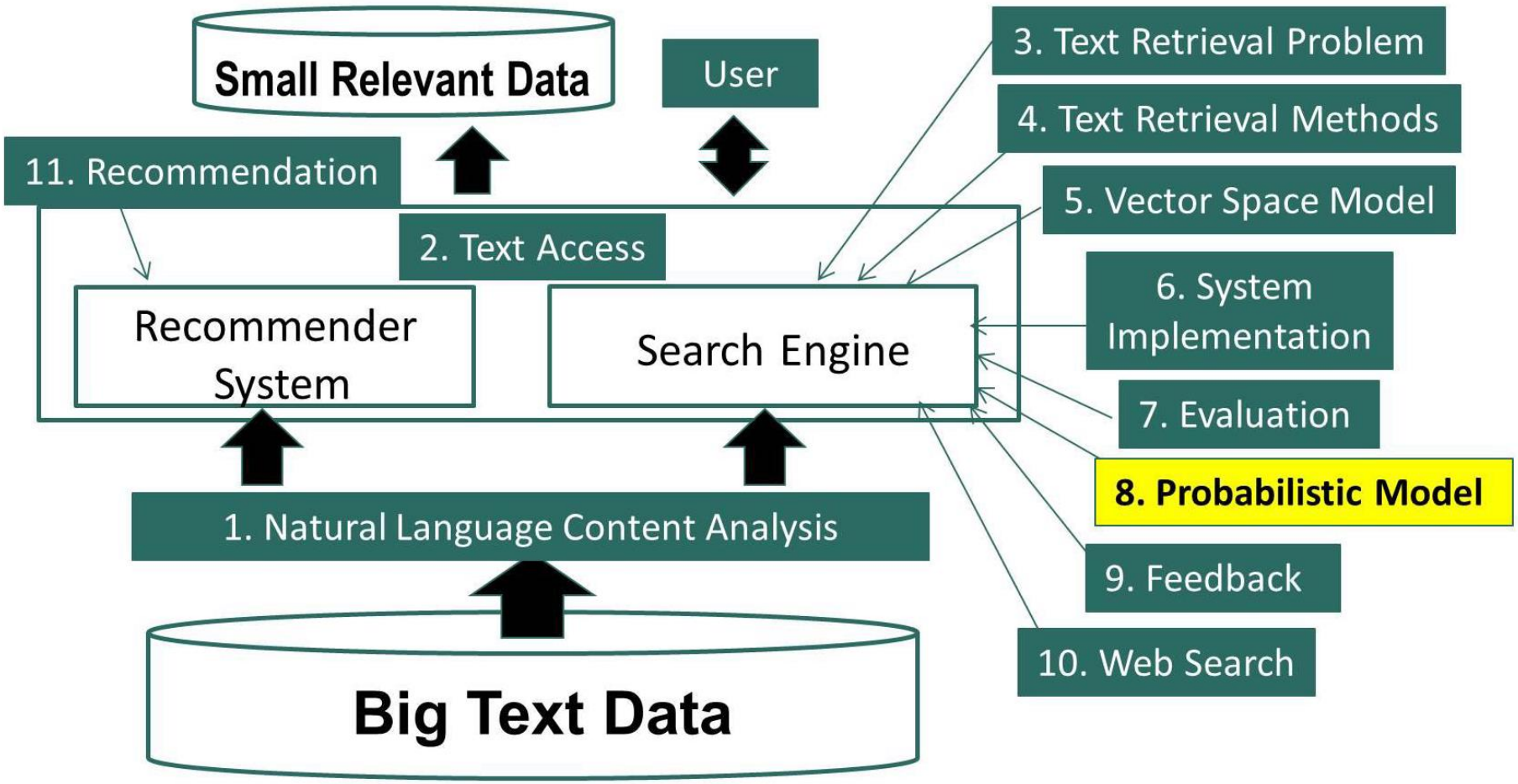
מודל אחזור הסתברותי


Small Relevant Data
User
3. Text Retrieval Problem
4. Text Retrieval Methods
11. Recommendation
5. Vector Space Model
2. Text Access
6. System Implementation
Recommender System
Search Engine
7. Evaluation
8. Probabilistic Model
1. Natural Language Content Analysis
9. Feedback
10. Web Search
Big Text Data

מהו מודל שפתי סטטיסטי (Language Model LM)?

- התפלגות הסתברות על פני רצפים של מילים
  - p(“*Today is Wednesday*”) ≈ 0.001
  - p(“*Today Wednesday **is***”) ≈ 0.0000000000001 משפט לא תקין..
  - p(“*The eigenvalue is positive*”) ≈ 0.00001
- ההתפלגות תלוייה בהקשר
  - בהחלט ייתכן אם מדובר על דומיין של מתמטיקה "*The eigenvalue is positive* " יהיה בעל הסתברות גבוהה יותר מאשר "*Today is Wednesday*"
- ניתן גם להתייחס לזה כמנגנון הסתברותי המחולל (generate) טקסט, מסיבה זו הוא נקראה גם מודל "ג'נרטיבי"

Today is Wednesday

Today Wednesday is

...

The eigenvalue is positive

היעילות של ה- LM - Language Model

- מכמת את אי הוודאות בשפה הטבעית
- מאפשר לנו לענות על כל מיני שאלות, כגון:
  - בהינתן שאנו רואים את "John" ו- "feels" מה הסבירות שנראה "happy" לעומת "habit" כמילה הבאה?

    (זיהוי דיבור, המילים נשמעות דומה)
  - בהינתן שאנו רואים במאמר חדשותי את המילה "בייסבול" שלוש פעמים ואת המילה "משחקים" פעם אחת, מה הסבירות שמדובר ב- "ספורט"?

    (אחזור מידע, קיטלוג מסמכים)
  - בהינתן שמשתמש מעוניין בחדשות ספורט, באיזו סבירות המשתמש ישתמש במילה "בייסבול" בשאילתה?

    (אחזור מידע)

מודל שפתי פשוט: יוניגראם מודל

בר כי שימושי

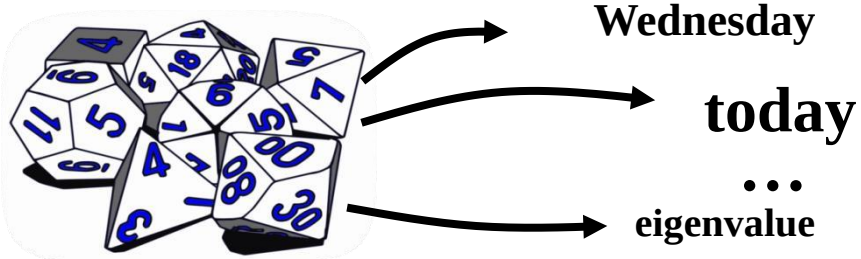
- צור טקסט על ידי התייחסות לכל מילה באופן **עצמאי** (דבר שאינו נכון בעולם האמתי, יש קשר בין המילים במשפט)

→ $p(w_1\ w_2 \ldots w_n)=p(w_1)p(w_2)\ldots p(w_n)$

Parameters: $\{p(w_i)\}$ $p(w_1)+\ldots+p(w_N)=1$ (N is voc. size)

- טקסט הוא דגימה של מילים מהתפלגות מסויימת

הגרלה...



בסתברות המשפט יהפוך המילים בדקות

p("today is Wed")

= p("today")p("is")p("Wed")

= 0.0002 x 0.001 x 0.000015

סוב של שמש בתוך המילים

יצירת טקסט ע"י מודל יוניגראם

Unigram LM p(w|θ) → **Sampling** → **Document =?**

Topic 1:
Text mining

text 0.2
mining 0.1
association 0.01
clustering 0.02
…
food 0.00001

מגבלים

Text mining article

נושאים

Topic 2:
Health

…
food 0.25
nutrition 0.1
healthy0.05
diet 0.02
…

Food nutrition article

הערכת יוניגראם מודל

Unigram LM p(w|θ)? ← **Estimation** — **Text Mining Paper d**

Total #words=**100**

בלתי מוכרע -
כל מילה לחלק
ההסתברות.

	…
10/100	text ?
5/100	mining ?
3/100	association ?
3/100	database ?
	…
1/100	query ?

text 10
mining 5
association 3
database 3
algorithm 2
…
query 1
efficient 1

Maximum Likelihood (ML) Estimator:

$$p(w|\theta) = p(w|d) = \frac{c(w,d)}{|d|}$$

מס' המילים ← c(w,d)
אורך ← |d| מספר

בהינתן מסמך d מה ההסתברות לw

האם זה הכי טוב?

כן, אבל ספציפית לנתונים אלו

מאוד לא מאוזן...

אין יוצאים מ' ה stopwords

LMs לייצוג נושא

מאוד לא ספציפי

המילון של אלפי מסמכים

General Background English Text	Computer Science Papers	Text mining article
B	**C**	**d**

the	**0.03**	**the**	**0.032**	**the**	**0.031**
a	0.02	a	0.019	…	
is	0.015	is	0.014	text	0.04
we	0.01	we	0.011	mining	0.035
…		…		association	0.03
food	0.003	**computer**	**0.004**	clustering	0.005
computer	**0.00001**	software	0.0001	**computer**	**0.0009**
text	0.000006	text	0.00006	…	
…		…		food	0.000001

נמוך

לא נמוך

מילים שלא כ"כ נפוצות אבל

Background LM: p(w|**B**) **Collection LM**: p(w|**C**) **Document LM**: p(w|**d**)

בהתבסס על נתונים שונים נקבל LM שונה,

LM תלוי **בהתפלגות** של המילים בדאטה-סט

LMs לניתוח הקשרים שונים

אלו מילים קשורות סמנטית למילה "computer"?

Topic LM: p(w|”**computer**”)

מסמכים המכילים את המילה "**computer**"

the	**0.032**
a	0.019
is	0.014
we	0.008
computer	**0.004**
software	0.0001

איזה LM יכול לתת לנו את המידע על מילים שכיחות ביותר שאנו נרצה "למחוק" אותן?

כדי לדרג את לשון ההשפה...

LMs לניתוח הקשרים שונים

אלו מילים קשורות סמנטית למילה "computer"?

נרמול:

מסמכים המכילים את המילה "computer"

Topic LM: p(w|"computer")

the	**0.032**
a	0.019
is	0.014
we	0.008
computer	**0.004**
software	0.0001

General Background English Text

B

Background LM: p(w|B)

the	**0.03**
a	0.02
is	0.015
we	0.01
computer	**0.00001**
text	0.000006
…	

Normalized Topic LM:

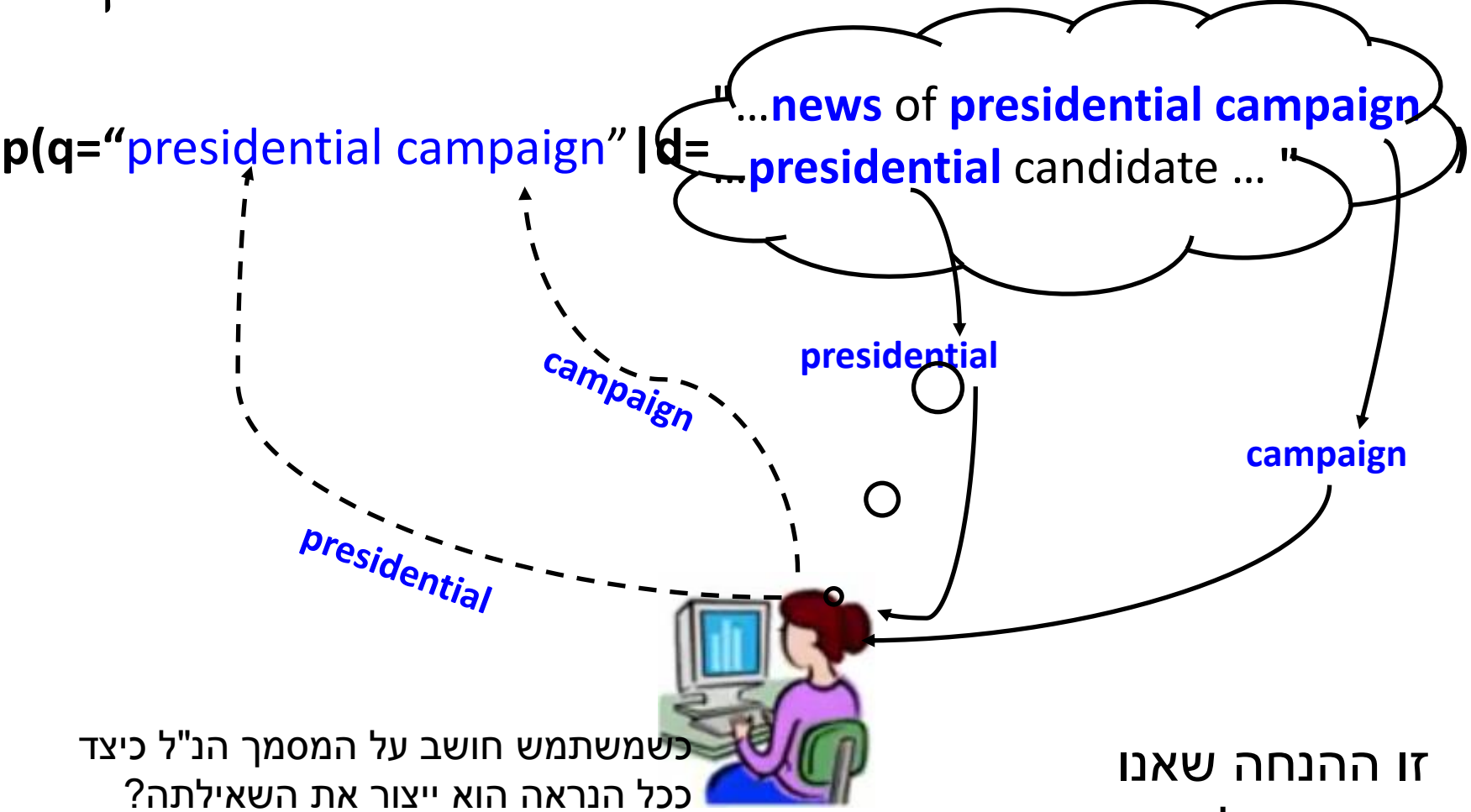
$$\frac{p(w|\text{"computer"})}{p(w|B)}$$

Computer	**400**
software	150
program	104
…	
text	3.0
…	
the	**1.1**
a	0.99
is	0.9
we	0.8

סיכום ביניים

- LM – מודל שפתי = התפלגות הסתברות על טקסט
- יוניגראם מודל = התפלגות המילים (בטקסט/מאגר נתון)
- שימושים ב- LM
  - מציאת נושא
  - מציאת קשרים בין מילים

יצירת שאילתות על ידי דגימת מילים ממסמך



כשמשתמש חושב על המסמך הנ"ל כיצד ככל הנראה הוא ייצור את השאילתה?

זו ההנחה שאנו מניחים על דרך יצירת השאילתה

Unigram Query Likelihood

p(q=“presidential campaign”**|d=**"…**news** of **presidential campaign** …**presidential** candidate … **"** **)**

$$p(\text{"}\boldsymbol{presidential}\text{"}|\mathbf{d}) * p(\text{"}\boldsymbol{campaign}\text{"}|\mathbf{d})$$

$$= \frac{c(\text{"}\boldsymbol{presidential}\text{"}, d)}{|d|} * \frac{c(\text{"}\boldsymbol{campaign}\text{"}, d)}{|d|}$$

הנחה:

אין תלות בין המילים

Query Likelihood -האם יש הגיון ב

$$p(q = "presidential\ campaign"|d) = \frac{c("presidential", d)}{|d|} * \frac{c("campaign", d)}{|d|}$$

P(q|d4= … news of **presidential campaign** … **presidential** candidate …) = $\frac{\mathbf{2}}{|\boldsymbol{d4}|} * \frac{\mathbf{1}}{|\boldsymbol{d4}|}$

P(q|d3= … news of **presidential campaign** …) = $\frac{\mathbf{1}}{|\boldsymbol{d3}|} * \frac{\mathbf{1}}{|\boldsymbol{d3}|}$

P(q|d3= … news about organic food **campaign**…) = $\frac{\mathbf{0}}{|\boldsymbol{d2}|} * \frac{\mathbf{1}}{|\boldsymbol{d2}|} = \mathbf{0}$

d4> d3 > d2 כמו שחשבנו

Query Likelihood -האם יש הגיון ב

q = "presidential campaign update"

דוגמא..

P(q|d4= … news of **presidential campaign** … **presidential** candidate …) = $\frac{2}{|d4|} * \frac{1}{|d4|} * \frac{0}{|d4|} = 0$

P(q|d3= … news of **presidential campaign** …) = $\frac{1}{|d3|} * \frac{1}{|d3|} * \frac{0}{|d3|} = 0$

P(q|d2= … news about organic food **campaign**…) = $\frac{0}{|d2|} * \frac{1}{|d2|} * \frac{0}{|d2|} = 0$

בגלל ההנחה שכל מילה בשאילתה חייבת להופיע במסמך

בגלל איזו הנחה עלתה הבעיה? כיצד נתקן זאת?

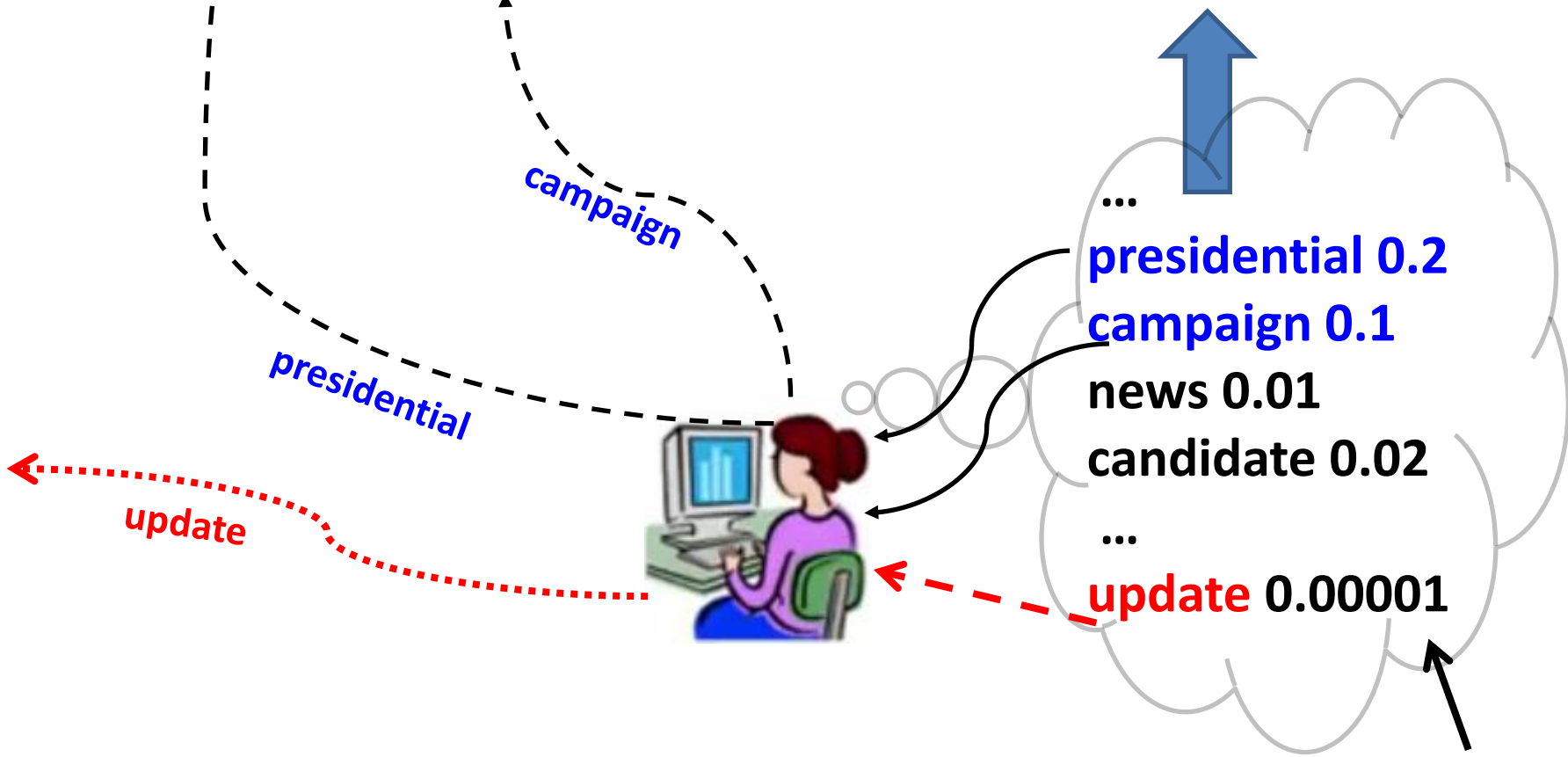
בשביל לתקן את זה אנחנו צריכים לקחת בחשבון שיהיו מילים בשאילתה שאינן מופיעות במסמך.

נכניס לתת רלוונטיות משקלים מסוימים

מודל משופר: המשתמש בוחר מילים מתוך **מודל** של המסמך

בלי קשר לאמת ל-מודל..

p(q="presidential campaign"|d= "...news of presidential campaign ...presidential candidate ... ")



מודל של מסמך

חישוב השאילתה

Query q = "data mining alg`"

מסמך

d1

Text mining article

$P(w|d_1)$

text 0.2
mining 0.1
association 0.01
clustering 0.02
…
food 0.00001

$P(\text{"data mining alg`"}|d_1)=$
$p(\text{"data"}|d_1)*$
$p(\text{"mining"}|d_1)*$
$p(\text{"alg`"}|d_1)$

d2

Food nutrition article

$P(w|d_2)$

…
food 0.25
nutrition 0.1
healthy0.05
diet 0.02
…

$P(\text{"data mining alg`"}|d_2)=$
$p(\text{"data"}|d_2)*$
$p(\text{"mining"}|d_2)*$
$p(\text{"alg`"}|d_2)$

סיכום ביניים: דירוג מבוסס QL

$q=w_1w_2\ldots w_n$ $\qquad$ $p(q \mid d) = p(w_1 \mid d) * \ldots * p(w_n \mid d)$

$\log(a \cdot b \cdot c \ldots)$ $\qquad$ $\log(a)+\log(b)+\log(c)\ldots$

$$f(q,d) = \log\big(p(q|d)\big) = \sum_{i=1}^{n} \log\big(p(w_i|d)\big) = \sum_{w\in V} c(w,q)\log\big(p(w|d)\big)$$

מדוע לעבור לחישוב עם log?

הסתברויות נמוכות מובילות למכפלה מאוד נמוכה

ולחריגה (under flow) בתוצאות

Document language model

המעבר ממכפלה לחיבור של log פשוטה

הסיגמא הראשונה מבוצעת על כל המילים בשאילתה

הסיגמא השנייה מבוצעת על כל המילים במילון אבל עם מכפלה של נוכחות המילה בשאילתה

Retrieval problem → Estimation of $p(w_i|d)$

Different estimation methods → different ranking functions

פונקציית דירוג מבוסס QL

$q=w_1w_2...w_n$ $\qquad$ $p(q \mid d) * p(w_1 \mid d) *....*p(w_n \mid d)$

$$f(q,d) = \log\big(p(q|d)\big) = \sum_{i=1}^{n} \log\big(p(w_i|d)\big) = \sum_{w \in V} c(w,q) \log\big(p(w|d)\big)$$

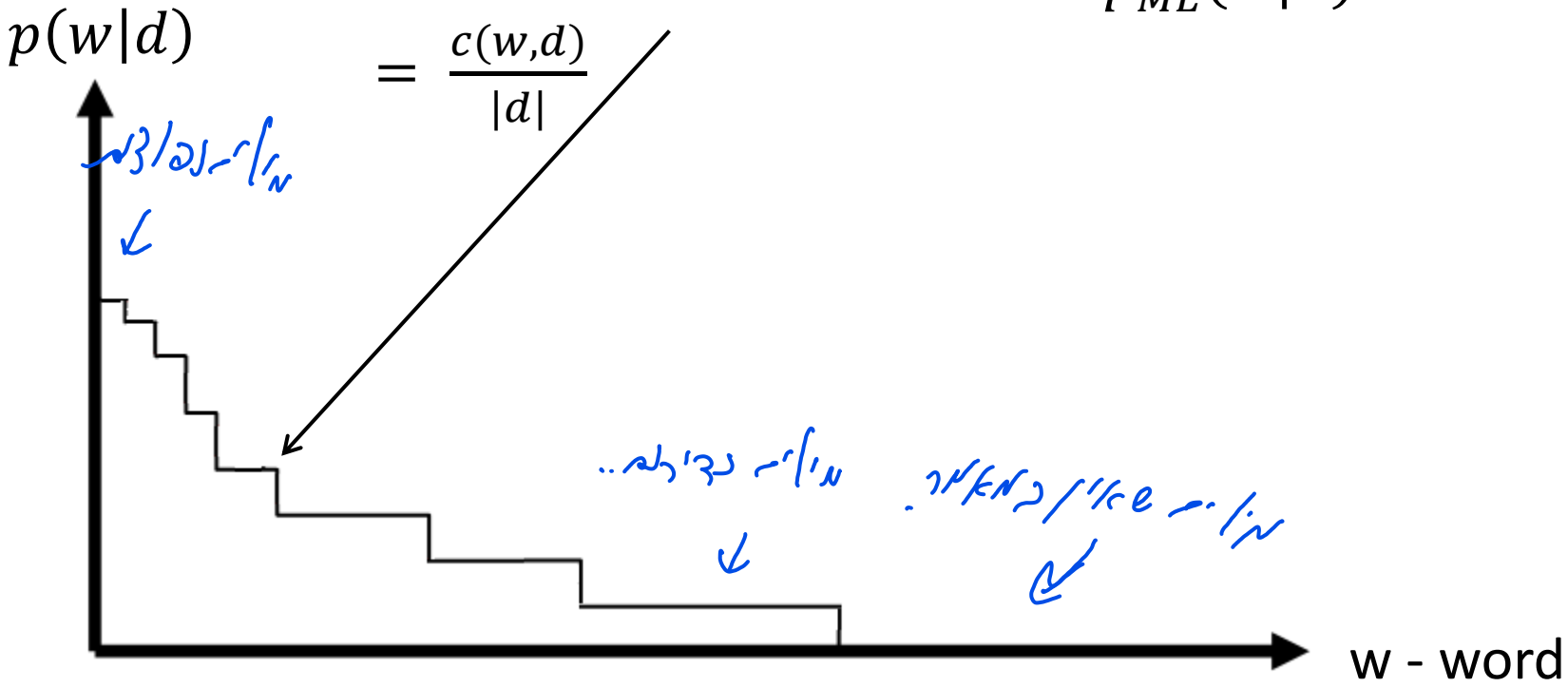
כיצד אנו נעריך את $p(w|d)$?

יש מילים שלא מופיעות באקסטרקט
ובן באמאמר..

חישוב $p(w|d)$

Max. Likelihood Estimate $p_{ML}(w|d)$

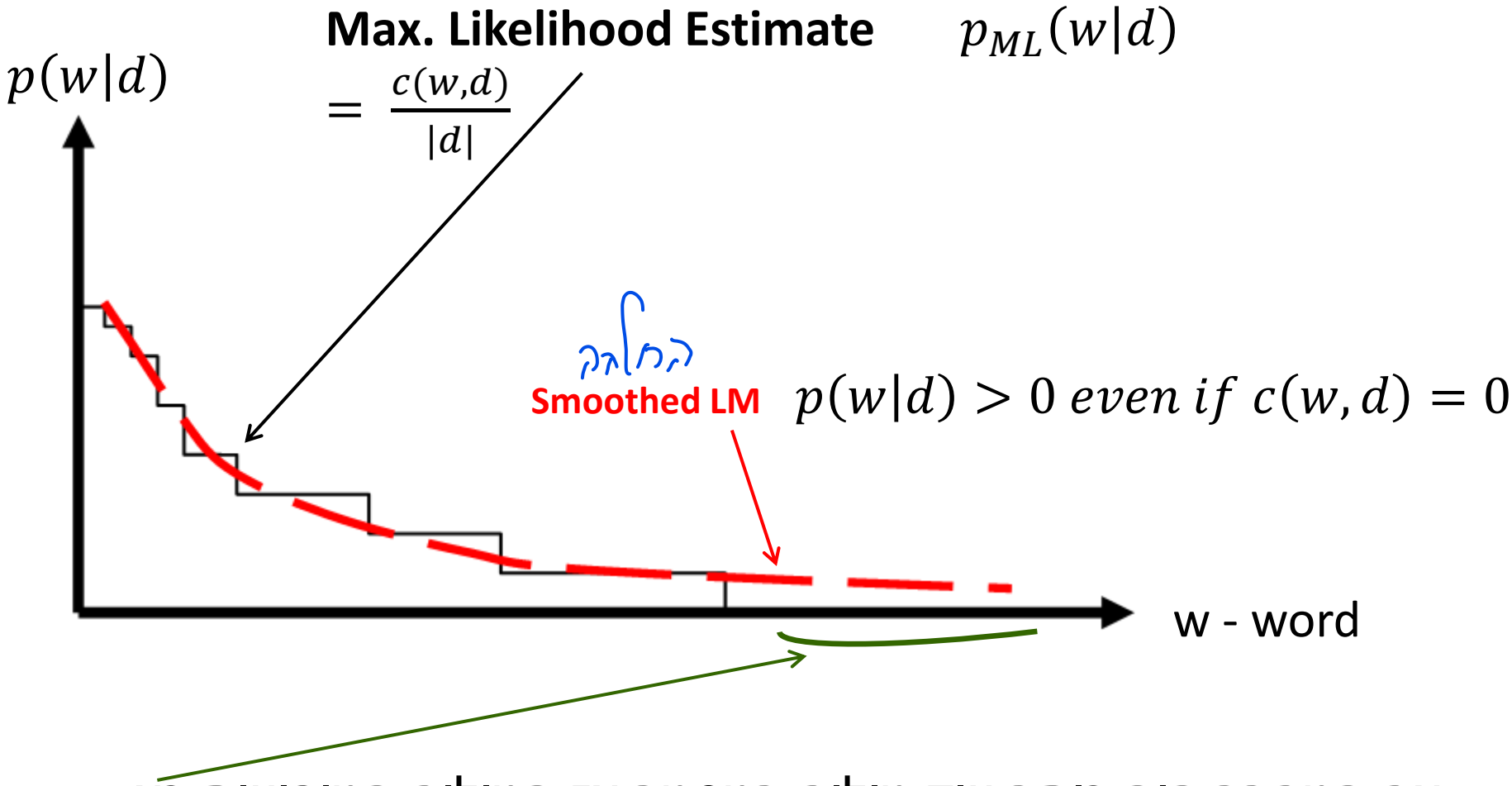
$$= \frac{c(w,d)}{|d|}$$



קו ישר: מילים בעלי שכיחות זהה

(נניח כי - מבוטלת לגמרי..) נרדב לאת ערכים למילים שלא מופיעות.

חישוב $p(w|d)$



אם המחבר היה כותב עוד מילים במסמך אז המילים המופיעות כאן היו בהסתברות חיובית, לכן לעשות smoothing זה הדבר הנכון

כיצד נעשה smoothing

- השאלה המרכזית: איזו הסתברות ניתן למילה שאינה במסמך?

  הכיון - ההסתברות של המילה בשפה..
- ההסתברות של מילה שאינה במסמך תהיה פרופורציונאלית להסתברות שלה הניתנת על ידי רפרנס LM (=מודל השפה)
  - אפשרות אחת: רפרנס LM = אוסף LM

Discounted ML estimate

$$p(w|d) = \begin{cases} p_{seen}(w|d) & \textit{if w is seen in d} \\ \alpha_d p(w|C) & \textit{otherwise} \end{cases}$$

מודל המסמך

לא מודל.

כמה נתן פחות לקולקציה.

α – מקדם של הקולקציה

Collection language model

נתון כאן שזו הבחנה..

כתיבה מחודשת של ה- smoothing

V – מילון השפה.

$$f(q,d) = \log\big(p(q|d)\big) = \sum_{i=1}^{n} \log\big(p(w_i|d)\big) = \sum_{w\in V} c(w,q)\log(p(w|d)) =$$

סה"כ המילים בשאילתא.

$$\sum_{w\in V, c(w,d)>0} c(w,q)\,log\,p_{seen}(w|d) + \sum_{w\in V, c(w,d)=0} c(w,q)\,log\,\alpha_d\, p(w|d)$$

מילים בשאילתה המופיעות במסמך

מילים בשאילתה שאינן מופיעות במסמך

$$\sum_{w\in V} c(w,q)\,log\,\alpha_d\, p(w|C) - \sum_{w\in V, c(w,d)>0} c(w,q)\,log\,\alpha_d\, p(w|d)$$

כל המילים בשאילתה

המילים בשאילתה **שנמצאות** במסמך

כתיבה מחודשת של ה- smoothing

$$\sum_{w\in V, c(w,d)>0} c(w,q)\log p_{seen}(w|d) + \sum_{w\in V, c(w,d)=0} c(w,q)\log \alpha_d\, p(w|C)$$

מילים בשאילתה המופיעות במסמך

מילים בשאילתה שאינן מופיעות במסמך

$$\sum_{w\in V} c(w,q)\log \alpha_d\, p(w|C) - \sum_{w\in V, c(w,d)>0} c(w,q)\log \alpha_d\, p(w|C)$$

כל המילים בשאילתה

המילים בשאילתה **שנמצאות** במסמך

$$\boxed{\begin{aligned}&\sum_{w\in V, c(w,d)>0} c(w,q)\log p_{seen}\, p(w|d) - \sum_{w\in V, c(w,d)>0} c(w,q)\log \alpha_d\, p(w|C)\\ &= \sum_{w\in V, c(w,d)>0} c(w,q)\log \frac{p_{seen}(w|d)}{\cancel{\log}\,\alpha_d\, p(w|C))}\end{aligned}}$$

כתיבה מחודשת של ה- smoothing

$$\sum_{w\in V, c(w,d)>0} c(w,q)\log p_{seen}(w|d) - \sum_{w\in V, c(w,d)>0} c(w,q)\log \alpha_d\, p(w|C)$$

$$= \sum_{w\in V, c(w,d)>0} c(w,q)\log \frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)}$$

$$\Downarrow$$

קבוע.

$$\sum_{w\in V} c(w,q)\log \alpha_d\, p(w|C) = |q|\log \alpha_d + \sum_{w\in V} c(w,q)\log p(w|C)$$

כתיבה מחודשת של ה- smoothing

$$\sum_{w\in V, c(w,d)>0} c(w,q)\log p_{seen}(w|d) + \sum_{w\in V, c(w,d)=0} c(w,q)\log \alpha_d\, p(w|C)$$

$$= \sum_{w\in V} c(w,q)\log \alpha_d\, p(w|C) + \sum_{w\in V, c(w,d)>0} c(w,q)\log p_{seen}(w|d)$$

$$- \sum_{w\in V, c(w,d)>0} c(w,q)\log \alpha_d\, p(w|C) =$$

$$\sum_{w\in V, c(w,d)>0} c(w,q)\log p_{seen}(w|d) + \sum_{w\in V, c(w,d)>0} c(w,q)\log \frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)}$$

סה"כ

$$\boxed{= \sum_{w\in V, c(w,d)>0} c(w,q)\log \frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)} + |q|\log \alpha_d + \sum_{w\in V} c(w,q)\log p(w|C)}$$

(כשרוצים לדעת ע"י הסדר לא משנה אם נרחיק נשים לב בכלל..)

כתיבה מחודשת של ה- smoothing

הנוסחה הראשונה

$$\sum_{w\in V, c(w,d)>0} c(w,q)\log(p_{seen}(w|d)) + \sum_{w\in V, c(w,d)=0} c(w,q)\log(\alpha_d(p(w|C)))$$

$$= \sum_{w\in V, c(w,d)>0} c(w,q)\log\frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)} + |q|\log\alpha_d + \sum_{w\in V} c(w,q)\log(p(w|C))$$

הנוסחה הסופית

תזכורת: נוסחת ה- IDF-TF

מספר מופעים של מילה/מונח/מאפיין ספציפי בשאילתה

Total # of docs in collection

$$f(q,d) = \sum_{i=1}^{N} x_i y_i = \sum_{w \in q \cap d} c(w,q)c(w,d)\log\frac{M+1}{df(w)}$$

All matched query words in d

Doc Frequency

$$\mathrm{IDF(w)} = log\frac{M+1}{df(w)}$$

נוסחה מעט שונה עם אותם מרכיבים

$$\sum_{w\in V, c(w,d)>0} c(w,q)\log p_{seen}(w|d) + \sum_{w\in V, c(w,d)=0} c(w,q)\log \alpha_d\, p(w|C)$$

$$= \sum_{w\in V, c(w,d)>0} c(w,q)\log \frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)} + |q|\log\alpha_d + \sum_{w\in V} c(w,q)\log p(w|C)$$

$$\sum_{w_i\in q\cap d} [\log \frac{p_{seen}(w_i|d)}{\alpha_d\, p(w_i|C)}] + n * \log\alpha_d + \sum_{i=1}^{n} \log p(w_i|C)$$

היתרונות של הכתיבה המחודשת

Smoothing עם p(w|C) מכיל תכונות של TF-IDF ממושקל עם נרמול בגודל מסמך

מספר המסמכים הכולל באוסף

$$\mathrm{IDF(w)} = log\frac{M+1}{df(w)}$$

בכמה מסמכים נמצאת המילה

$$TF(w) = c(w,d)$$

מספר מופעי המילה בשאילתה

מספר מופעי המילה במסמך

$$\boldsymbol{FT-IDF} = \sum_{w_i \in q \cap d} \mathbf{c(w,q)c(w,d)}\, log\frac{M+1}{df(w)}$$

כגודל המופעים במסמך כך גודל המשקל

$$\sum_{w_i \in q \cap d} \mathbf{c(w,q)}[\boldsymbol{log}\frac{\boldsymbol{p_{seen}(w_i|d)}}{\boldsymbol{\alpha_d\, p(w_i|C)}}] + \boldsymbol{n * log\alpha_d} + \sum_{i=1}^{n} logp(w_i|C)$$

פופולריות של מונח באוסף

נרמול באורך המשפט, בהמשך יהיה ברור יותר

היתרונות של הכתיבה המחודשת

Smoothing עם p(w|C) מכיל תכונות של TF-IDF ממושקל עם נרמול בגודל מסמך

מספר המסמכים הכולל באוסף

$$\mathrm{IDF(w)} = log\frac{M+1}{df(w)}$$

בכמה מסמכים נמצאת המילה

$$TF(w) = c(w,d)$$

מספר מופעי המילה בשאילתה

מספר מופעי המילה במסמך

$$\boldsymbol{FT-IDF} = \sum_{w_i \in q \cap d} \mathbf{c(w,q)c(w,d)}\, log\frac{M+1}{df(w)}$$

מספר קבוע עבור כל המסמכים,
אינו משפיע על הדירוג
לכן טיפה לא אמיתי

$$\boxed{\sum_{w_i \in q \cap d} c(w,q)[log\frac{p_{seen}(w_i|d)}{\alpha_d\, p(w_i|C)}] + n * log\alpha_d} + \cancel{\sum_{i=1}^{n} \boldsymbol{logp(w_i|C)}}$$

כבר לא
רלוונטי

הנחה שמילה
שמופיעה בהרבה
מסמכים
אין הנחה מילה
שהסתברות גדולה...

סיכום ביניים על Smoothing

- Smoothing על p(w|d) הכרחי בשביל QL
- בכלליות: Smoothing עם p(w|C)
  - הנחה: מילה שאינה מופיעה ב- d ספציפי היא פרופורציונאלית ל- p (w|C)
  - מוביל אותנו לנוסחת דירוג כללית בעלת תכונות של TF-IDF ממושקל ועם נרמול בגודל המסמך
  - הציון/דירוג מבוסס בעיקר על סכום משקולות המונחים המופיעים בשאילתה
- עדיין נותרה השאלה כיצד בדיוק יתבצע ה- Smoothing

Query Likelihood + Smoothing with p(w|C)

$$logp(q|d) = \sum_{w_i \in q \cap d} c(w,q)[log\frac{p_{seen}(w_i|d)}{\alpha_d\, p(w_i|C)}] + n * log\alpha_d + \boxed{\sum_{\boldsymbol{w} \in \boldsymbol{V}} \boldsymbol{c(w,q)logp(w|C)}}$$

$$\boxed{f(q,d) = \sum_{w_i \in q \cap d} c(w,q)[log\frac{p_{seen}(w_i|d)}{\alpha_d\, p(w_i|C)}] + n * log\alpha_d}$$

איך נחשב בדיוק סיכוי של מילה?

$P_{seen}(w_i|d)$?
α_d?

איך נקבע את אלפא?

How to smooth p(w|d)?

(Jelinek-Mercer) Smoothing אינטרפולציה לינארית

Unigram LM p(w|θ)?

	…
10/100	text ?
5/100	mining ?
3/100	association ?
3/100	database ?
	…
1/100	query ?
0/100	**network ?**

Docoment d
Total #words=**100**

text 10
mining 5
association 3
database 3
algorithm 2
…
query 1
efficient 1

Collection LM
P(w|C)

the 0.1
a 0.08
..
computer 0.02
database 0.01
……
text 0.001
network 0.001
mining 0.0009
…

$$p(w|d) = \underbrace{(1-\lambda)}\frac{c(w,d)}{|d|} + \underbrace{\lambda}p(w,C) \quad \lambda \in [0,1]$$

מה יקרה לפי מלים

אינטרפולציה לינארית (Jelinek-Mercer) Smoothing

כך מקבלים הסתברות שאיננה שלילית גם למונחים שאינם במסמך

דוגמה

אינטרפולציה לינארית

$(1-\lambda)+\lambda=1$

$$p(w|d) = (1-\lambda)\frac{c(w,d)}{|d|} + \lambda p(w,C) \quad \lambda \in [0,1]$$

Max` likelihood

Smoothing parameter

Collection LM

Collection LM
P(w|C)

the 0.1
a 0.08
..
computer 0.02
database 0.01
.....
text 0.001
network 0.001
mining 0.0009
...

אינטרפולציה לינארית (Jelinek-Mercer) Smoothing

Unigram LM p(w|θ)?

…
10/100 → **text ?**
5/100 → mining ?
3/100 → association ?
3/100 → database ?
…
1/100 → query ?
0/100 → **network ?**

Docoment d
Total #words=**100**

text 10
mining 5
association 3
database 3
algorithm 2
…
query 1
efficient 1

Collection LM
P(w|C)

the 0.1
a 0.08
..
computer 0.02
database 0.01
……
text 0.001
network 0.001
mining 0.0009
…

$$p(w|d) = (1-\lambda)\frac{c(w,d)}{|d|} + \lambda p(w,C) \quad \lambda \in [0,1]$$

$$p(\text{"}text\text{"}|d) = (1-\lambda)\frac{\mathbf{10}}{\mathbf{100}} + \lambda * \mathbf{0.001}$$

אינטרפולציה לינארית (Jelinek-Mercer) Smoothing

Unigram LM p(w|θ)?

10/100 → text ?
5/100 → mining ?
3/100 → association ?
3/100 → database ?
…
1/100 → query ?
0/100 → **network ?**

Docoment d
Total #words=**100**

text 10
mining 5
association 3
database 3
algorithm 2
…
query 1
efficient 1

Collection LM
P(w|C)

the 0.1
a 0.08
..
computer 0.02
database 0.01
……
text 0.001
network 0.001
mining 0.0009
…

$$p(w|d) = (1-\lambda)\frac{\boldsymbol{c(w,d)}}{|d|} + \lambda p(w,C) \quad \lambda \in [0,1]$$

zero

$$p(\text{"network"}|d) = \lambda * \mathbf{0.001}$$

עלמילה אחת

לא מכפיל קבוע אחת

$$p(w|d) = (1-\lambda)\frac{c(w,d)}{|d|} + \lambda p(w,C) \quad \lambda \in [0,1]$$

Max` likelihood

Constant Smoothing parameter

Collection LM

$$p(w|d) = \frac{c(w,d)+\mu p(w|C)}{|d|+\mu} = \frac{|d|}{|d|+\mu}\frac{c(w,d)}{|d|} + \frac{\mu}{|d|+\mu}p(w|C) \quad \mu \in [0, \infty)$$

מספר מילים במסמך.

$$\boxed{\frac{|d|}{|d|+\mu} + \frac{\mu}{|d|+\mu} = 1}$$

dynamic Smoothing parameter

ככל שהמסמך גדול יותר
כך המקדם קטן יותר

ומילה שלא הופיעה ולכן נכבה בשביטה משקל גמוך.

Dirichlet Prior (Bayesian) Smoothing

Unigram LM p(w|θ)?

10/100 → **text ?**
5/100 → mining ?
3/100 → association ?
3/100 → database ?
1/100 → query ?
0/100 → **network ?**

Docoment d
Total #words=**100**

text 10
mining 5
association 3
database 3
algorithm 2
…
query 1
efficient 1

Collection LM
P(w|C)

the 0.1
a 0.08
..
computer 0.02
database 0.01
……
text 0.001
network 0.001
mining 0.0009
…

$$p(w|d) = \frac{c(w,d) + \mu p(w|C)}{|d| + \mu} = \frac{|d|}{|d| + \mu}\frac{c(w,d)}{|d|} + \frac{\mu}{|d| + \mu}p(w|C)$$

$\mu \in [0, \infty)$

$$p(\text{"}text\text{"}|d) = \frac{\mathbf{10} + \mu * \mathbf{0.001}}{100 + \mu}$$

Dirichlet Prior (Bayesian) Smoothing

Unigram LM p(w|θ)?

10/100 → **text ?**
5/100 → mining ?
3/100 → association ?
3/100 → database ?
1/100 → query ?
0/100 → **network ?**

Docoment d
Total #words=**100**

text 10
mining 5
association 3
database 3
algorithm 2
…
query 1
efficient 1

Collection LM
P(w|C)

the 0.1
a 0.08
..
computer 0.02
database 0.01
……
text 0.001
network 0.001
mining 0.0009
…

$$p(w|d) = \frac{c(w,d) + \mu p(w|C)}{|d| + \mu} = \frac{|d|}{|d| + \mu}\frac{c(w,d)}{|d|} + \frac{\mu}{|d| + \mu}p(w|C)$$

$\mu \in [0, \infty)$

zero

$$p(\text{"network"}|d) = \frac{\mu * \mathbf{0.001}}{100 + \mu}$$

פונקציית דירוג עבור (Dirichlet Prior) Smoothing

$$f(q,d) = \sum_{w_i \in q \cap d} c(w,q)\log[\frac{\boldsymbol{p_{seen}(w_i|d)}}{\boldsymbol{\alpha_d\, p(w_i|C)}}] + n * log\alpha_d$$

Maximum Likelihood (ML) :

$$p(w|d) = \frac{c(w,d)}{|d|}$$

$$p(w|d) = (1-\lambda)\frac{c(w,d)}{|d|} + \lambda p(w,C) \quad \lambda \in [0,1]$$

$$\frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)} = \frac{(1-\lambda)p_{ML}(w|d) + \lambda p(w|C)}{\lambda p(w|C)} = \boxed{1 + \frac{1+\lambda}{\lambda}\frac{c(w,d)}{|d|p(w|C)}}$$

$$f_{JM}(q,d) = \sum_{w_i \in q \cap d} c(w,q)\log[1 + \frac{1+\lambda}{\lambda}\frac{c(w,d)}{|d|p(w|C)}] + n * log\alpha_d$$

פונקציית דירוג עבור Smoothing

$$f(q,d) = \sum_{w_i \in q \cap d} c(w,q)\log[\frac{\boldsymbol{p_{seen}(w_i|d)}}{\boldsymbol{\alpha_d\, p(w_i|C)}}] + n * log\alpha_d$$

$$\frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)} = \frac{(1-\lambda)p_{ML}(w|d) + \lambda p(w|C)}{\boldsymbol{\lambda} p(w|C)} = 1 + \frac{1+\lambda}{\lambda}\frac{c(w,d)}{|d|p(w|C)}$$

$$f_{JM}(q,d) = \sum_{w_i \in q \cap d} c(w,q)\log[1 + \frac{1+\lambda}{\lambda}\frac{c(w,d)}{|d|p(w|C)}] + n * log\alpha_d$$

$\lambda = \alpha_d \rightarrow$ $n * log\alpha_d$ is constant ← קבוע ולכן ניתן למחוק אותו.

$$f_{JM}(q,d) = \boxed{\sum_{w_i \in q \cap d} c(w,q)\log[1 + \frac{1+\lambda}{\lambda}\frac{c(w,d)}{|d|p(w|C)}]} \cancel{+ n * log\alpha_d}$$

פונקציית דירוג עבור Smoothing

$$f_{JM}(q,d) = \sum_{w_i \in q \cap d} c(w,q)\log[1 + \frac{1+\lambda}{\lambda}\frac{c(w,d)}{|d|p(w|C)}]$$

TF = c(w,d) $IDF = p(w|C)$ |d|נרמול באורך המשפט

הנוסחה הזו "תופשת" את כל הרעיונות הנ"ל

בעוד ב – VSM השתמשנו בהיוריסטיקות בשביל שלושת הרעיונות במודל ההסתברותי זה נובע ישירות (לאחר הנחות בסיסיות

אין קשר בין המילים (לא נכון)
קיבלת הערכה מצוינת שהוולים המילה.

$|d|p(w|C)$ - משמעות של הסיכוי לקבל את המילים במסמך מתוך כל הקולקציה של המסמכים

$\frac{c(w,d)}{|d|p(w|C)}$ — מספר מופעי המילה בפועל מנורמל במספר המופעים של המילה שהיינו מצפים לקבל מכל הקולקציה

פונקציית דירוג עבור Smoothing (Dirichlet Prior)

$$f(q,d) = \sum_{w_i \in q \cap d} c(w,q)\log\left[\frac{\boldsymbol{p_{seen}(w_i|d)}}{\boldsymbol{\alpha_d\, p(w_i|C)}}\right] + n * log\alpha_d$$

$$p(w|d) = \frac{c(w,d)+\mu p(w|C)}{|d|+\mu} = \frac{|d|}{|d|+\mu}\frac{c(w,d)}{|d|} + \frac{\mu}{|d|+\mu}p(w|C) \quad \mu \in [0,\infty)$$

לא תלוי בd

$$\frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)} = \frac{\dfrac{c(w,d)+\mu p(w|C)}{|d|+\mu}}{\dfrac{\mu p(w|C)}{|d|+\mu}} = 1 + \frac{c(w,d)}{\mu p(w|C)} \qquad \alpha_d = \frac{\mu}{|d|+\mu}$$

לא תלוי בq ובמסמך

$$f_{DIR}(q,d) = \left[\sum_{w_i \in q \cap d} c(w,q)\log\left[1 + \frac{c(w,d)}{\mu p(w|C)}\right]\right] + n * log\frac{\mu}{|d|+\mu}$$

$$p(w|d) = \frac{c(w,d)+\mu p(w|C)}{|d|+\mu} = \frac{|d|}{|d|+\mu}\frac{c(w,d)}{|d|} + \frac{\mu}{|d|+\mu}p(w|C) \quad \mu \in [0,\infty)$$

$$\frac{p_{seen}(w|d)}{\alpha_d\, p(w|C)} = \frac{\frac{c(w,d)+\mu p(w|C)}{|d|+\mu}}{\frac{\mu p(w|C)}{|d|+\mu}} = 1 + \frac{c(w,d)}{\mu p(w|C)} \qquad \alpha_d = \frac{\mu}{|d|+\mu}$$

אנחנו משווים את מספר המופעים האמתי במשפט עם דגימה של μ מילים מהמאגר

$$\alpha_d = \frac{\mu}{|d|+\mu}$$

היכן הנרמול באורך המשפט?

$$f_{DIR}(q,d) = [\sum_{w_i \in q \cap d} c(w,q)\log[1 + \frac{c(w,d)}{\mu p(w|C)}]] + n * log\frac{\mu}{|d|+\mu}$$

סיכום הם smoothing

- למדנו שתי שיטות של smoothing
  - Jelinek-Mercer: אינטרפולציה לינארית, פרמטר קבוע
  - Dirichlet Prior: אינטרפולציה אדפטיבית
- שתיהן מובילות לפונקציות IR עם הנחות יסוד מוגדרות(פחות היוריסטיות)
  - מכילות תבונות של FT-IDF עם נרמול באורך המסמך
  - בעלות פרמטר smoothing אחד בלבד

סיכום

- פונקציות דירוג יעילות המתקבלות באמצעות מודל הסתברותי "נקי" תחת ההנחות הבאות:
  1. Relevance(q,d) = p(R=1|q,d) ≈ p(q|d,R=1) **≈p(q|d)**
  2. מילות השאילתה אינן תלויות בניהן (BoW) ← תכונה שלילית
  3. Smoothing with p(w|C) ← הוא אופן מיוחד
  4. שתי שיטות של Smoothing
- פחות היוריסטיות יחסית ל- VSM
- ישנן לא מעט הרחבות לפונקציות הדירוג