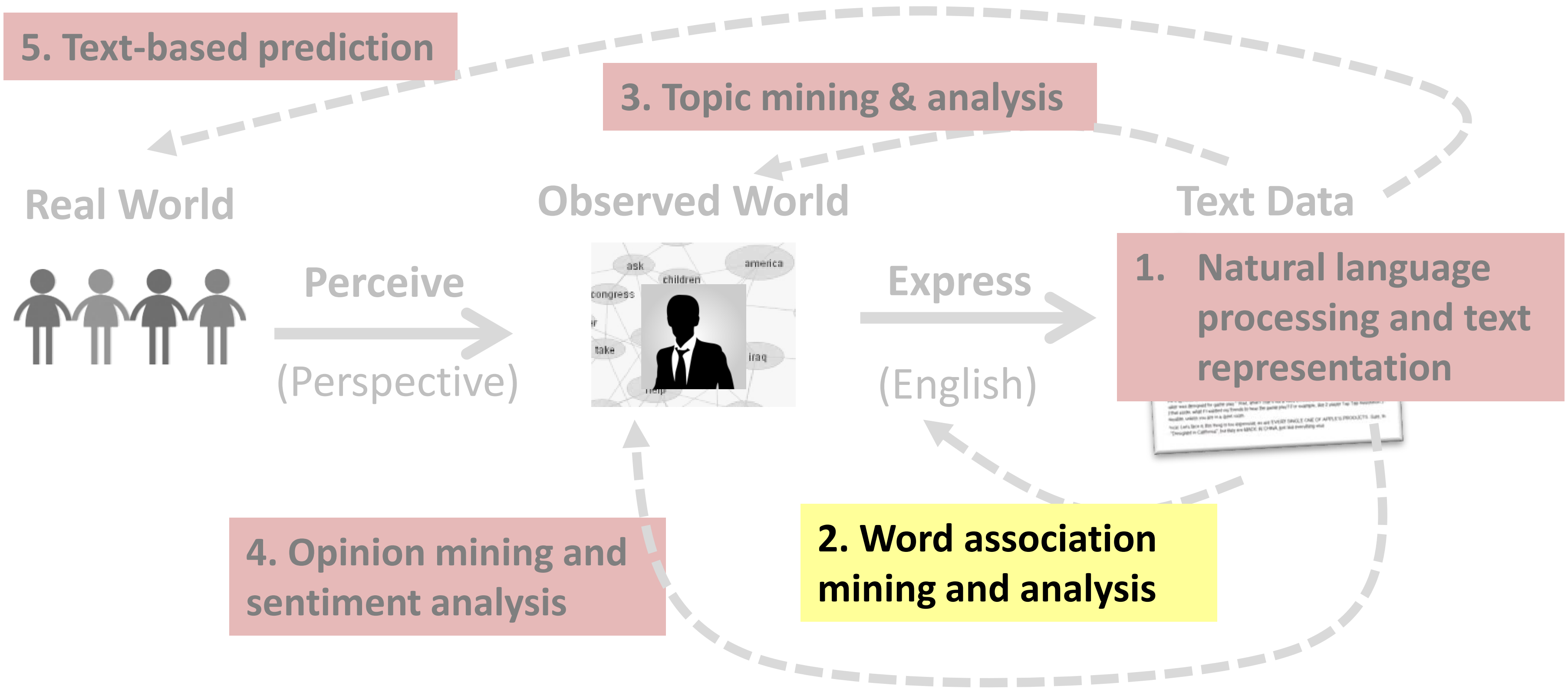
Paradigmatic Relation Discovery

Parts 1-3

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

Paradigmatic Relation Discovery



Word Context as “Pseudo Document”

1 **Left1(“cat”) = {“my”, “his”, “big”, “a”, “the”,…}**

[illegible]所words在cat周围.

cat:

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

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

2 **Right1(“cat”) = {“eats”, “ate”, “is”, “has”, ….}**

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

Measuring Context Similarity

计算2词的相似度.

Sim("Cat", "Dog") =

Combine all the perspective

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

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

…

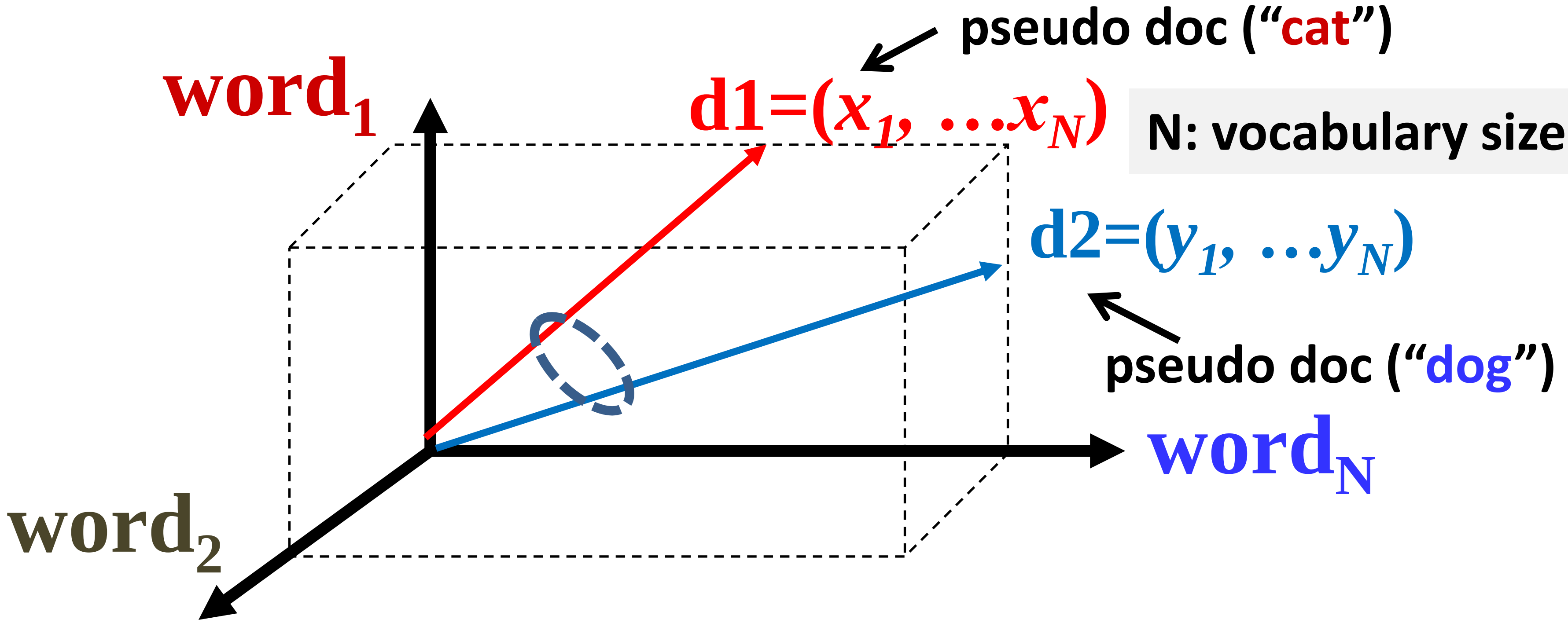
+ Sim(Window8("cat"), Window8("dog"))=?

2词相似度很高 推出 两者有聚合关系.

High sim(word1, word2)

➔ word1 and word2 are **paradigmatically related**

Bag of Words ➔ Vector Space Model (VSM)

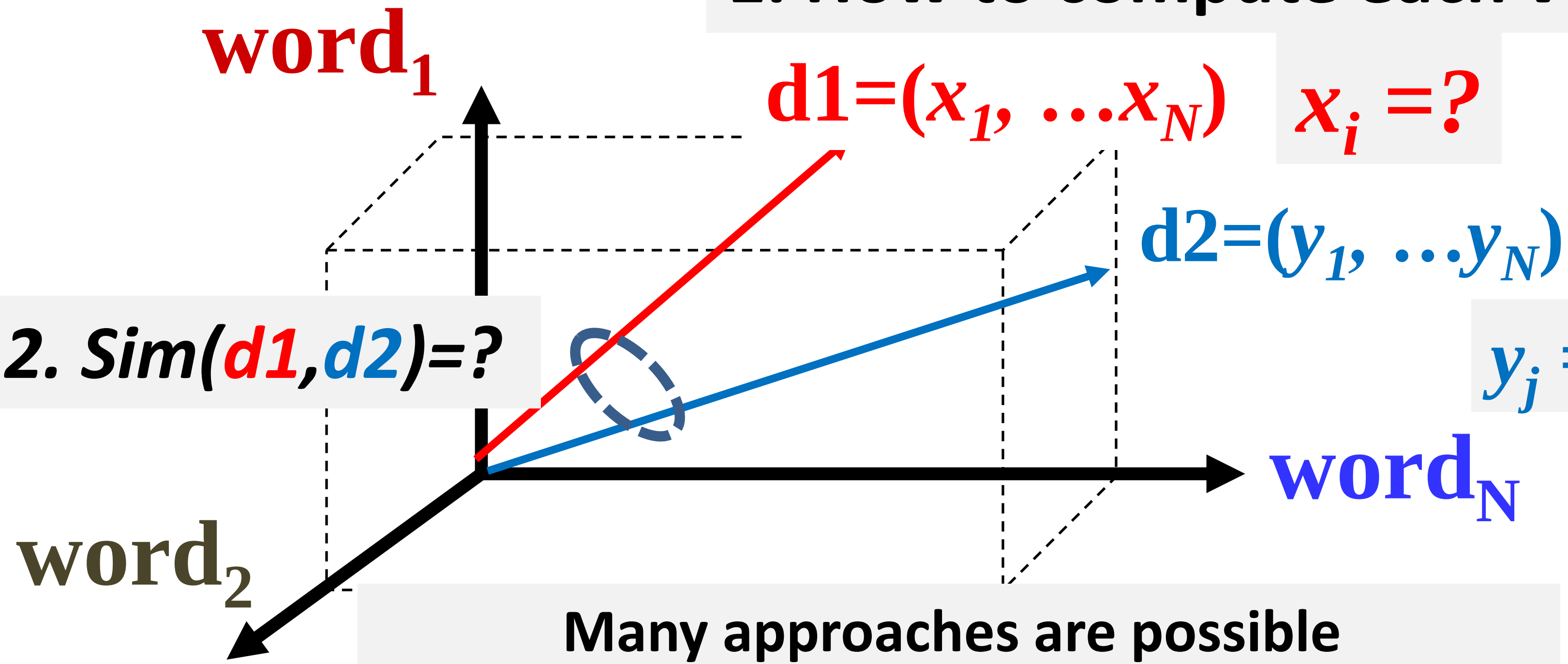


N: vocabulary size

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

VSM for Paradigmatic Relation Mining

1. How to compute each vector?



x_i =?

y_j =?

2. *Sim*(*d1*,*d2*)=?

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 每个词的权重是：随机从d1从选取一个词是当前词的概率。也就是词频。

Count of word wi in d1

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

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

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. if they are 聚合关系

are not identical. if they are not 聚合关系.

Would EOWC Work Well?

EOWC核心思想

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

缺点

  - It favors matching one frequent term very well over matching more distinct terms. 对于出现次数少的词不友好.
  - It treats every word equally (overlap on “the” isn’t as so meaningful as overlap on “eats”).

这种词就算有 overlap 也对我们评价文本也很无用

Expected Overlap of Words in Context (EOWC)

Probability that a randomly picked word from d1 is wi

Count of word wi in d1

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

Total counts of words in d1

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

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

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

Improving EOWC with Retrieval Heuristics

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

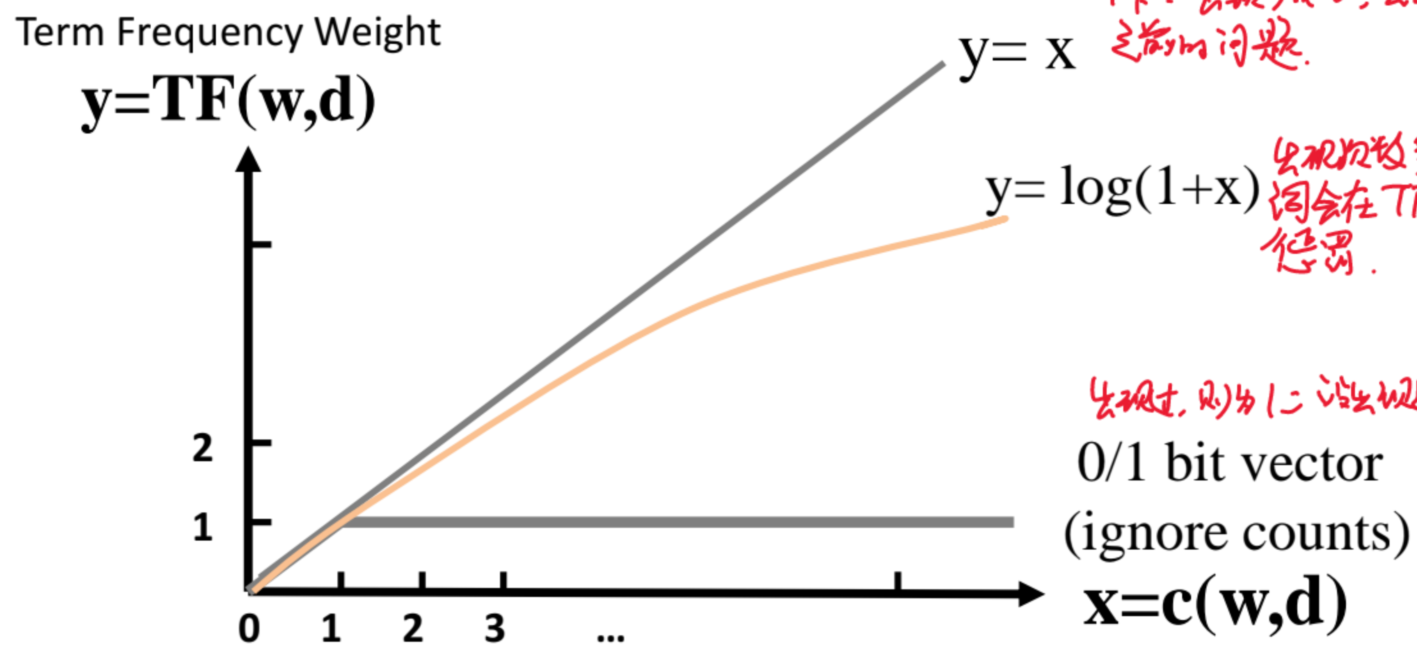
  ➔ **Sublinear transformation of Term Frequency (TF)**

- 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**

put more weight on rare word

TF Transformation: c(w,d)➔TF(w,d)

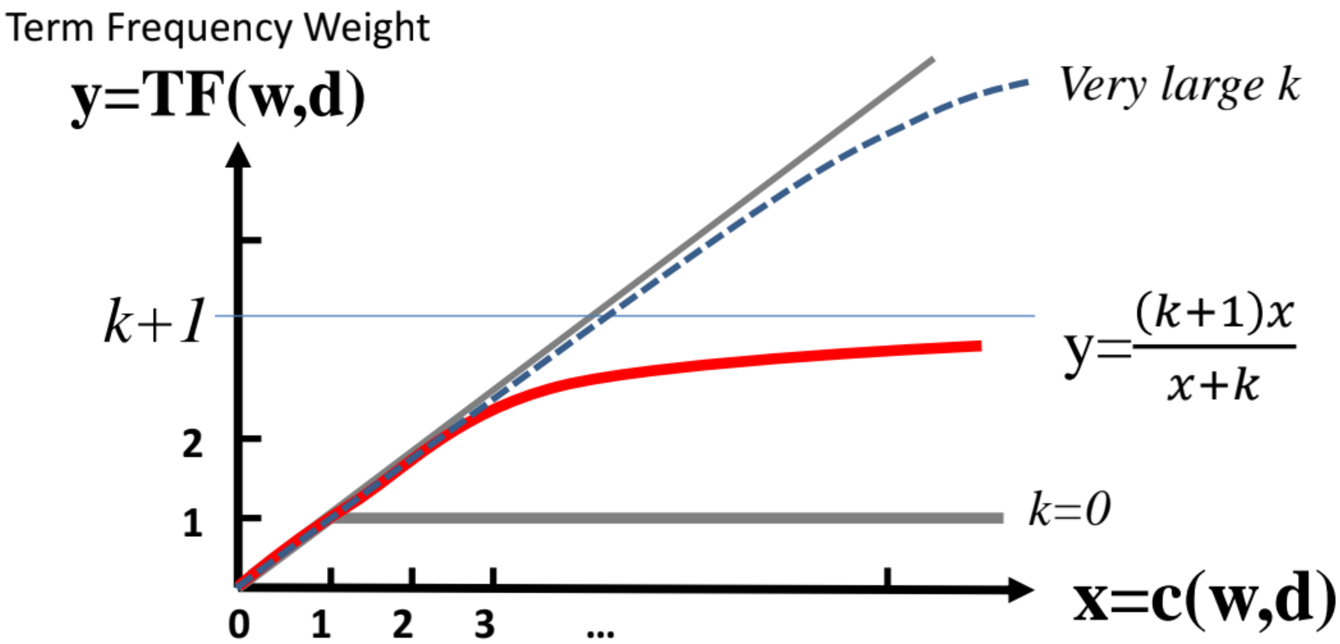


TF：出现次数，会出现之前的问题.

出现次数多的词会在TF上有惩罚.

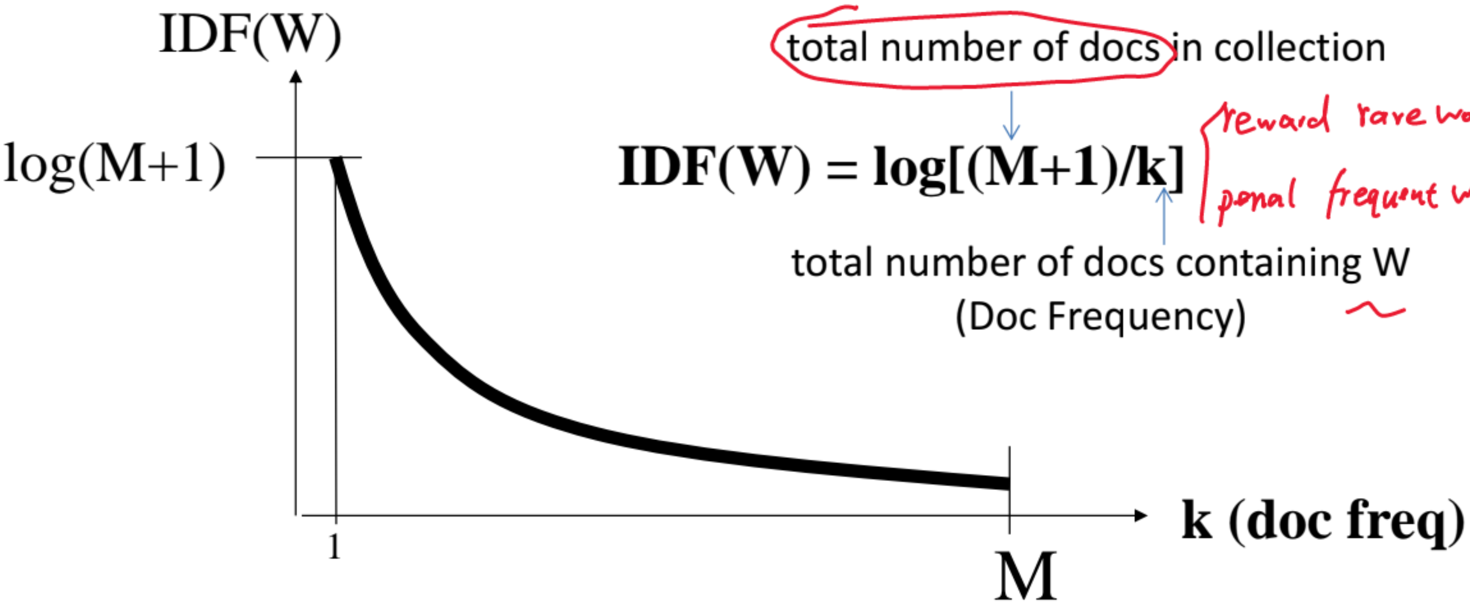
出现过，则为1；没出现过，则为0.

TF Transformation: BM25 Transformation



上界为k+1.

IDF Weighting: Penalizing Popular Terms



total number of docs in collection

$$\mathbf{IDF(W) = \log[(M+1)/k]} \left\{ \begin{array}{l} \text{reward rare word} \\ \text{penal frequent word} \end{array} \right.$$

total number of docs containing W
(Doc Frequency)

Adapting BM25 Retrieval Model for Paradigmatic Relation Mining

当前doc长度

平均长度 for all docs

d1=(x_1, …x_N)

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

控制长度的 normalization

normalized,
为了 $\sum_{i=1}^{N} x_i = 1$

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

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

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

d2=(y_1, …y_N)

y_i is defined similarly

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

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 $d1=(x_1{*}IDF(w_1), \ldots, x_N{*}IDF(w_N))$

使用了IDF 从而 综合了 词频以及 词的稀有度.

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" 副作用,