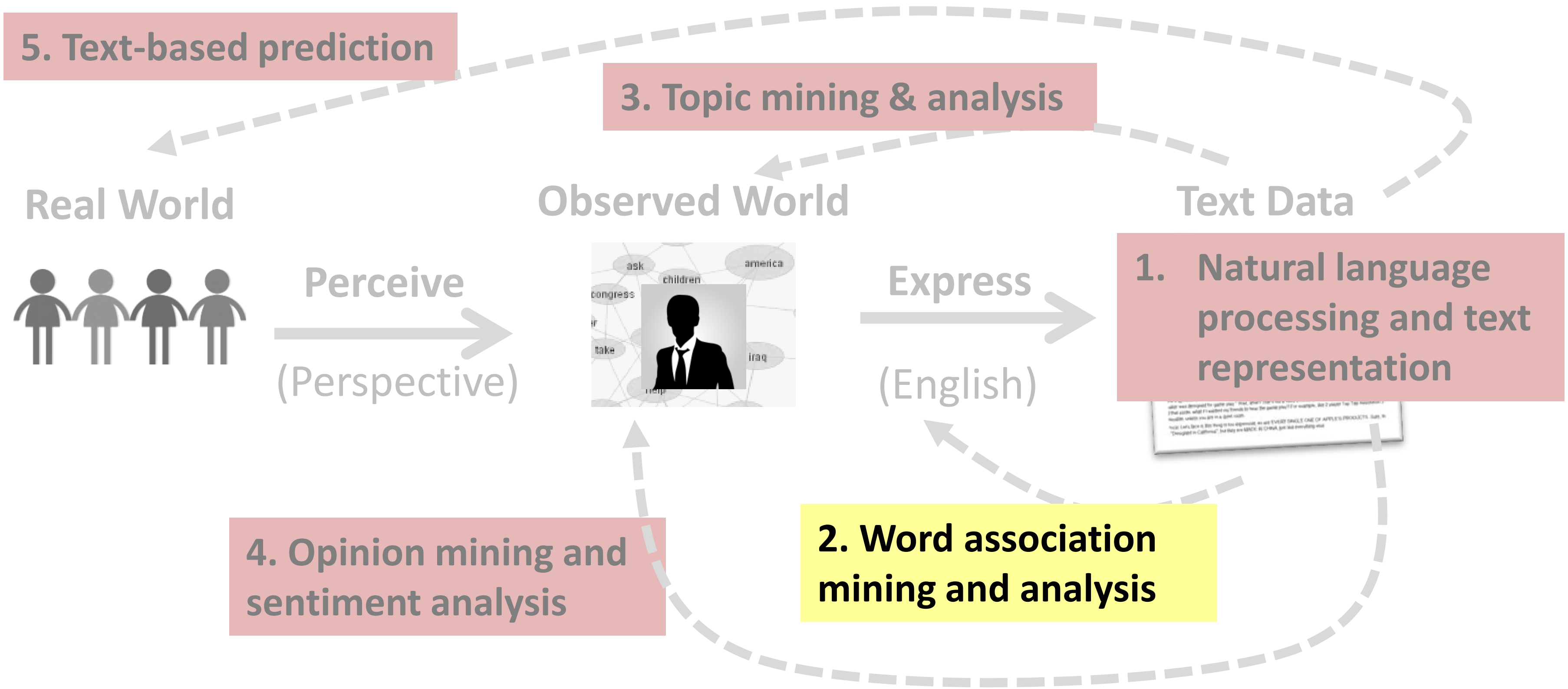
Paradigmatic Relation Discovery

Parts 1-3

ChengXiang "Cheng" Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

Paradigmatic Relation Discovery



Word Context as "Pseudo Document"

Left1("cat") = {"my", "his", "big", "a", "the",…}

cat:

My ___ eats fish on Saturday
His ___ eats turkey on Tuesday
…

Window8("cat") = {"my", "his", "big", "eats", "fish", …}

Right1("cat") = {"eats", "ate", "is", "has", ….}

语境

Context = pseudo document = "bag of words"
Context may contain adjacent or non-adjacent words

Measuring Context Similarity

Sim("Cat", "Dog") =

Sim(Left1("cat"), Left1("dog"))

+ Sim(Right1("cat"), Right1("dog")) +

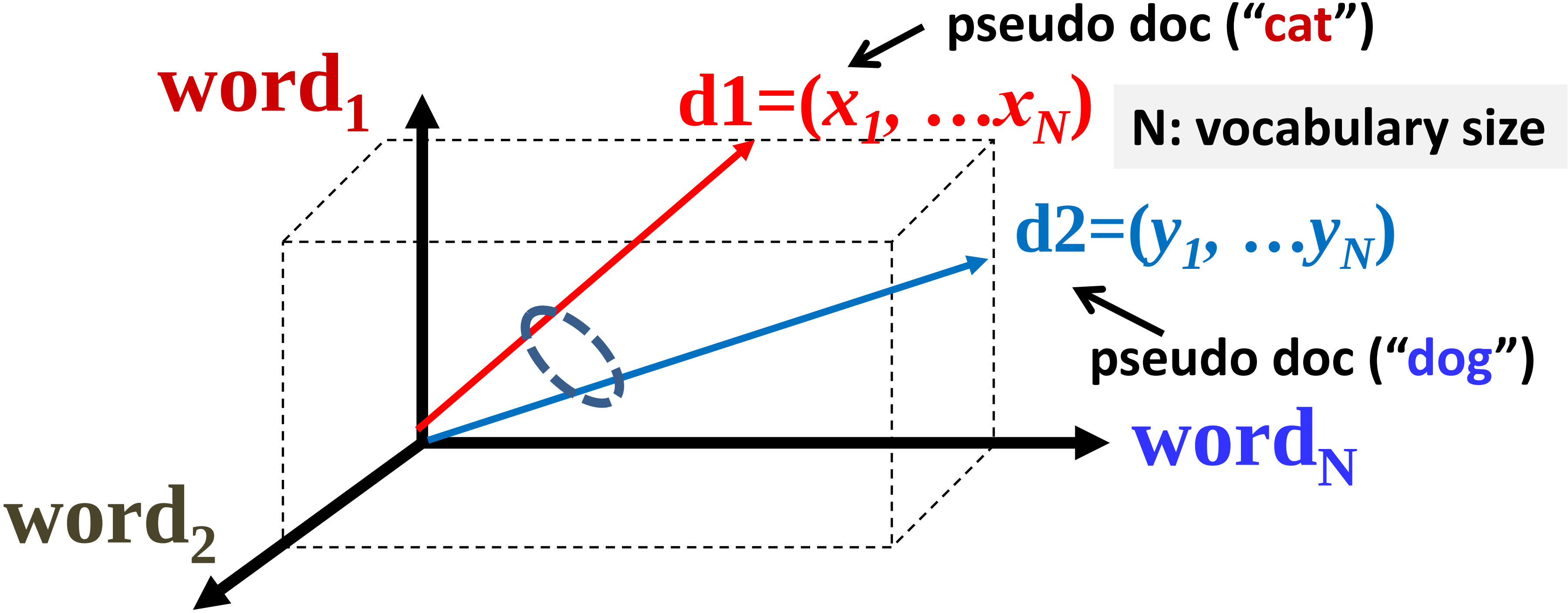
…

+ Sim(Window8("cat"), Window8("dog"))=?

High sim(word1, word2)

➔ word1 and word2 are **paradigmatically related**

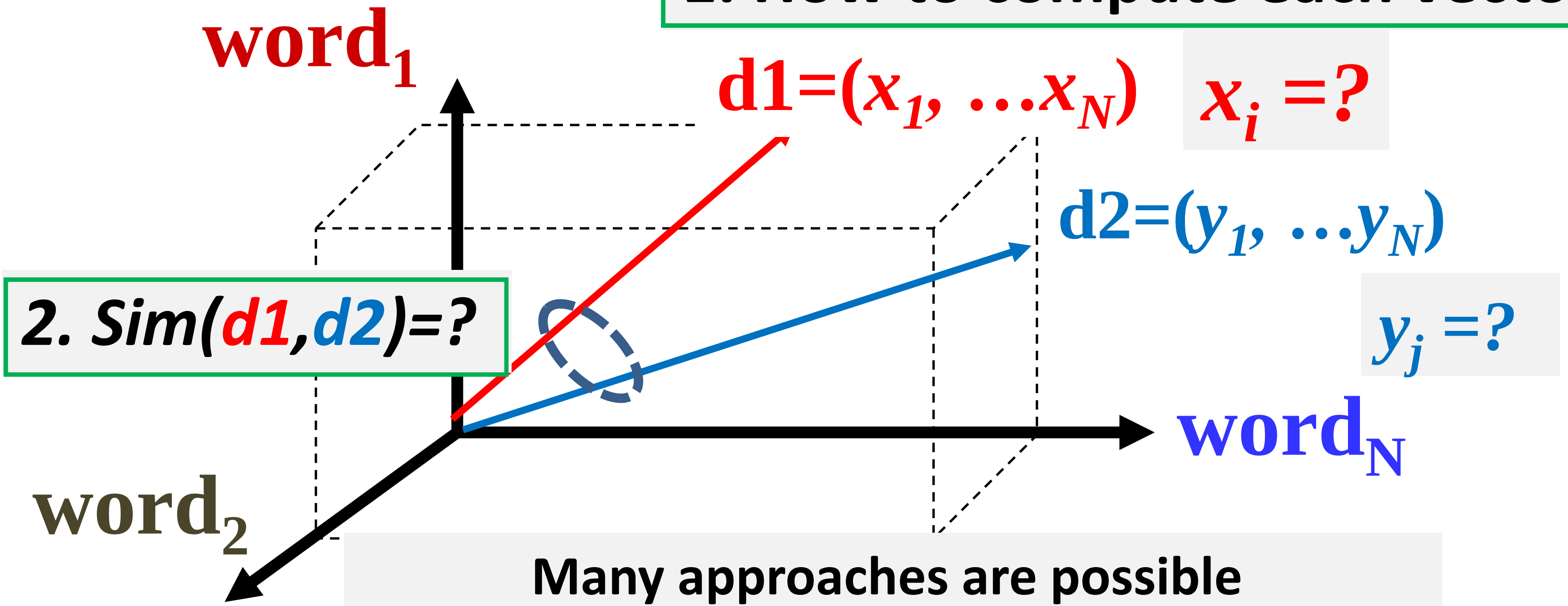
Bag of Words ➔ Vector Space Model (VSM)



Terms:		**"eats"**	**"ate"**	**"is"**	**"has"**	**....**
Vector:	**(**	**5,**	**3,**	**10,**	**3**	**....)**

VSM for Paradigmatic Relation Mining

1. How to compute each vector?



Many approaches are possible (most developed originally for text retrieval).

Expected Overlap of Words in Context (EOWC)

Probability that a randomly picked word from d1 is wi

Count of word wi in d1

$d1=(x_1, \ldots x_N)$ $\quad$ $x_i = c(w_i, d1)/|d1|$

$d2=(y_1, \ldots y_N)$ $\quad$ $y_i = c(w_i, d2)/|d2|$

Total counts of words in d1

$$Sim(d1,d2)=d1.d2= x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

Probability that two randomly picked words from d1 and d2, respectively, are identical.

Would EOWC Work Well?

- Intuitively, it makes sense: The more overlap the two context documents have, the higher the similarity would be.
- However:

  权重 or 贡献： 常见词 > 匹配特定词

  – It favors matching one frequent term very well over matching more distinct terms.

  每个词的权重相同，包括常见但没有太多意义的介词

  – It treats every word equally (overlap on “the” isn’t as so meaningful as overlap on “eats”).

Expected Overlap of Words in Context (EOWC)

Probability that a randomly picked word from d1 is wi

Count of word wi in d1

$$d1=(x_1, \ldots x_N) \qquad \mathbf{x_i = c(w_i, d1)/|d1|}$$

$$\mathbf{d2=(y_1, \ldots y_N)} \qquad y_i = c(w_i, d2)/|d2|$$

Total counts of words in d1

$$\mathbf{Sim(d1,d2)=d1.d2=} \; x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

Probability that two randomly picked words from d1 and d2, respectively, are identical.

Improving EOWC with Retrieval Heuristics

权重 or 贡献： 常见词 > 匹配特定词

- It favors matching one frequent term very well over matching more distinct terms.

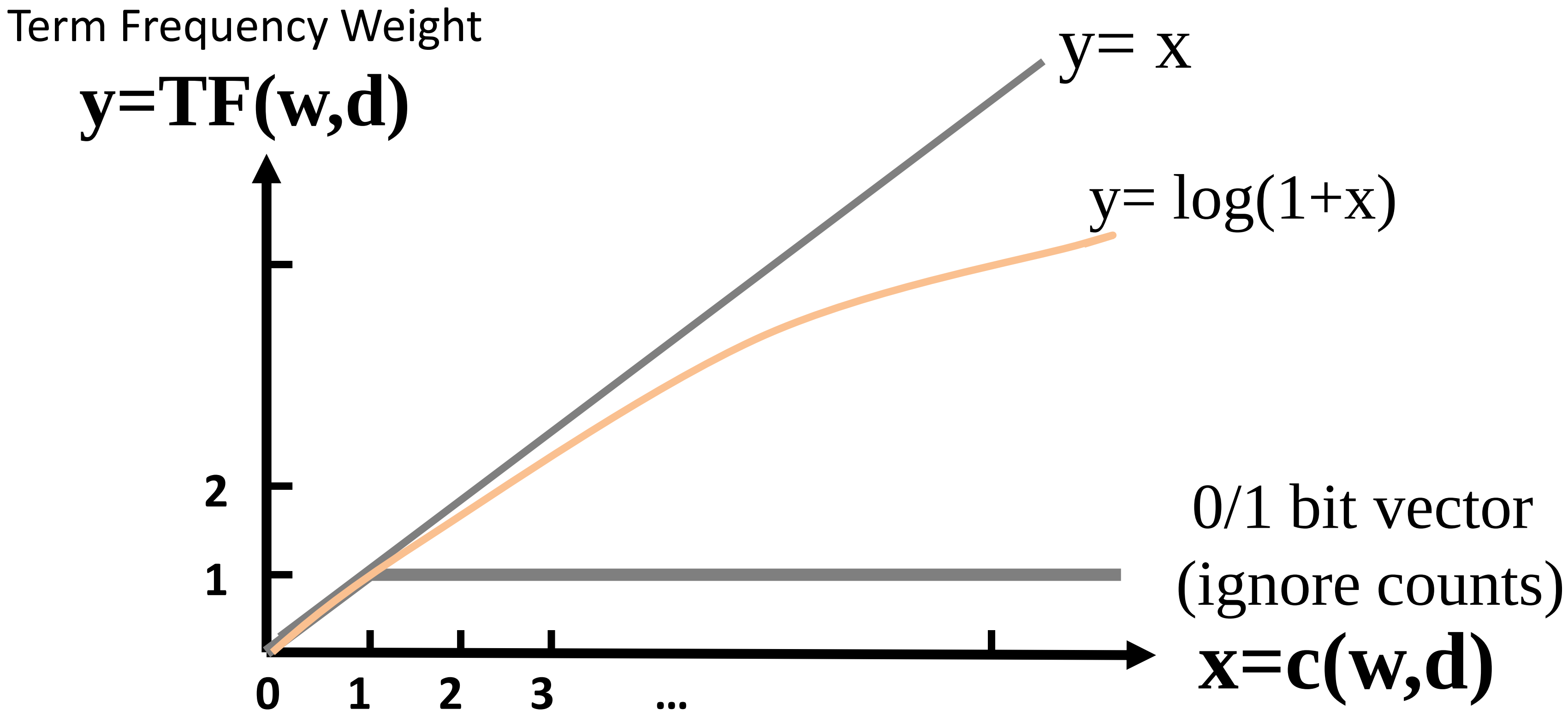
➔ **Sublinear transformation of Term Frequency (TF)**

每个词的权重相同，包括常见但没有太多意义的介词

- It treats every word equally (overlap on “the” isn’t as so meaningful as overlap on “eats”).

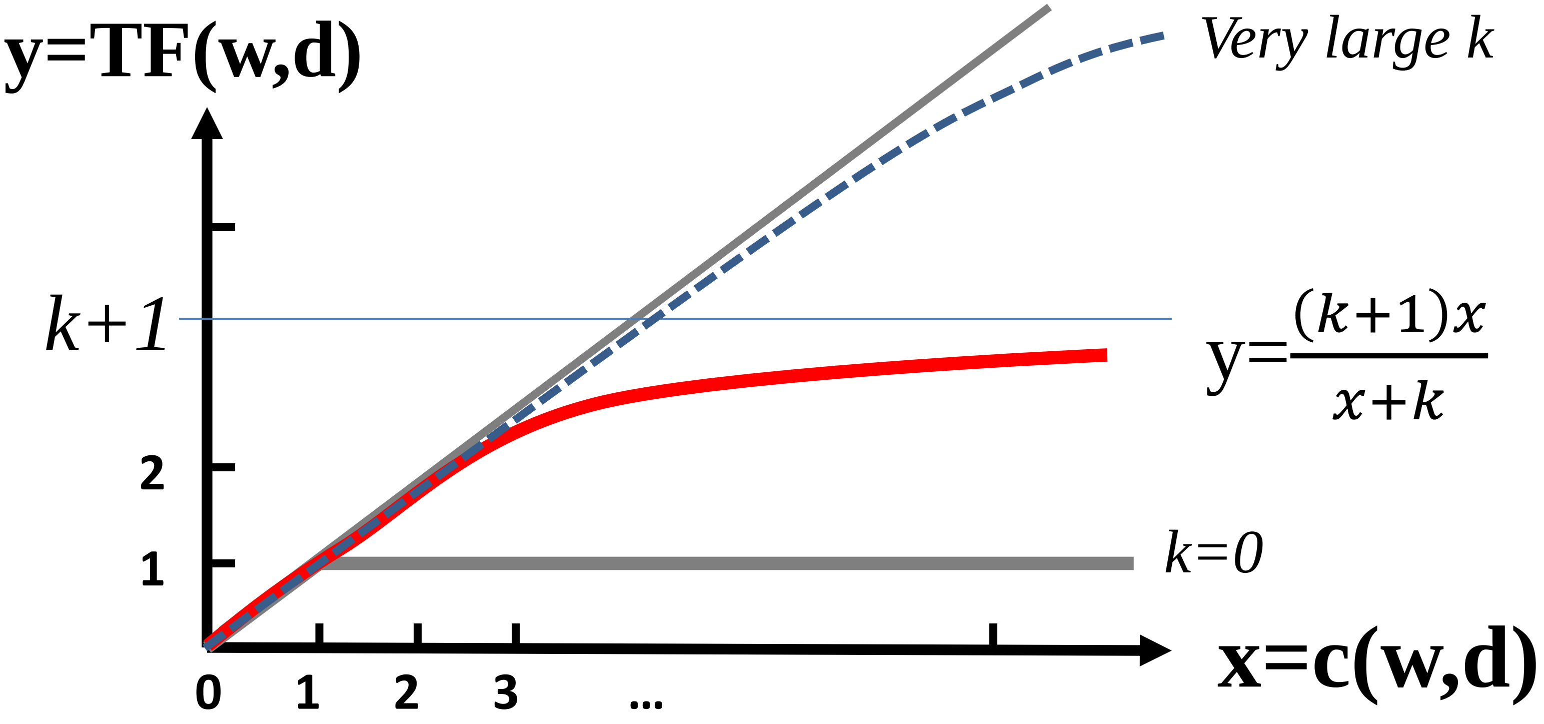
➔ **Reward matching a rare word: IDF term weighting**

TF Transformation: c(w,d)➔TF(w,d)



TF Transformation: BM25 Transformation

Term Frequency Weight 减小高频词的贡献



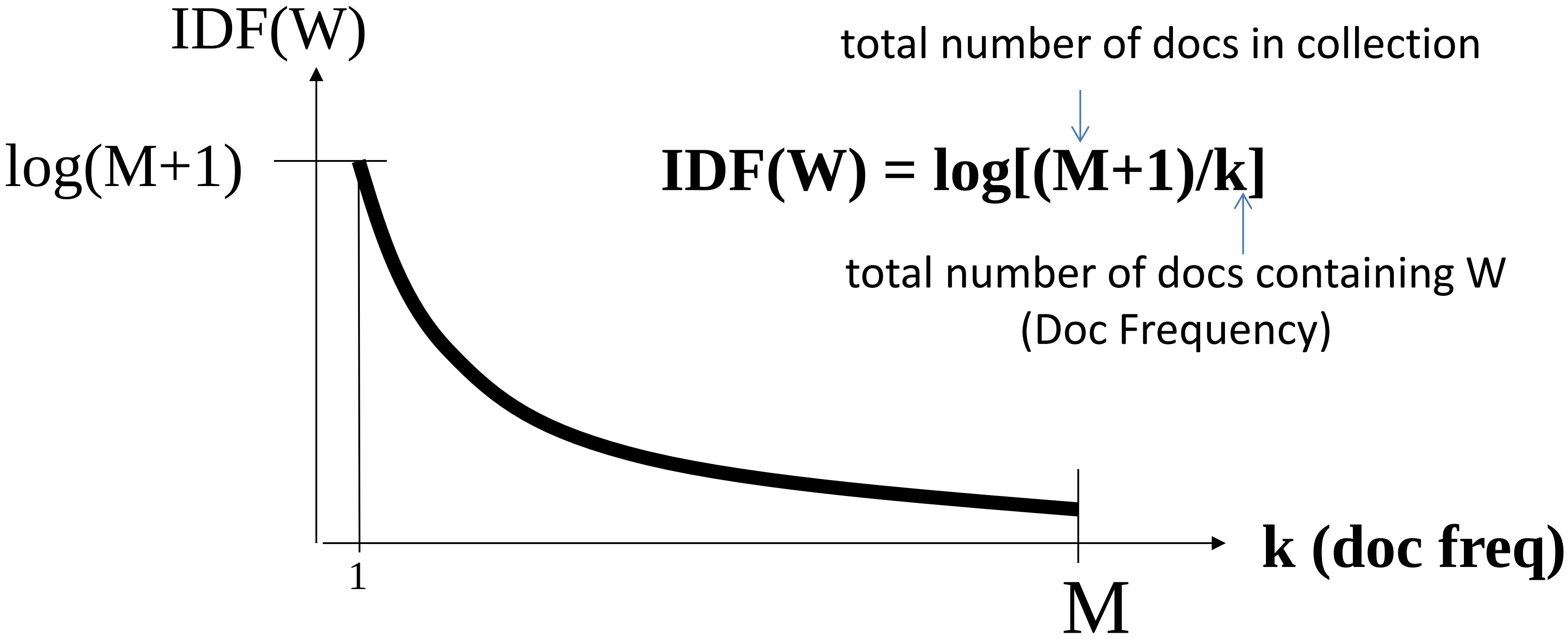
IDF Weighting: Penalizing Popular Terms

"惩罚"常见词

total number of docs in collection

$$\mathbf{IDF(W) = log[(M+1)/k]}$$

total number of docs containing W
(Doc Frequency)



Adapting $\boxed{\text{BM25 Retrieval}}$ Model for Paradigmatic Relation Mining

(k+1)x/(k+x)

$d1=(x_1, \ldots x_N)$

$$BM25(w_i, d1) = \frac{(k+1)c(w_i, d1)}{c(w_i, d1) + k\underline{(1-b+b*|d1|/avdl)}}$$

$$x_i = \frac{BM25(w_i, d1)}{\sum_{j=1}^{N} BM25(w_j, d1)}$$

$b \in [0,1]$

$k \in [0,+\infty)$

$d2=(y_1, \ldots y_N)$

y_i is defined similarly

$$Sim(d1,d2)=\sum_{i=1}^{N} \boxed{IDF(w_i)}\, x_i\, y_i$$

BM25 can also Discover Syntagmatic Relations

$d1=(x_1, \ldots x_N)$

$$BM25(w_i, d1) = \frac{(k+1)c(w_i, d1)}{c(w_i, d1) + k(1 - b + b * |d1| / avdl)}$$

$$x_i = \frac{BM25(w_i, d1)}{\sum_{j=1}^{N} BM25(w_j, d1)}$$

$b \in [0,1]$

$k \in [0,+\infty)$

IDF-weighted $d1=(x_1 * IDF(w_1), \ldots, x_N * IDF(w_N))$

The highly weighted terms in the context vector of word w are likely syntagmatically related to w.

Summary

- Main idea for discovering paradigmatic relations:
  - Collecting the context of a candidate word to form a pseudo document (bag of words)
  - Computing similarity of the corresponding context documents of two candidate words
  - Highly similar word pairs can be assumed to have paradigmatic relations
- Many different ways to implement this general idea
- Text retrieval models can be easily adapted for computing similarity of two context documents
  - BM25 + IDF weighting represents the state of the art
  - Syntagmatic relations can also be discovered as a "by product"