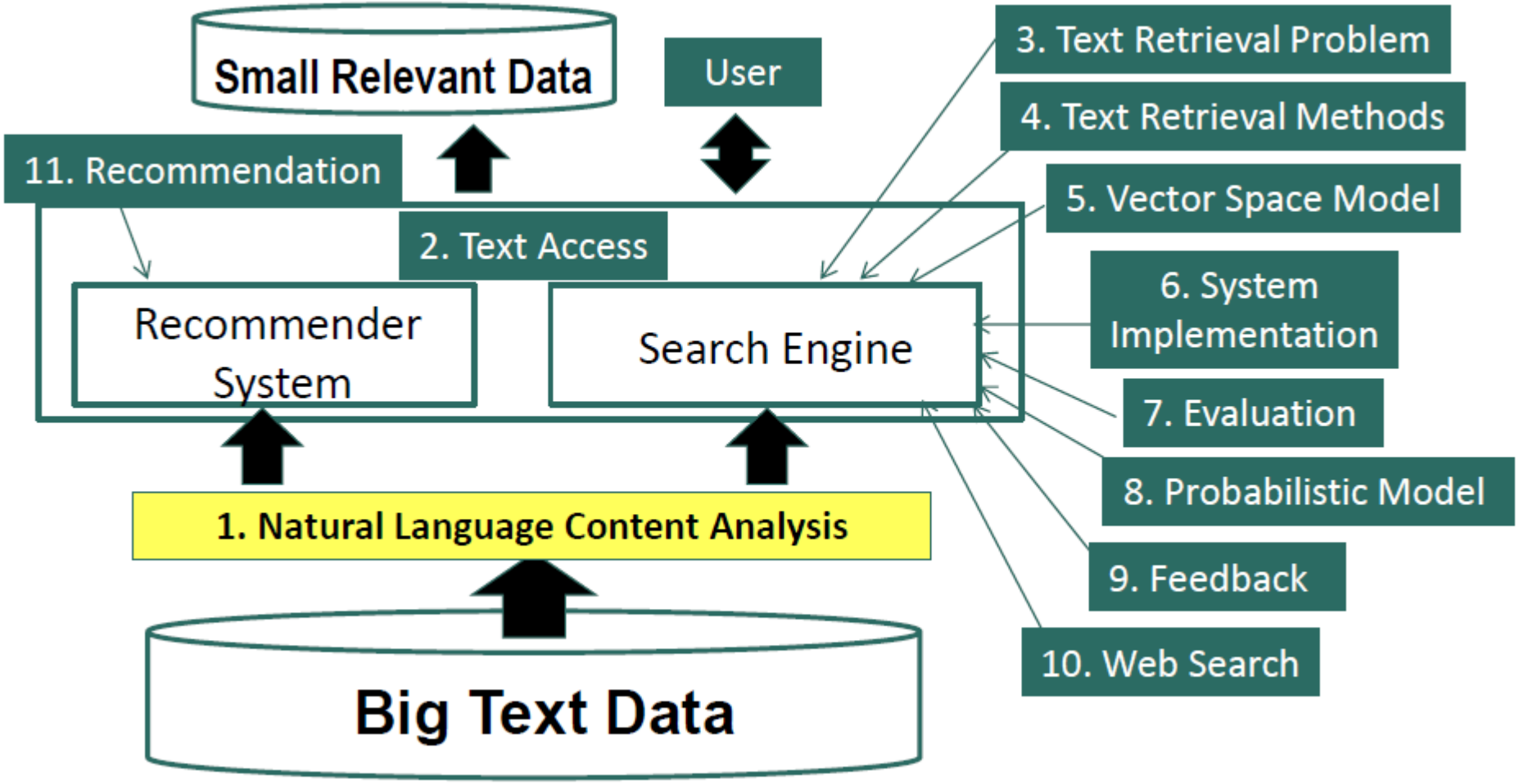
Information Retrieval

Course number:

lecturer: Dror Mughaz

מודל כללי של מנוע חיפוש

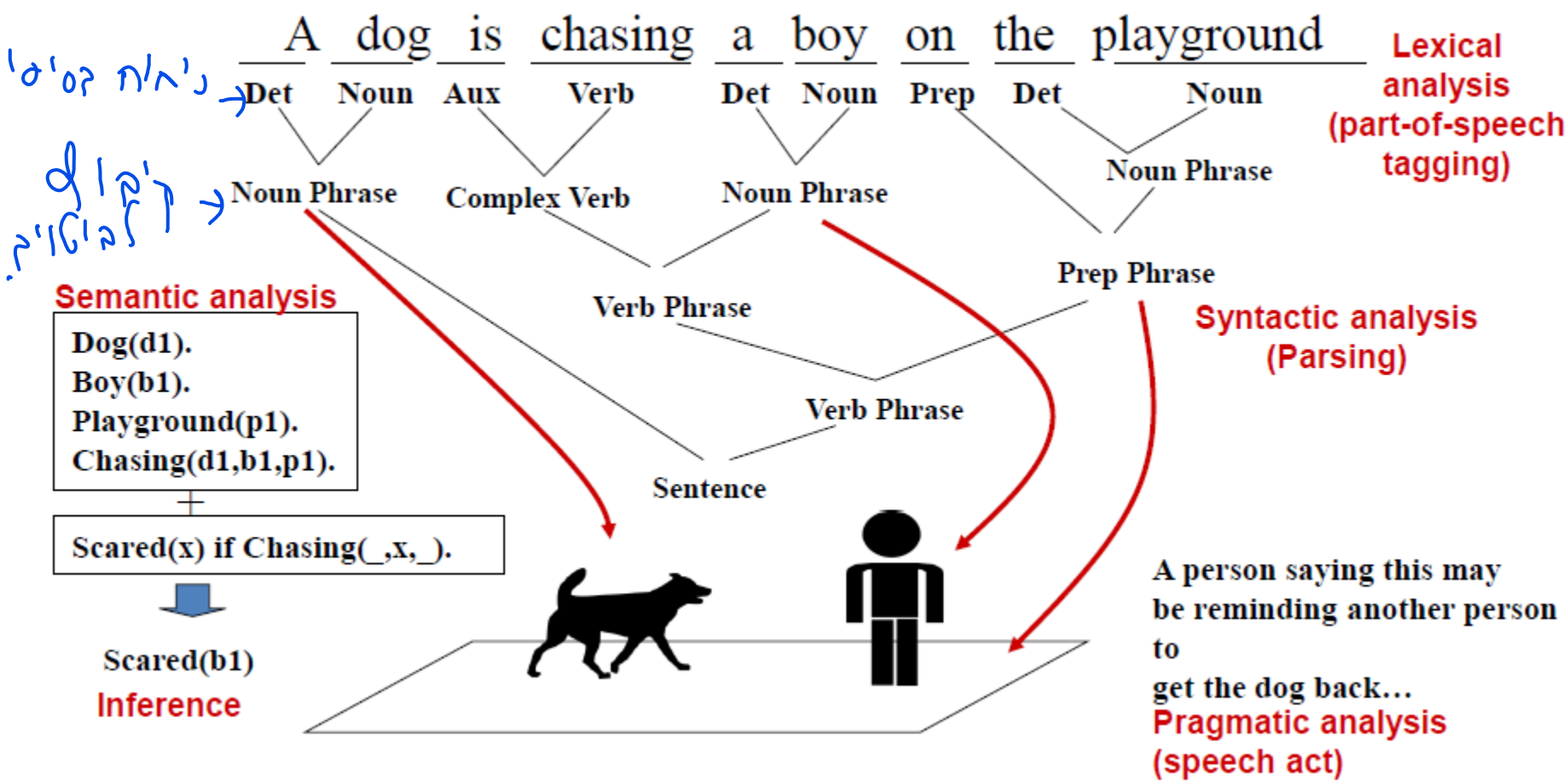


עיבוד שפה טבעית

- מה זה עיבוד שפה טבעית?
  - Natural Language Processing (NLP) –
- גישה חדשנית בעיבוד שפה טבעית (NLP)
- שימוש בעיבוד שפה טבעית (NLP) לצורך אחזור מידע

דוגמא ל- NLP

ניתוח משפט.



כיום יש כלים שיודעים לספר על האמונה ללא יכולת לאמת

קשיים ב- NLP

שפה טבעית נועדה ליצור תקשורת יעילה בין בני אדם.

התוצאה:

- אנו משמיטים הרבה ידע על "השכל הישר", שאנו מניחים שיש לשומע/לקורא
- משפטים אנושיים מכילים הרבה עמימות שאנו מניחים שהשומע/קורא יודע לפתור
  - לפעמים אנו לא מודעים לעמימות שקיימת במשפט

קשיים ב- NLP (המשך)

זה הופך את כל הצעדים ב- NLP לקשים

- רב משמעותיות "הורסת כל חלקה טובה"
- יש צורך בידע מקדים רב ביותר

דוגמאות לאתגרים ב- NLP

רב משמעותיות ברמת המילה:

- "design" יכול להיות פועל או שם-עצם (עמימות ב- POS)
- "root" רב משמעי (עמימות ברמת המשמעות)
  - שורש של צמח
  - שורש ריבועי
  - שורש של עץ קבצים

רב משמעותיות ברמה התחבירית:

- Natural Language Processing (NLP)
  - עיבוד שפה טבעית
  - עיבוד שפה הוא טבעי
- "A man saw a boy <u>with a telescope</u>."
  - גבר ראה ילד בעזרת טלסקופ
  - גבר ראה ילד מחזיק טלסקופ

האם למשהו יש רעיון למשמעויות שונות של המילה "ציר"?

- אוכל
- דלת
- דרך
- לידה
- קונסול – שגרירות

דוגמאות לאתגרים ב- NLP

אנפורה: (הגיבוסלה אתלוה)

- "John persuaded Bill to buy a TV for himself."
  - John - Himself?
  - Bill - Himself?

הנחות מראש של דברים:

- "He has quit smoking."
  - יש רמז על כך שבעבר הוא עישן

דוגמאות לאתגרים ב- NLP

בעקבות הדוגמאות שהראינו לעיל אם נרצה שמחשב יבין שפה טבעית יהיה עליו:

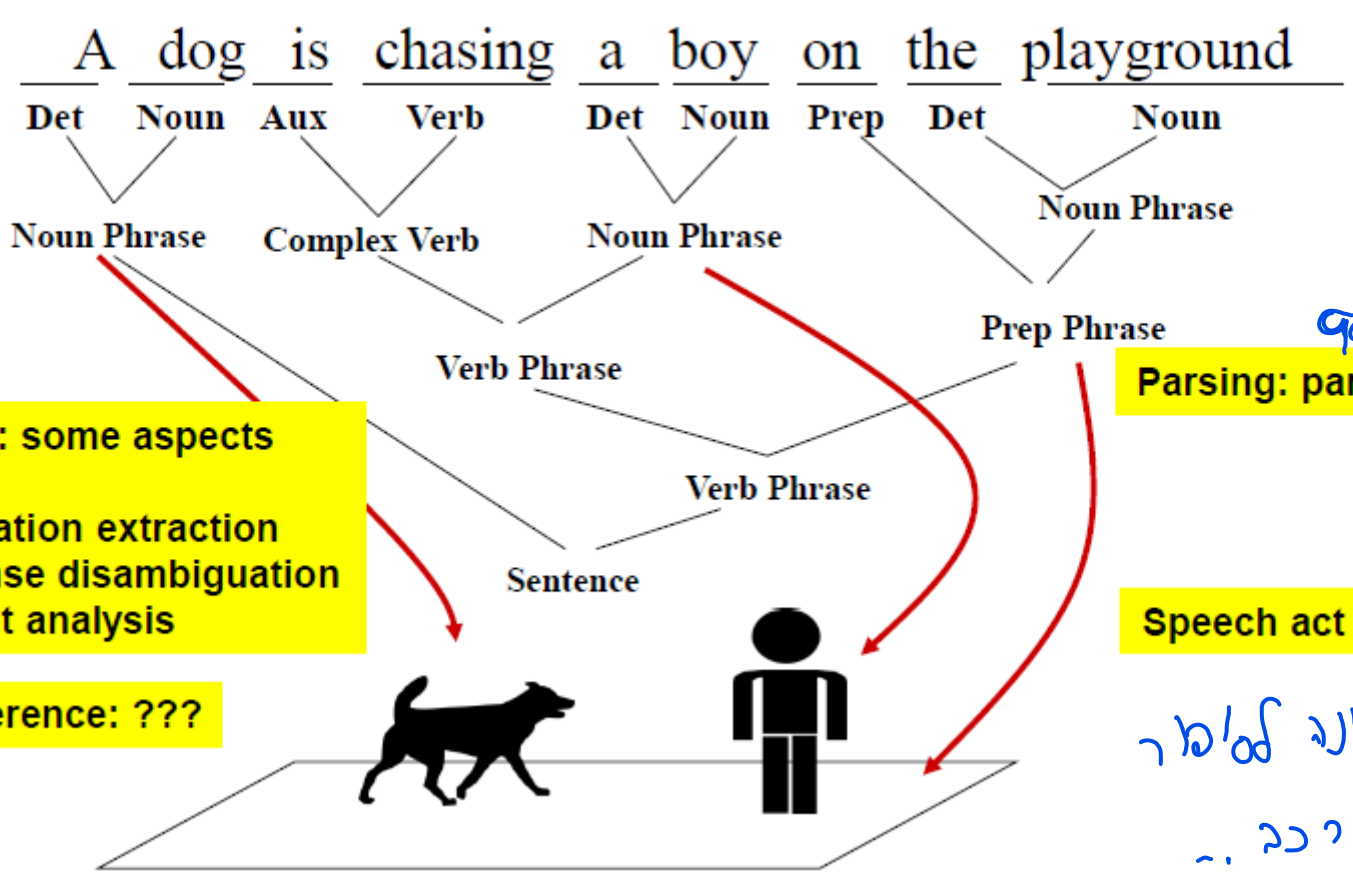
- להשתמש במידע רב ביותר
- לשמור ידע רב על משמעות של מילים
- לקשר בין מילים למידע אנושי בסיסי

- מסקנה: הדבר מאוד קשה לביצוע ע"י מחשב.

← המצב היום, אנחנו עדיין לא יכולים לעשות NLP בצורה מושלמת, למעשה אנחנו עדיין רחוקים מכך.

המצב הנוכחי של NLP

סרבמאלי



POS Tagging: 97%

קטן מסף

Parsing: partial >90%(?)

Semantics: some aspects

- **Entity/relation extraction**
- **Word sense disambiguation**
- **Sentiment analysis**

Speech act analysis: ???

Inference: ???

לרמיר מאוד לפלסר
יותר מורכב...

ד מתקדמים בשלוש אלגוריתמים מרתמונה.

מה אנחנו **לא** יכולים לעשות ב- NLP

← תלוי דיבור.

100% דיוק בתיוג POS:

- "He turned **off** the highway." vs "He turned **off** the light."
  - הוא **ירד** מהכביש המהיר
  - הוא **כיבה** את האור

ניתוח כללי מלא (General complete parsing)

- "A man saw a boy with a telescope."
- משפטים ארוכים יצרים מגוון רב של אפשרויות שיוך

ניתוח סמנטי עמוק ומדויק:

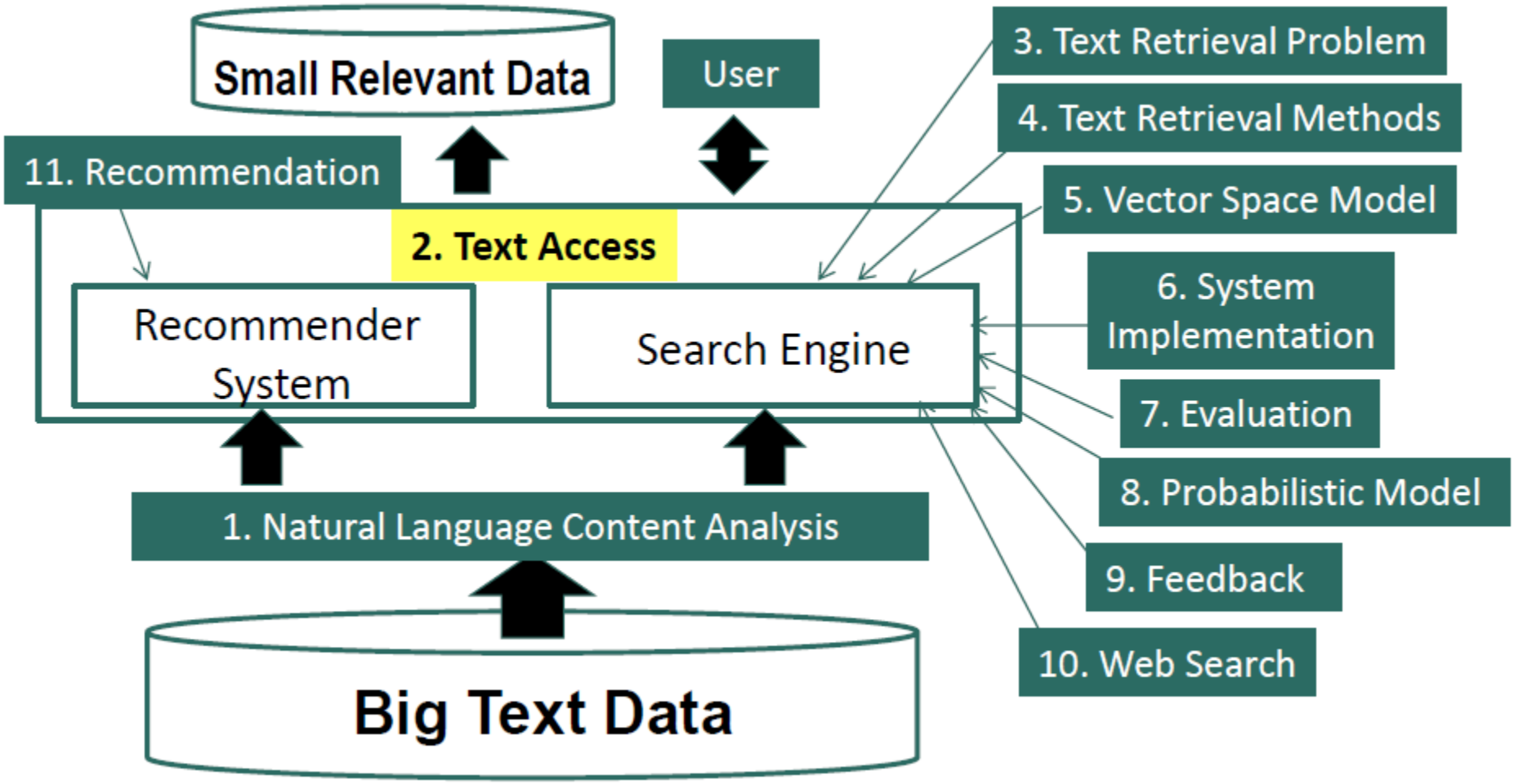
- יהיה לנו קשה ביותר להגדיר את המילה "own" במשפט " John owns a restaurant "

← NLP יציב וכללי נוטה להיות "רדוד" ואילו הבנה "עמוקה" עדיין אינה בנמצא

NLP בשביל אחזור מידע

- חייב להיות יציב, כללי ויעיל ← NLP יחסית רדוד
- " Bag of words" (BOW)
  – בד"כ מספיק טוב לרוב משימות החיפוש (אך לא לכולן!)
- חלק מטכניקות אחזור טקסט יכולות להתמודד עם בעיות NLP באופן טבעי

עם זאת, יש צורך ב- NLP עמוק יותר למשימות חיפוש מורכבות (אבל הוא עדיין לא בנמצא, כמו שהזכרנו קודם)


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

גישה לנתוני טקסט רלוונטיים

כיצד מערכת המכילה טקסט יכולה לעזור למשתמשים לקבל גישה לנתוני הטקסט הרלוונטיים לבקשת המשתמש?

- Push vs. Pull

לעומת שאילתא

- Querying vs. Browsing

מיון לפי ינק

שני מודלים של גישה לטקסט:
Pull vs. Push

- **Pull** Mode (מנועי חיפוש)
  - היוזמה בידי המשתמש
  - צורך במידע בצורה מיידית

- **Push** Mode (מערכות המלצה [Information filtering])
  - היוזמה בידי המערכת
  - צורך במידע יציב או שלמערכת יש ידע טוב על הצורך של המשתמש

Pull Mode:
שאילתות לעומת גלישה/דפדוף

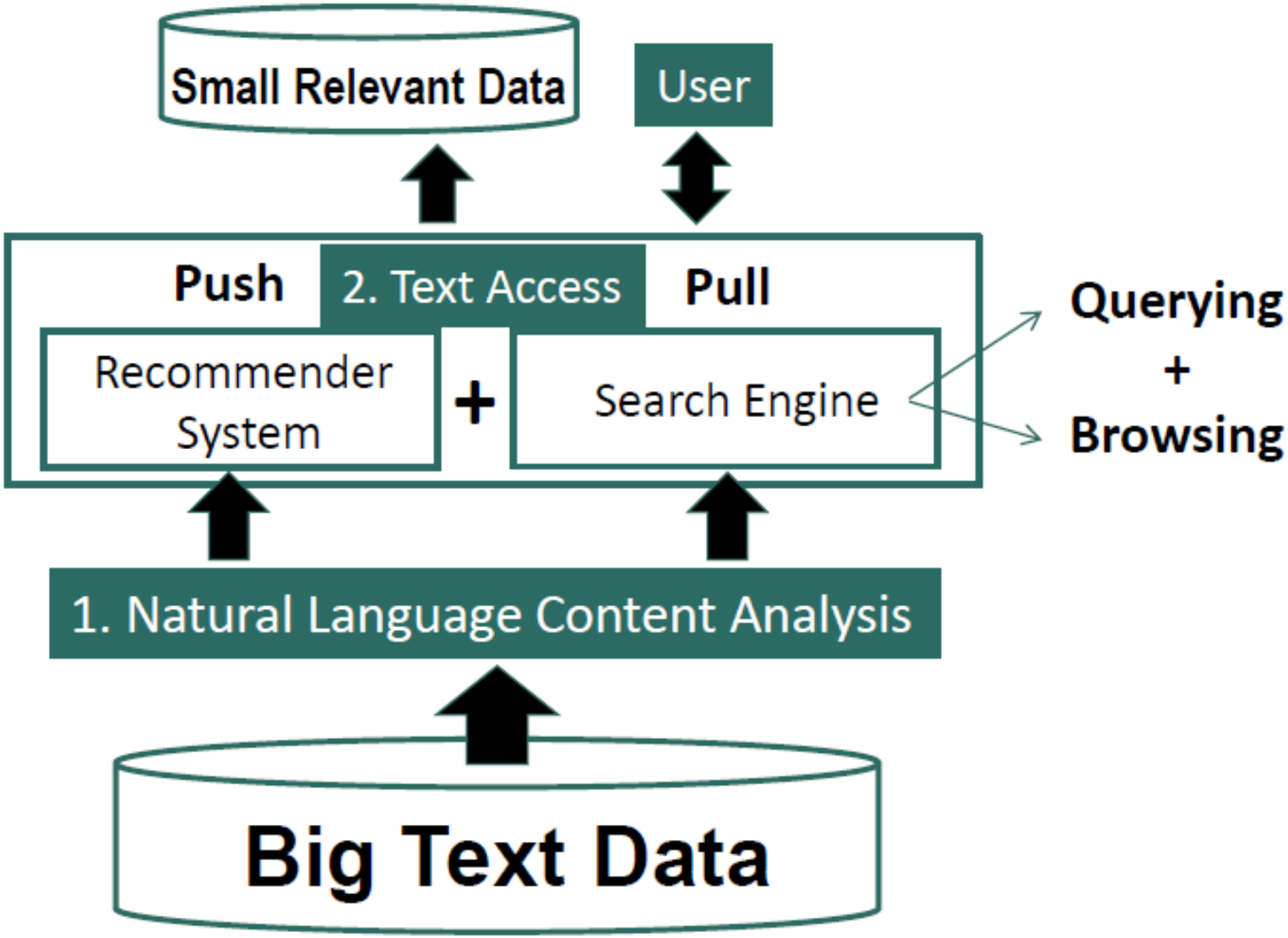
- שאילתות:
  - המשתמש מכניס את השאילתא (מילות מפתח)
  - המערכת מחזירה מסמכים רלוונטיים
    - עובד טוב כאשר המשתמש יודע אלו מילים להזין למערכת

- גלישה/דפדוף:
  - המשתמש מנווט למידע רלוונטי על ידי מעבר בנתיב המופיע במסמכים (דומה למבנה של עץ הקבצים במחשב)
    - עובד טוב כאשר המשתמש רוצה לחקור מידע, אינו יודע באילו מילות מפתח להשתמש, או שאינו יכול להזין שאילתה בצורה נוחה

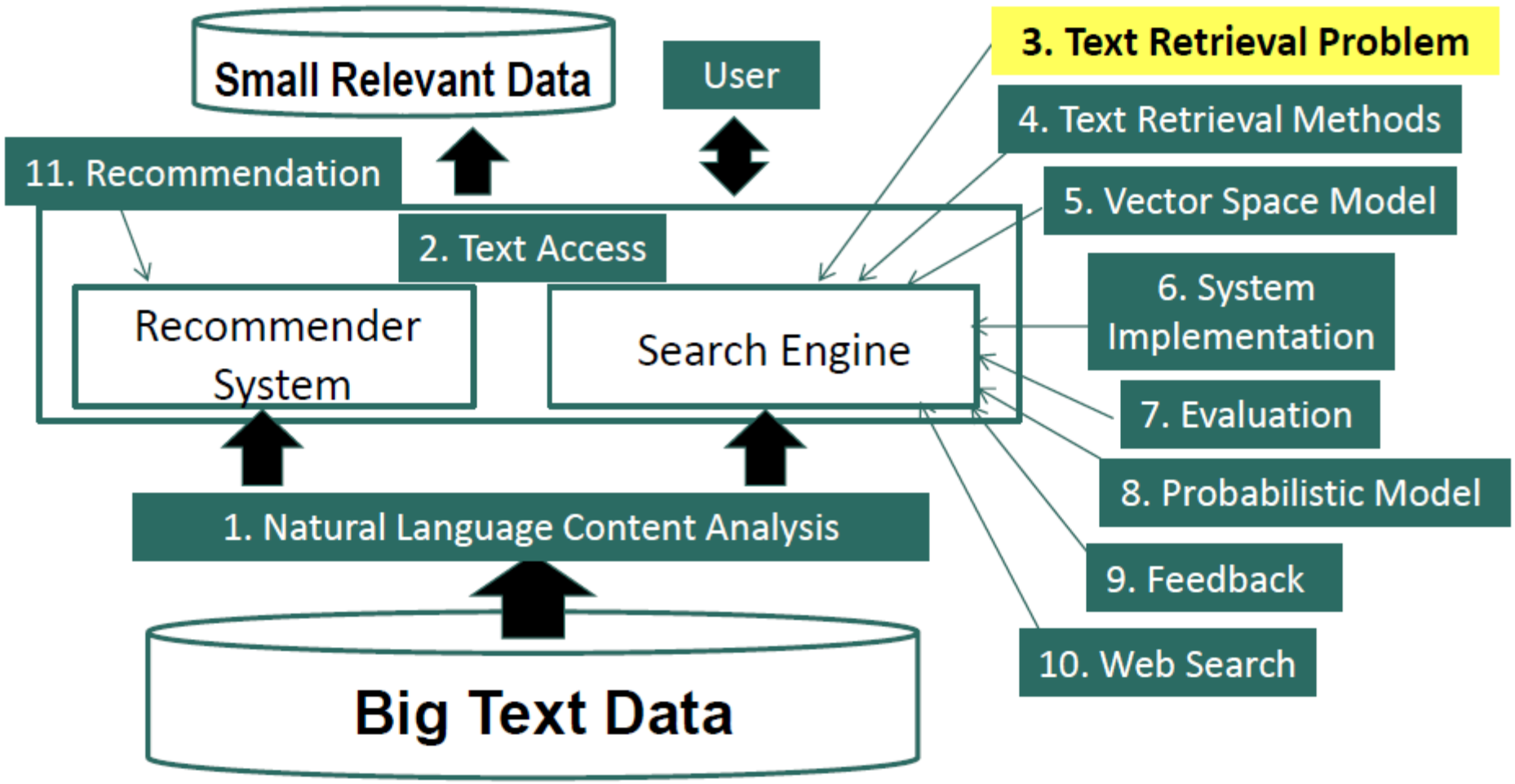
חיפוש מידע ע"י סיור במסמכים

- סיור: מכירים את הכתובת המבוקשת?
  - כן ← קח "מונית" ישירות ליעד המבוקש
  - לא ← תחפש באזור בו אתה נמצא או תיקח "מונית" למקום סמוך ואז תחפש
- חיפוש מידע: האם יודע בדיוק מה אתה רוצה למצוא?
  - כן ← השתמש במילות המפתח הנכונות לשאילתה ומצא את המידע ישירות
  - לא ← עיין באינטרנט או התחל בשאילתה גסה ואז דפדף

Pull and Push



Text Retrieval



Text Retrieval

- מהוא אחזור טקסט?
- אחזור **טקסט** לעומת אחזור **מסד נתונים**
- **בחירת** מסמכים לעומת **דירוג** מסמכים

מה זה אחזור טקסט
Text Retrieval (TR)?

- קיים אוסף של מסמכי טקסט
- המשתמש מזין שאילתה המגדירה את המידע המבוקש
- מערכת מנועי החיפוש מחזירה מסמכים רלוונטיים למשתמשים שאילתה טקסט.
- נקרא לעתים קרובות "אחזור מידע" (IR), אולם IR הוא תחום רחב יותר (מכיל אחזור מידע מדיבור, תמונות ועוד)
- בתעשייה זה מכונה "טכנולוגיית חיפוש"

אחזור **טקסט** (TR) לעומת אחזור **מסד נתונים**

מֵידָע	
טקסט	**מסד נתונים**
טקסט לא מובנה/חופשי	נתונים מובנים
רב-משמעותיות/עמימות	מוגדר היטב

שאילתא	
טקסט	**מסד נתונים**
רב-משמעותיות/עמימות	מוגדרת היטב
לא מושלמת	מושלמת

תשובות	
טקסט	**מסד נתונים**
מסמכים רלוונטיים	רשומות תואמות

אחזור **טקסט** (TR) לעומת אחזור **מסד נתונים**

- TR היא בעיה אמפירית
  - לא ניתן להוכיח מתמטית ששיטה אחת טובה יותר מהשנייה
  - עלינו להסתמך על **הערכה אמפירית** בה מעורבים משתמשים!

ייצוג פורמאלי של TR

נרצה שיהיה כמה שיותר מוגדר בצורה אינטרנטית

- **Vocabulary**: $V=\{w_1, w_2, \ldots, w_N\}$ מילים בשפה
- **Query**: $q = q_1,\ldots,q_m$ where $q_i \in V$ מילים
- **Document**: $d_i = d_{i1},\ldots,d_{im}$, where $d_{ij} \in V$ מילים
- **Collection**: $C= \{d_1, \ldots, d_M\}$ מסמכים
- **Set of Relevant documents**: $R(q) \subseteq C$ המסמכים הרלוונטיים

היא רלוונטי מול המסמכים

– בד"כ לא ידוע ותלוי במשתמש

תלוי במשתמש

– השאליתא היא "רמז" לקבוצת המסמכים הרלוונטיים

- **Task** = compute R'(q), an approximation of R(q)

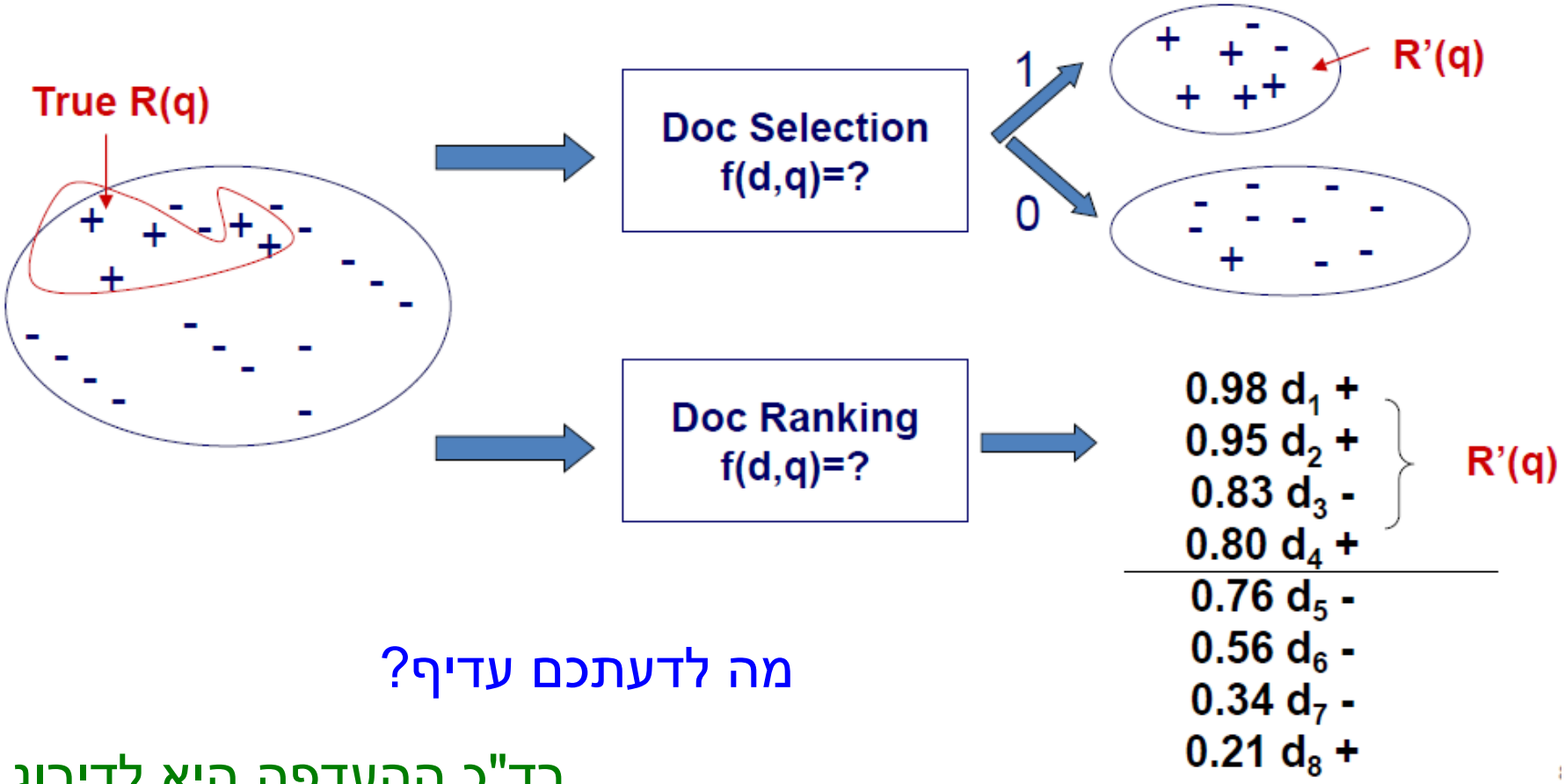
נרצה פונקציה שתיתן לנו הערכה איך כל המסמכים הרלוונטיים

לפי השאילתא..

כיצד לחשב את R’(q)?

- גישה **בינארית**
  - $R'(q)=\{d \in C \mid f(d,q)=1\}$, where $f(d,q) \in \{0,1\}$
    - פונקציה המספקת אינדיקציה או מסווג בינארי
  - על המערכת להחליט אם מסמך רלוונטי או לא (רלוונטיות מוחלטת) רלוונטי/לא.
- **דירוג** המסמכים
  - $R'(q)=\{d \in C \mid f(d,q)>\theta\}$, where $f(d,q) \in \Re$
    - היא פונקציית מדד רלוונטיות; θ הוא סף שנקבע על ידי המשתמש (למשל המשתמש מעיין רק ב- 15 מסמכים ראשונים)
    - המערכת צריכה להחליט אם מסמך אחד רלוונטי יותר מאשר אחר (רלוונטיות יחסית)

בחירת מסמכים לעומת דירוג מסמכים



מה לדעתכם עדיף?

בד"כ ההעדפה היא לדירוג

מה פשר הדגיש כי'כה

בעיות בבחירת מסמכים (0/1)

סביר להניח שהמסווג לא מדייק

– שאילתה מוגבלת מידי ← אין מסמכים רלוונטיים מאוחזרים

– שאילתה רחבה מידי ← יותר מידי מסמכים רלוונטיים מאוחזרים

- קשה למצוא את הנקודה הנכונה בין שתי האופציות

גם אם המסווג מדויק, לא כל המסמכים הרלוונטיים, רלוונטיים באותה מידה (רלוונטיות היא עניין של דירוג!)

- יש צורך בתיעדוף

←לפיכך, דירוג בדרך כלל עדיף

הצדקה תיאורטית לשיטת הדירוג

עיקרון דירוג מסמכים לפי הסתברות [Robertson 77]

- אחזור רשימה של מסמכים רלוונטיים לשאילתה בסדר יורד

בהסתברות, היא האסטרטגיה האופטימלית בשתי **ההנחות** הבאות:

- התועלת של מסמך (למשתמש) אינה תלויה בתועלת של כל מסמך אחר
- משתמש גולש בתוצאות בסדר יורד, הראשון הכי מתאים הסתברותית השני פחות וכו'

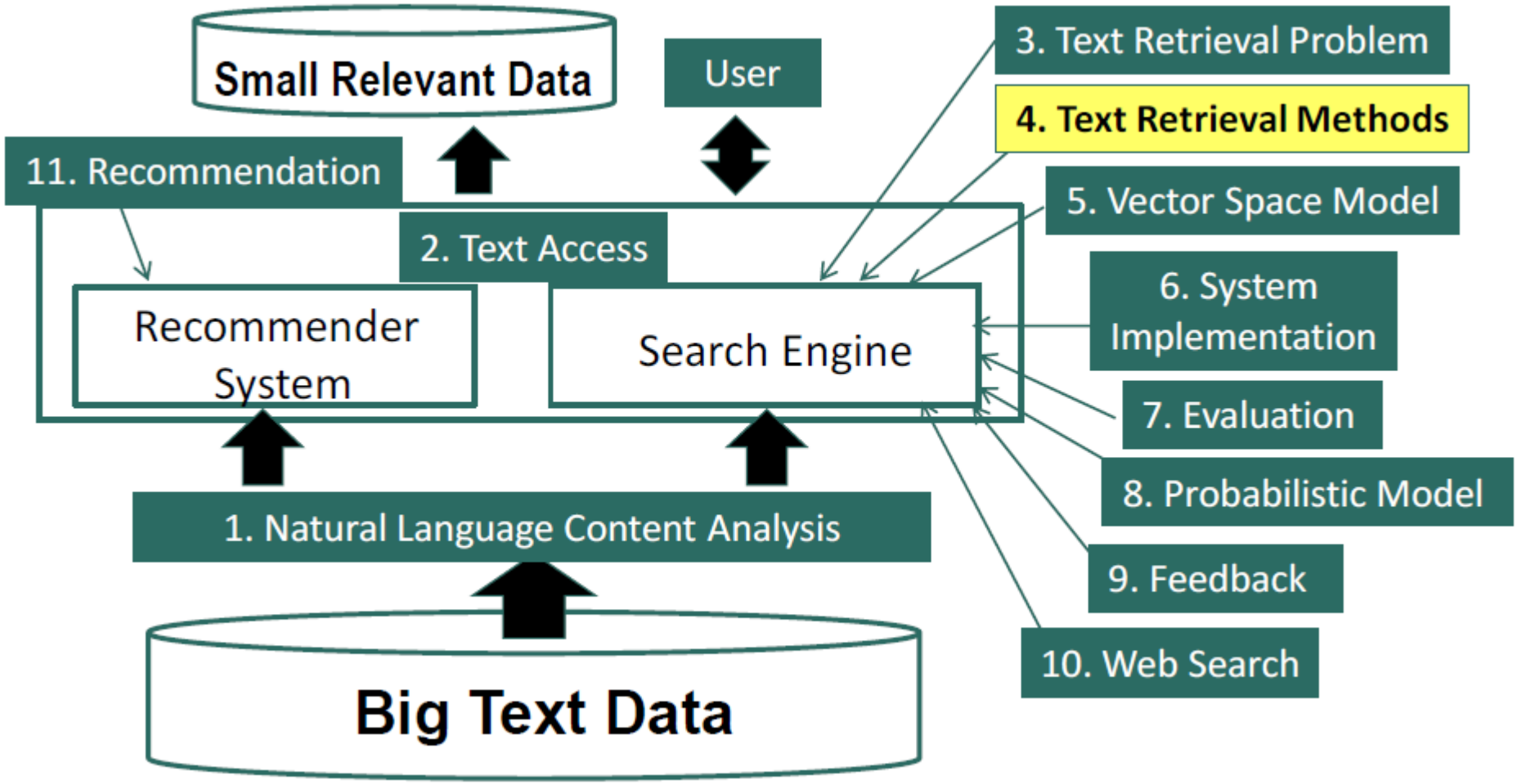
האם שתי ההנחות הללו מתקיימות?

מה ראינו עד כעת בנושא TR

- אחזור טקסטים הוא בעיה מוגדרת אמפירית
  - איזה אלגוריתם עדיף?
    - נשפט על ידי המשתמשים
- בדרך כלל מעדיפים דירוג מסמכים
  - עוזר למשתמש בהערכת עדיפות של מסמכים המופיעים בתוצאות החיפוש
  - עוקף את הקושי בקביעת הרלוונטיות המוחלטת (המשתמשים עוזרים להחליט על הפסקת הרשימה המדורגת)

אתגר מרכזי: תכנון פונקציית דירוג יעילה $f(q,d) =?$

שיטות לאחזור מידע



כיצד לבנות פונקציית דירוג

- **Vocabulary**: $V=\{w_1, w_2, \ldots, w_N\}$
- **Query**: $q = q_1,\ldots,q_m$ where $q_i \in V$
- **Document**: $d_i = d_{i1},\ldots,d_{im}$, where $d_{ij} \in V$
- **Ranking function:** $f(d,q) \in \Re$

- פונקציית דירוג טובה צריכה לדרג מסמכים רלוונטיים מעל מסמכים לא רלוונטיים
- אתגר מרכזי: כיצד למדוד את הסבירות שמסמך d רלוונטי לשאילתה q

מודל אחזור = לתת הגדרה פורמאלית של רלוונטיות
(הגדרה חישובית של רלוונטיות)

ישנם כמה וכמה מודלים שונים של רלוונטיות

- **מודל הדמיון:** f(q,d) = similarity(q,d)
  - שימוש בווקטורים
- **מודל הסתברותי:** f(d,q) = p(R=1|d,q), where R∈{0,1}
  - מודל הסתברותי קלאסי
  - מודל שפה
  - ענף ממודל האקראיות
- **מודל הסקה הסתברותי:** f(q,d) = p(d→q) [לכאורה הכיוון הפוך], השאילתה "נובעת" מהמסמך
- **מודל אקסיומטי:** פונקציית הדירוג, f(q,d) חייבת לספק קבוצה של אילוצים

מערכות רול"פ - מערכת מבוססת כללים.

מודלים שונים אלו נוטים לפונקציות דירוג דומות הכוללות משתנים דומים

גישות נפוצות בדגמי אחזור חדשים

f(q="presidential campaign news", d)

"Bag of Words"

g("presidential", d) g("campaign", d) g("news", d)

How many times does "presidential" occur in d?

Term Frequency (TF): c("presidential", d)

How long is d? **Document length**: |d|

How often do we see "presidential" in the entire collection?

Document **F**requency: df("presidential") compare to

p("presidential"|collection)

איזה מודלים עובדים הכי טוב

- כאשר הם מגיעים לאופטימיזציה, הדגמים הבאים מגיעים לתוצאות טובות באותה מידה :[Fang et al. 11]
  - Pivoted length normalization
  - BM25
  - Query likelihood
  - PL2
- BM25 הכי פופולארית

מה שלמדנו עד כעת בשיטות לאחזור מידע

- תכנון פונקציית דירוג f(d,q) דורש מראש הגדרה חישובית של רלוונטיות (מודל אחזור)
- ישנם מספר מודלים שיעילים באותה מידה
- פונקציות דירוג חדשות נוטות להסתמך על:
  - ייצוג של קבוצת מילים (BoW)
  - שכיחות ביטוי (TF) ושכיחות מסמכים המכילים את המילה.
  - אורך המסמך (במילים)