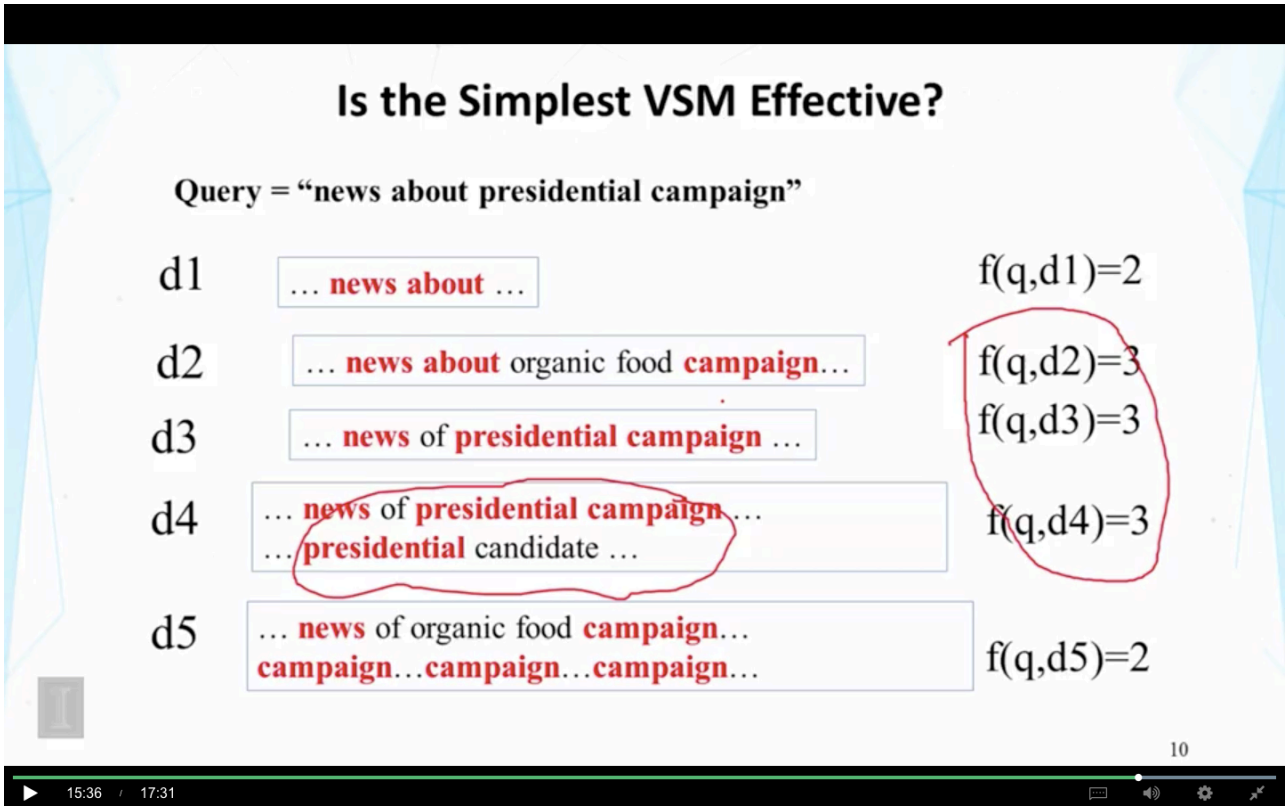


Let's see how our simplest instantiation of VSM does on matching this query with relevant documents.



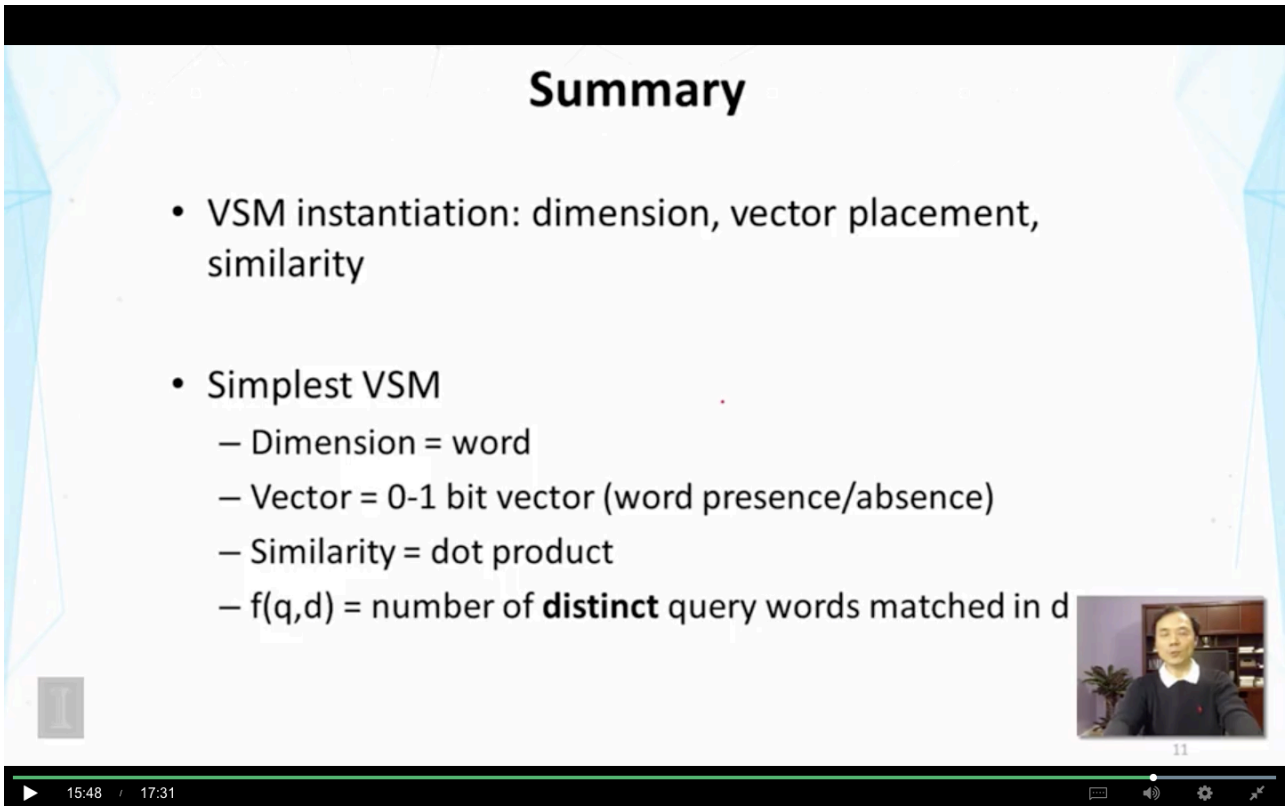
Observation: Whenever one of the words is not present in query or document, it's contribution to dot product is 0, meaning we are counting no. of 1–1 pairs.
So, the ranking function effectively tells how many unique words in query match with those in document.

With this knowledge, let's see if the simplest VSM is effective at ranking documents.



The **problem** is that this model ranks d2 above d3, because both have 3 matched words. But, we know that matching presidential is more important than matching about. So, there are still unresolved problems in this model.

Summary of simplest VSM:



————————X————————