Overview of Text Mining and Analytics

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

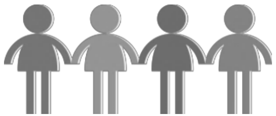
Text Mining and Analytics

- Text mining (TM) ≈ Text analytics ➔ to get
  - ✓ **high-quality info** (minimizes human effort (text consumption))
  - ✓ **actionable knowledge** (optimal decision making)
- Related to **text retrieval (TR)**, an essential component of text mining
  - TR can be a preprocessor for TM
  - TR needed for knowledge provenance

Text vs. Non-Text Data: Humans as Subjective "Sensors"

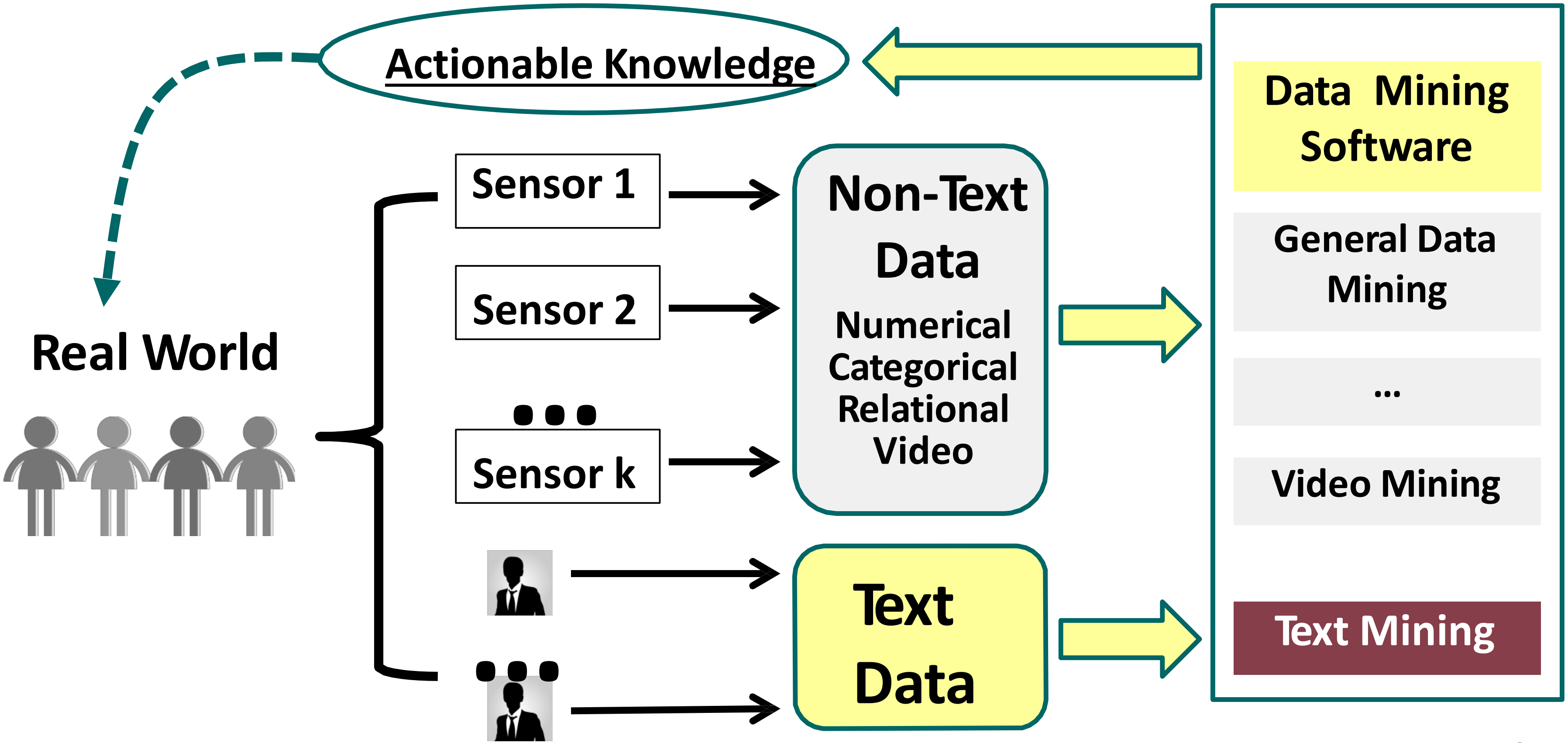
Real World $\xrightarrow{\textbf{Sense}}$ **Sensor** $\xrightarrow{\textbf{Report}}$ **Data**

Thermometer, Geo Sensor, Network Sensor

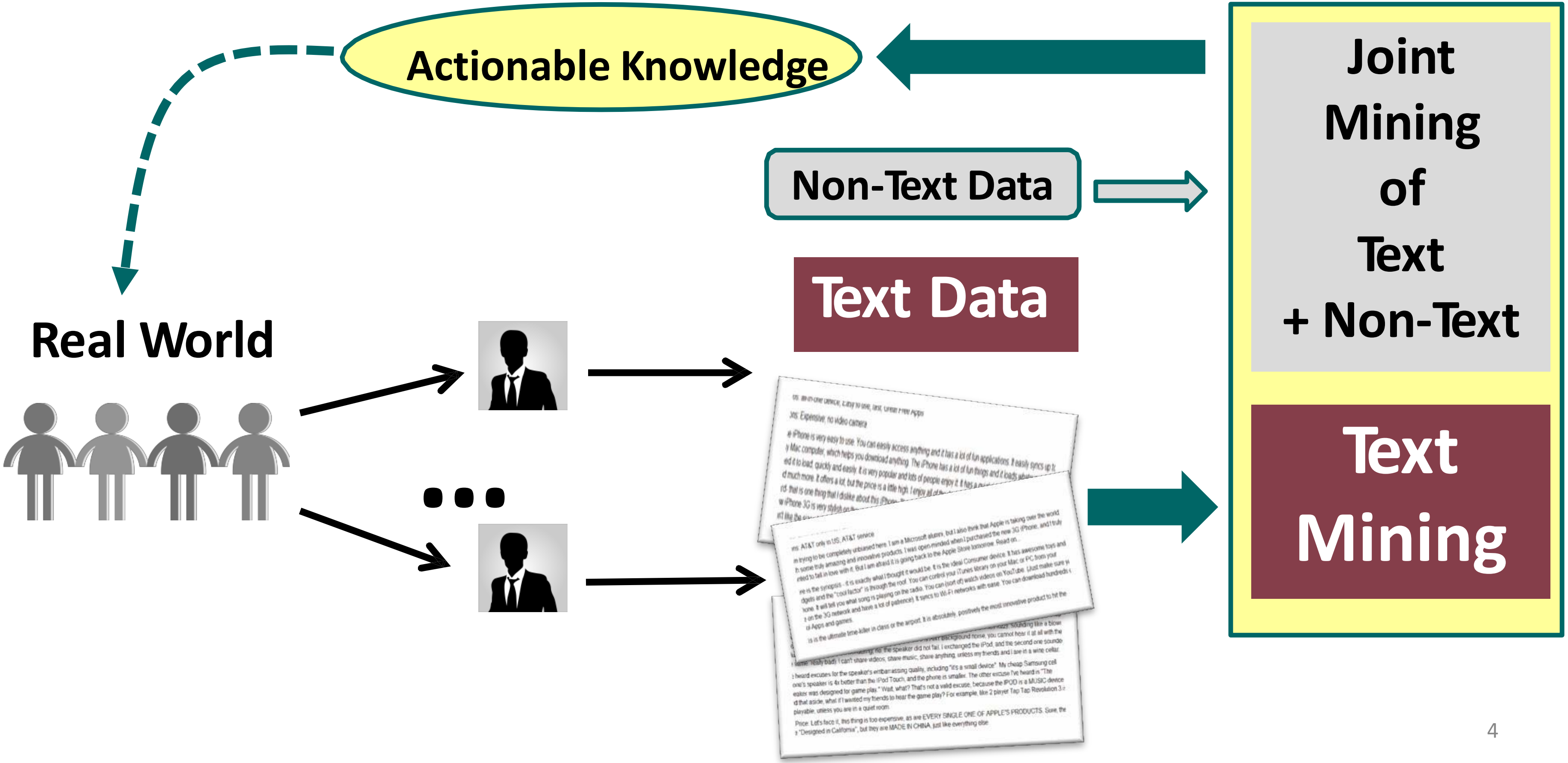
 $\xrightarrow{\textbf{Perceive}}$  $\xrightarrow{\textbf{Express}}$ 

"Human Sensor"

General Data Mining

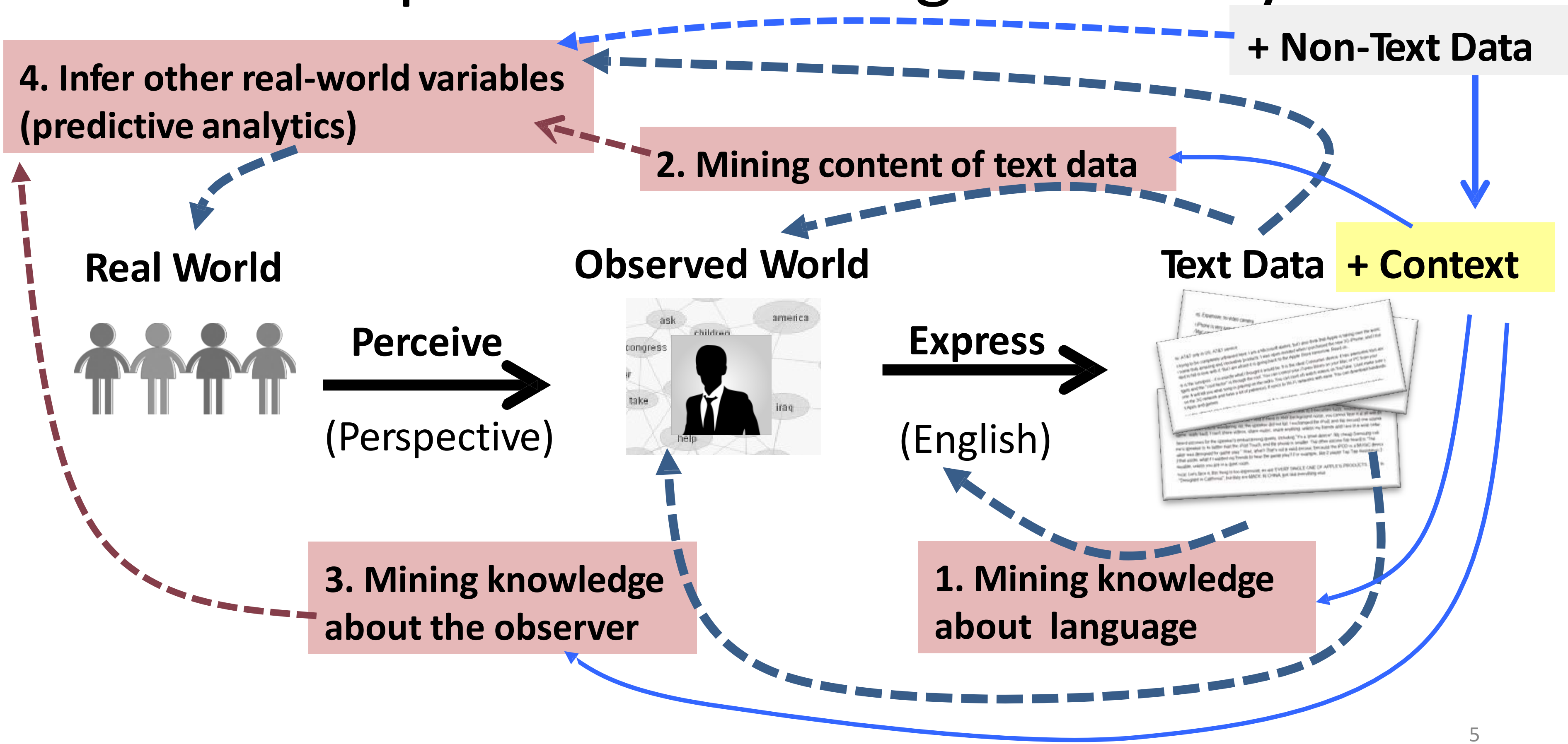


Text Mining

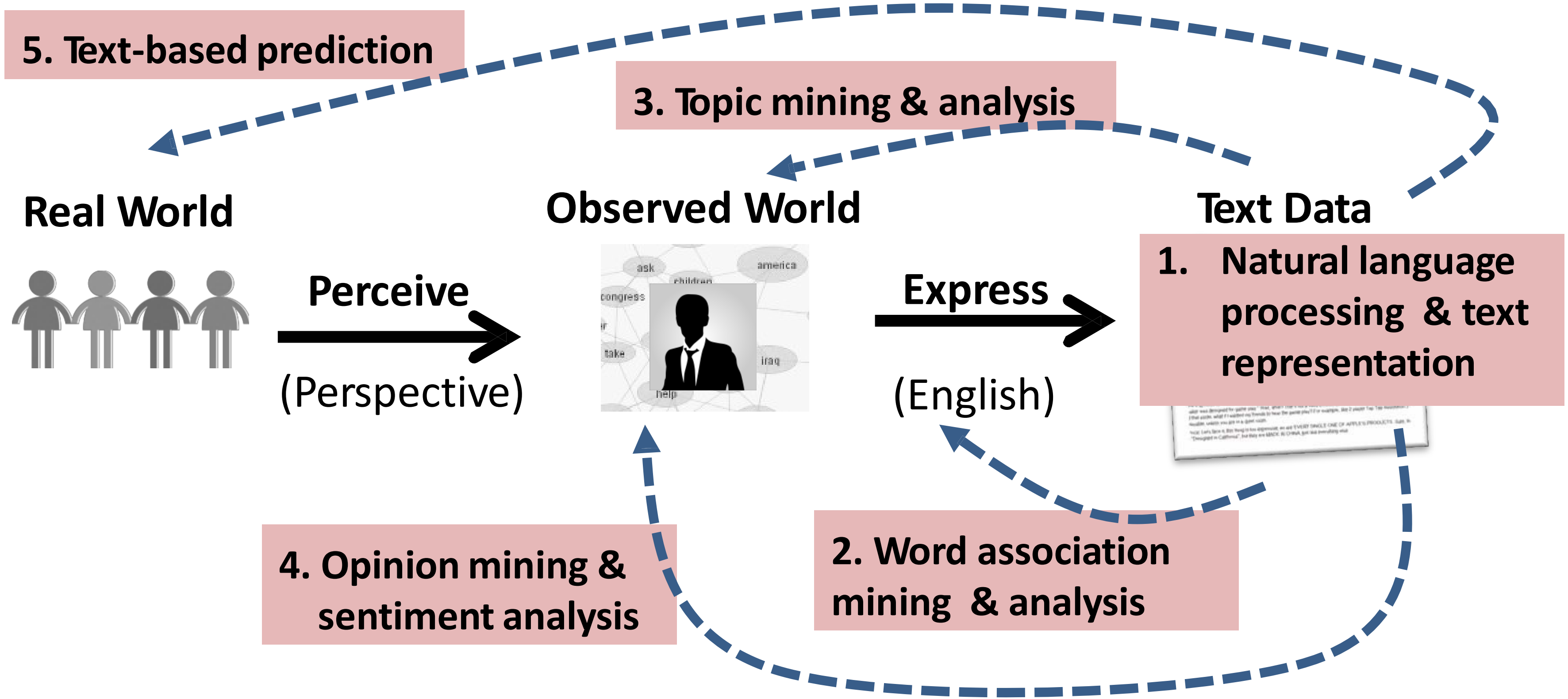


Actionable Knowledge
Joint Mining of Text + Non-Text
Non-Text Data
Text Data
Real World
...
Text Mining

Landscape of Text Mining and Analytics

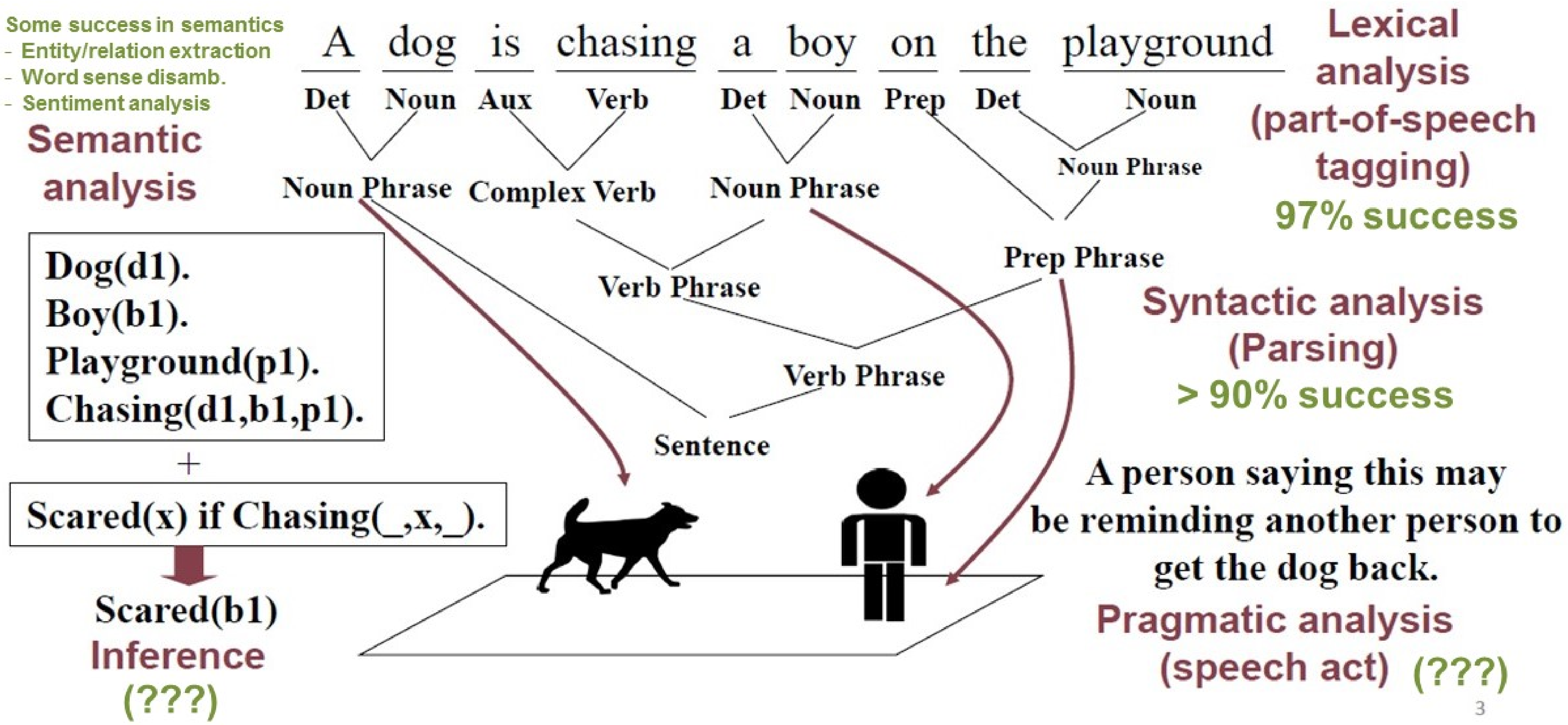


Topics Covered in This Course



NLP

Basic Concepts in NLP



NLP Is Difficult!

Natural language – for efficient human communication. Therefore we

- **omit common knowledge**, assuming the reader has it (NLP requires common knowledge and inferences, thus working for very limited domains)
- **keep ambiguities**, assuming the reader knows them (ambiguity is a *killer*).

Examples:

- **Word-level** ambiguity: "design" (N vs. V), "root" (multiple meanings)
- **Syntactic** ambiguity:
  - "natural language processing"
  - "A man saw a boy *with a telescope*." (PP Attachment)
- **Anaphora** resolution: "John persuaded Bill to buy a TV for *himself*."
- **Presupposition**: "He has quit smoking" implies he smoked.

- NLP - foundation of text mining
- **Shallow Statistical NLP (**can be done in large scale ➔ more broadly applicable) is the basis; humans help as needed.
- Computers are far from understanding natural language

***Shallow* NLP is robust and general**
***Deep* understanding doesn't scale up**

What we can't do (yet?)

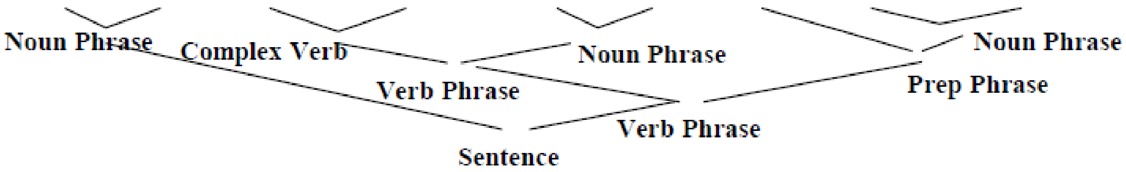
- 100% POS tagging ("He turned off the highway." vs "He turned off the fan.")
- General complete parsing ("A man saw a boy with a telescope.")
- Precise deep semantic analysis (how precisely define the meaning of "own" in "John owns a restaurant"?)

Text Representation

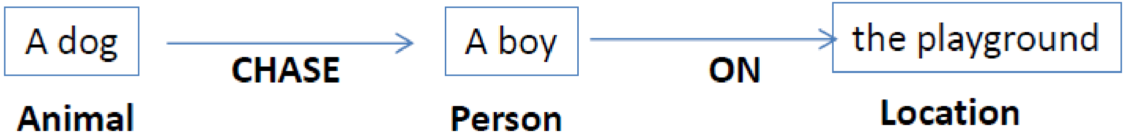
A dog is chasing a boy on the playground — **String of characters**

A dog is chasing a boy on the playground — **Sequence of words**

Det Noun Aux Verb Det Noun Prep Det Noun — **+ POS tags**



+ Syntactic structures



+ Entities and relations

Dog(d1). Boy(b1). Playground(p1). Chasing(d1,b1,p1). — **+ Logic predicates**

Speech Act = REQUEST — **+ Speech acts**

Deeper NLP: requires more human effort; less accurate

Closer to knowledge representation

Text Representation and Enabled Analysis

- Text representation determines the mining algorithms
- **Multiple ways** of text representation (string, words, syntactic structures, ER graphs, predicates) **are combined** in real applications
- This course - **word-based representation**
  - **General and robust** (any natural language)
  - **No/little manual effort**
  - **Powerful** for many, but not all applications)
  - **Can be combined** with more sophisticated representations

This course

Text Rep	Generality	Enabled Analysis	Examples of Application
String		String processing	Compression
Words		Word relation analysis; topic analysis; sentiment analysis	Thesaurus discovery; topic and opinion related applications
+ Syntactic structures		Syntactic graph analysis	Stylistic analysis; structure-based feature extraction
+ Entities & relations		Knowledge graph analysis; information network analysis	Discovery of knowledge and opinions about specific entities
+ Logic predicates		Integrative analysis of scattered knowledge; logic inference	Knowledge assistant for biologists

Word Association Mining and Analysis

Basic Complementary Word Relations

- **Paradigmatic**: A & B can **substitute** each other (same class):
  - "Cat" and "dog", "Monday" and "Tuesday"
- **Syntagmatic**: A & B can be combined with each other (related semantically):
  - "Cat" and "sit", "car" and "drive"
- Can be generalized to **describe any relations** in a language

Why Mine Word Associations?

- **Improve accuracy of NLP**
  - Grammar learning, POS tagging, parsing, entity recognition, acronym expansion
- **Directly useful in TR and text mining**
  - TR - e.g. suggest a <u>query variation</u>
  - Automatic <u>topic map for browsing</u>: words as nodes and associations as edges
  - Compare and summarize <u>opinions</u> (e.g., strong positive and negative associations with "battery" in iPhone reviews)

Mining Word Associations: Intuitions

Paradigmatic: similar context

My **cat** eats fish on Saturday
His **cat** eats turkey on Tuesday
My **dog** eats meat on Sunday
His **dog** eats turkey on Tuesday
…

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

dog:
My ___ eats meat on Sunday
His ___ eats turkey on Tuesday
…

Similar **left** context

Similar **Right** context

Similar **General** context

How similar are context ("cat") and context ("dog")?
How similar are context ("cat") and context ("computer")?

Mining Word Associations: Intuitions

Syntagmatic: correlated occurrences

My cat **eats** fish on Saturday
His cat **eats** turkey on Tuesday
My dog **eats** meat on Sunday
His dog **eats** turkey on Tuesday
...

My __ **eats** __ on Saturday
His __ **eats** __ on Tuesday
My __ **eats** __ on Sunday
His __ **eats** __ on Tuesday
...

What words tend to occur to the **left** of "**eats**"?

What words to the **right?**

Whenever "**eats**" occurs, what **other words** also tend to occur?
How helpful is the occurrence of "**eats**" for predicting occurrence of "**meat**"?
How helpful is the occurrence of "**eats**" for predicting occurrence of "**text**"?

Mining Word Associations:

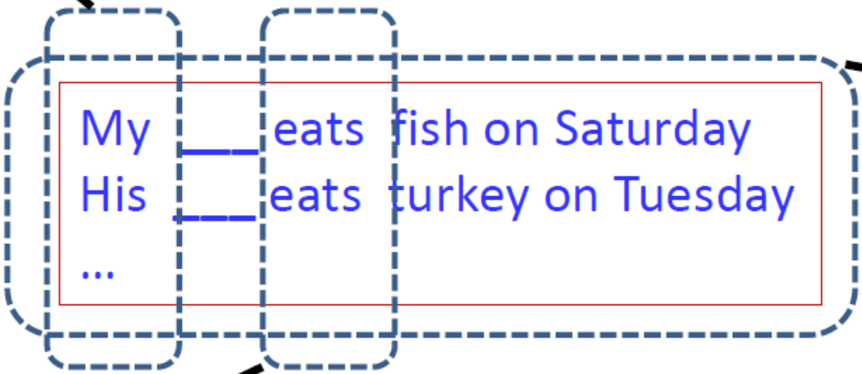
- **Paradigmatic**
  - Words represented by **context**
  - Compute **context similarity**
  - Words with **high context similarity** likely have paradigmatic relation
- **Syntagmatic**
  - Compute **co-occurrence** of two words in a context (sentence, paragraph, etc.)
  - Compare their **co-occurrences** with individual occurrences
  - Words with **high co-occurrences / relatively low individual occurrences** likely have syntagmatic relation
- Paradigmatically related words tend to have syntagmatic relation with the same words ➔ **joint discovery** of the two relations
- There are many implementations!

Paradigmatic Relation Discovery

Word Context as "Pseudo Document"

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

cat:



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

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

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")) +

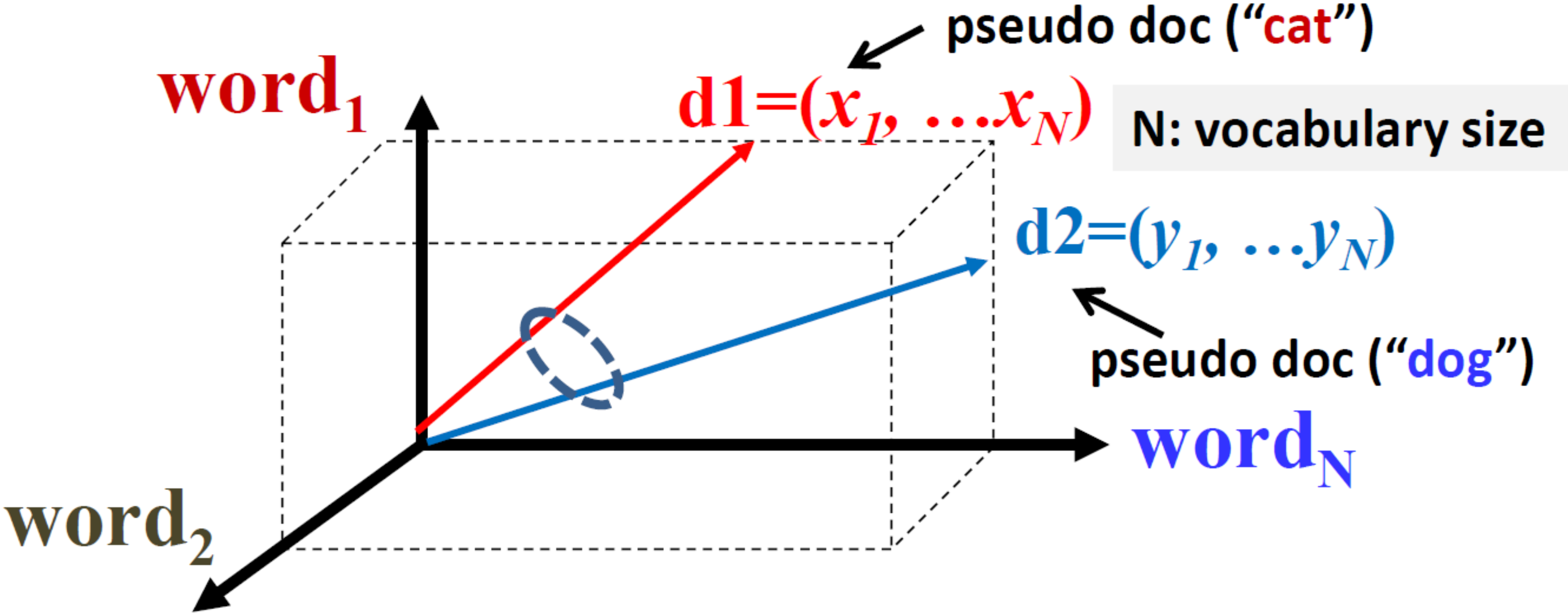
…

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

High sim(word1, word2)

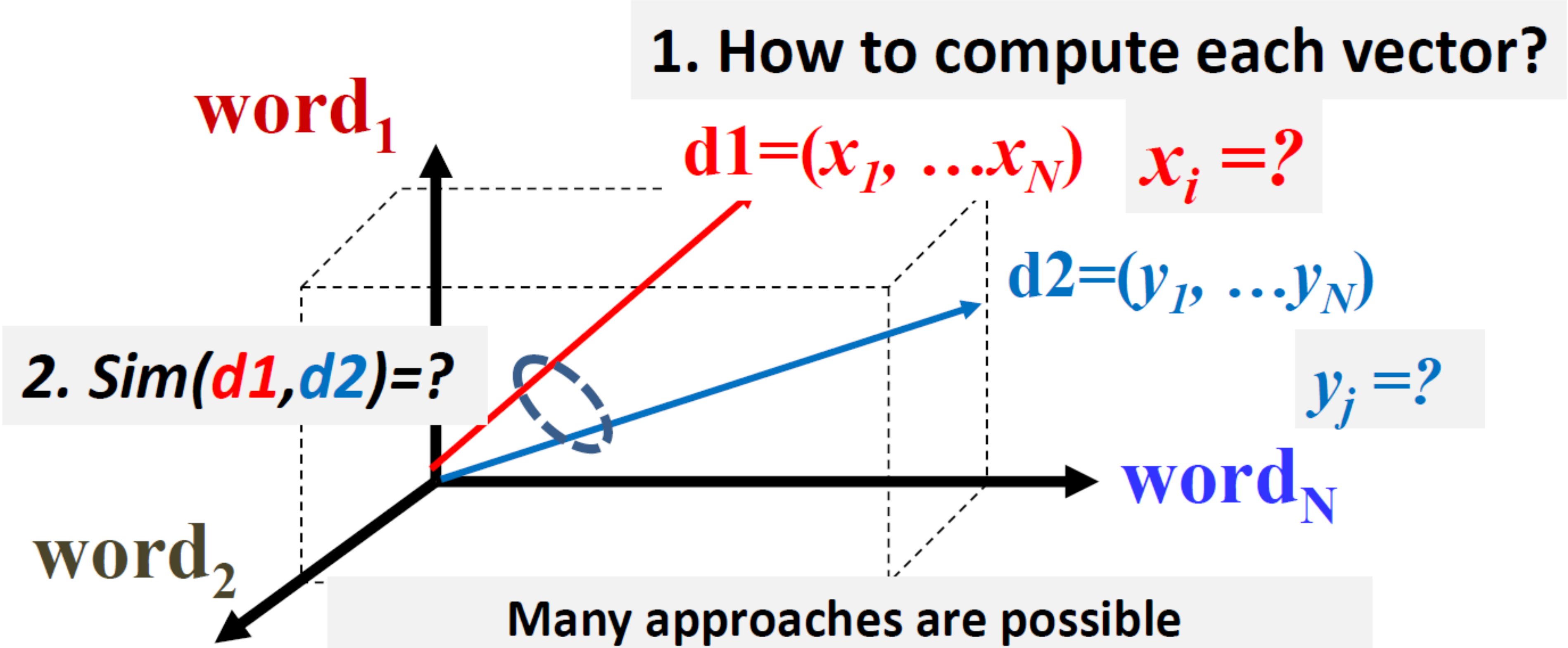
➔ word1 and word2 are **paradigmatically related**

Bag of Words ➔ Vector Space Model (VSM)



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

VSM for Paradigmatic Relation Mining



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

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

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

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?

- **Makes sense**: the more overlap, the higher the similarity.
- However:
  – It favors **matching one frequent term very well** over matching more distinct terms.
  – 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

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

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

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.

Improving EOWC with Retrieval Heuristics

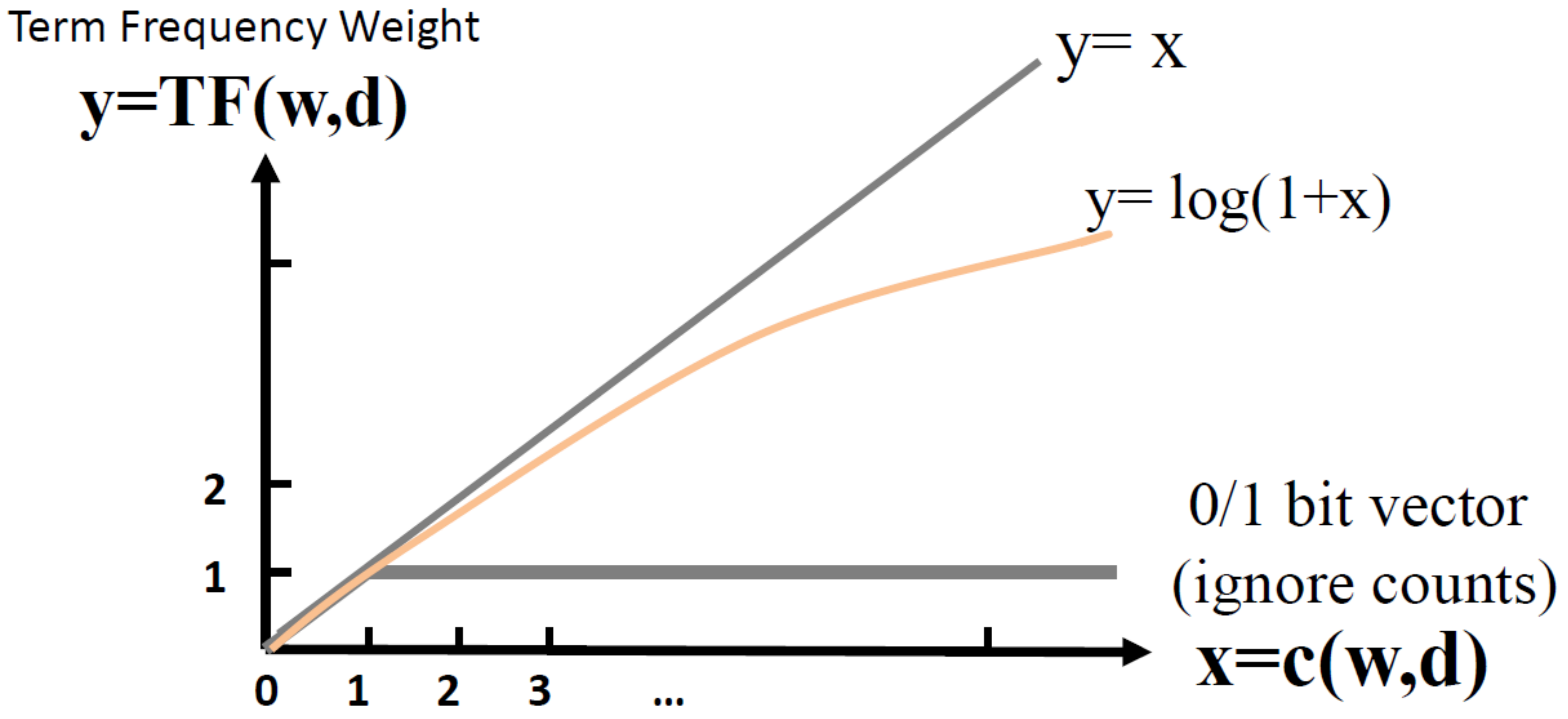
- It favors <u>matching frequent terms</u> very well over matching more distinct terms.

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

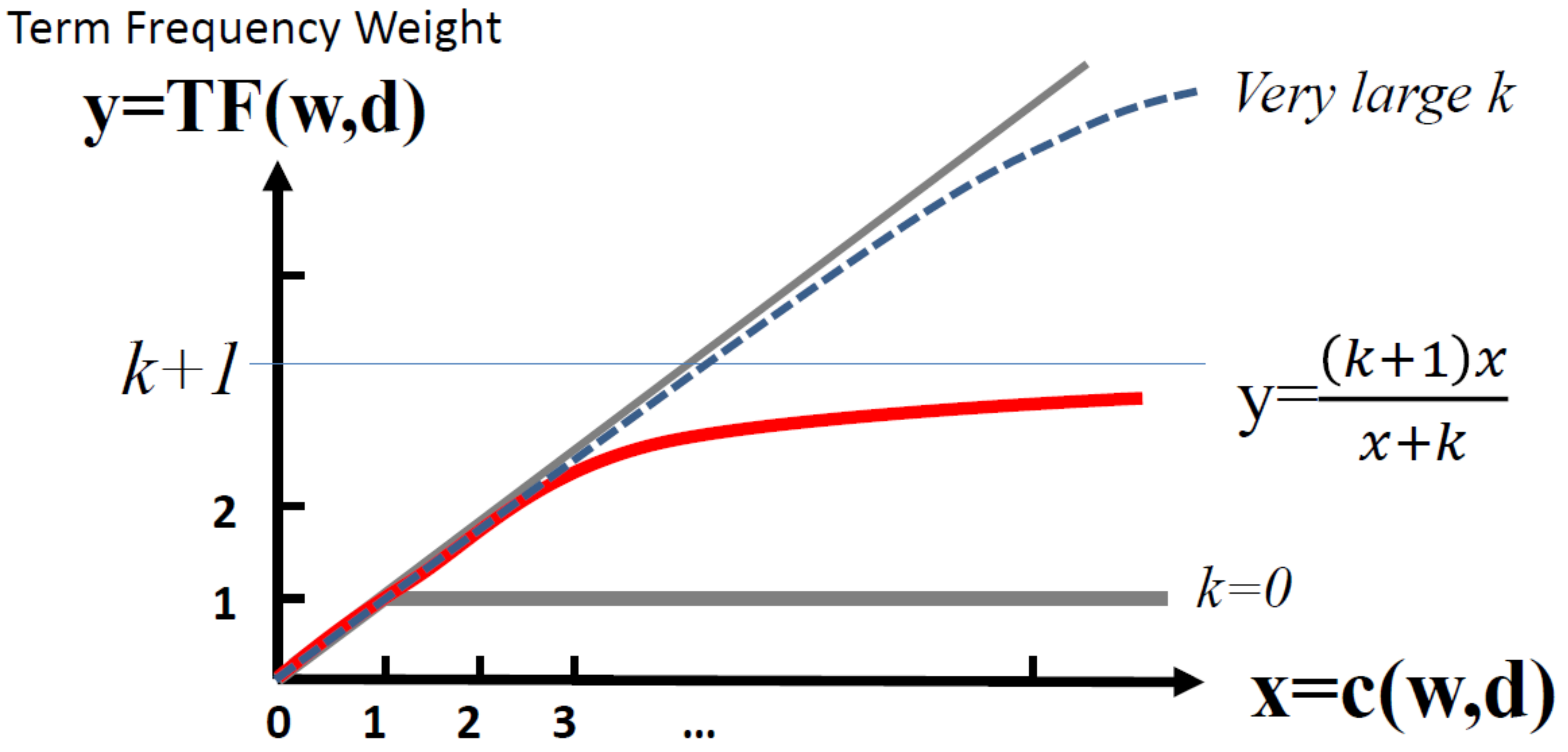
- It <u>treats every word equally</u> (overlap on “the” isn’t as so meaningful as overlap on “eats”).

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

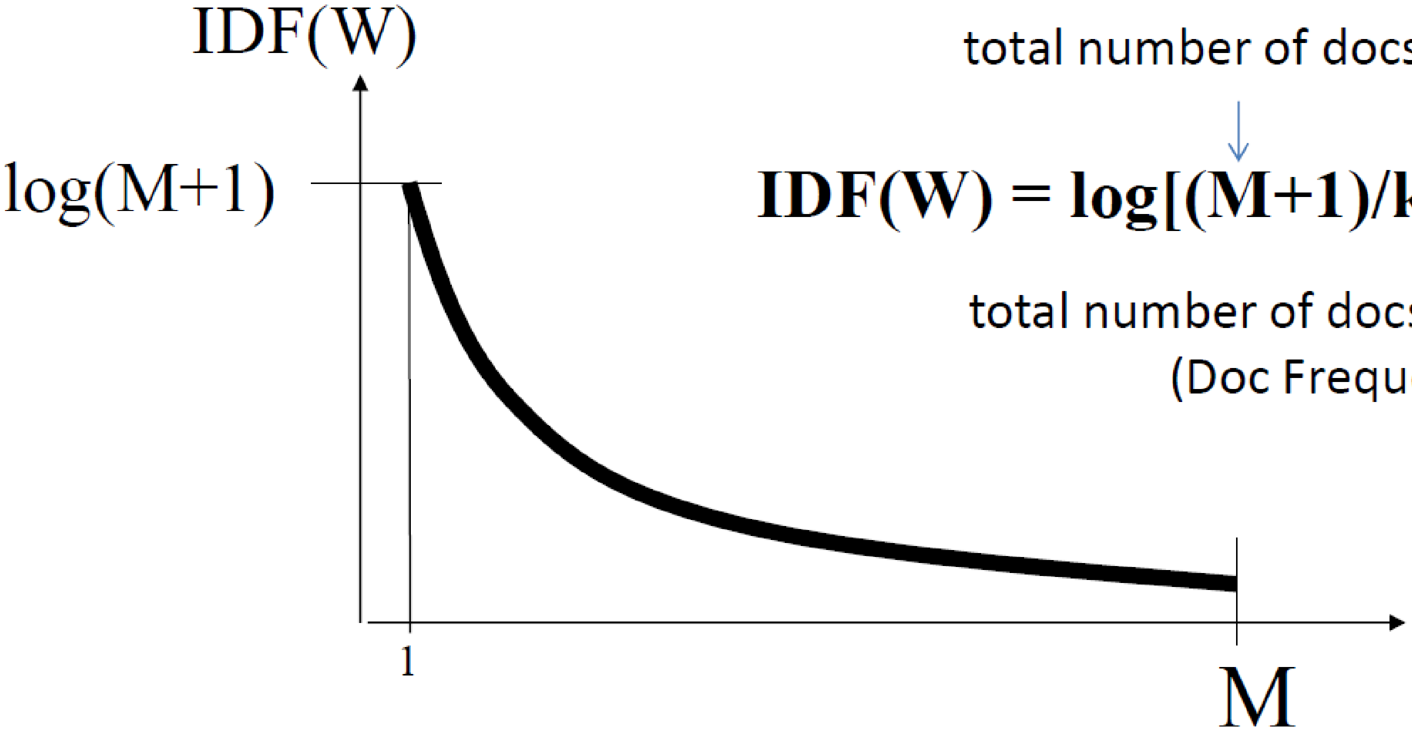
TF Transformation: c(w,d)➔TF(w,d)



TF Transformation: BM25 Transformation



IDF Weighting: Penalizing Popular Terms



total number of docs in collection

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

total number of docs containing W
(Doc Frequency)

Adapting BM25 Retrieval Model for Paradigmatic Relation Mining

$d1=(x_1, \ldots x_N)$

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

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

$b \in [0,1]$

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

$d2=(y_1, \ldots y_N)$

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

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

Summary

- 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 implementations
- 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 discovered as a “by product”