Overview Text Mining and Analytics

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

Text Mining and Analytics

- Text mining ≈ Text analytics
- Turn text data into **high-quality information** or **actionable knowledge**
  - **Minimizes human effort** (on consuming text data)
  - Supplies knowledge for **optimal decision making**
- Related to **text retrieval**, which is an essential component in any text mining system
  - Text retrieval can be a preprocessor for text mining
  - Text retrieval is needed for knowledge provenance

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

Real World —Sense→ **Sensor** —Report→ **Data**

Weather  ⇢ **Thermometer** ⇢ 3°C , 15°F, …

Locations  ⇢ **Geo Sensor** ⇢ 41°N and 120°W ….

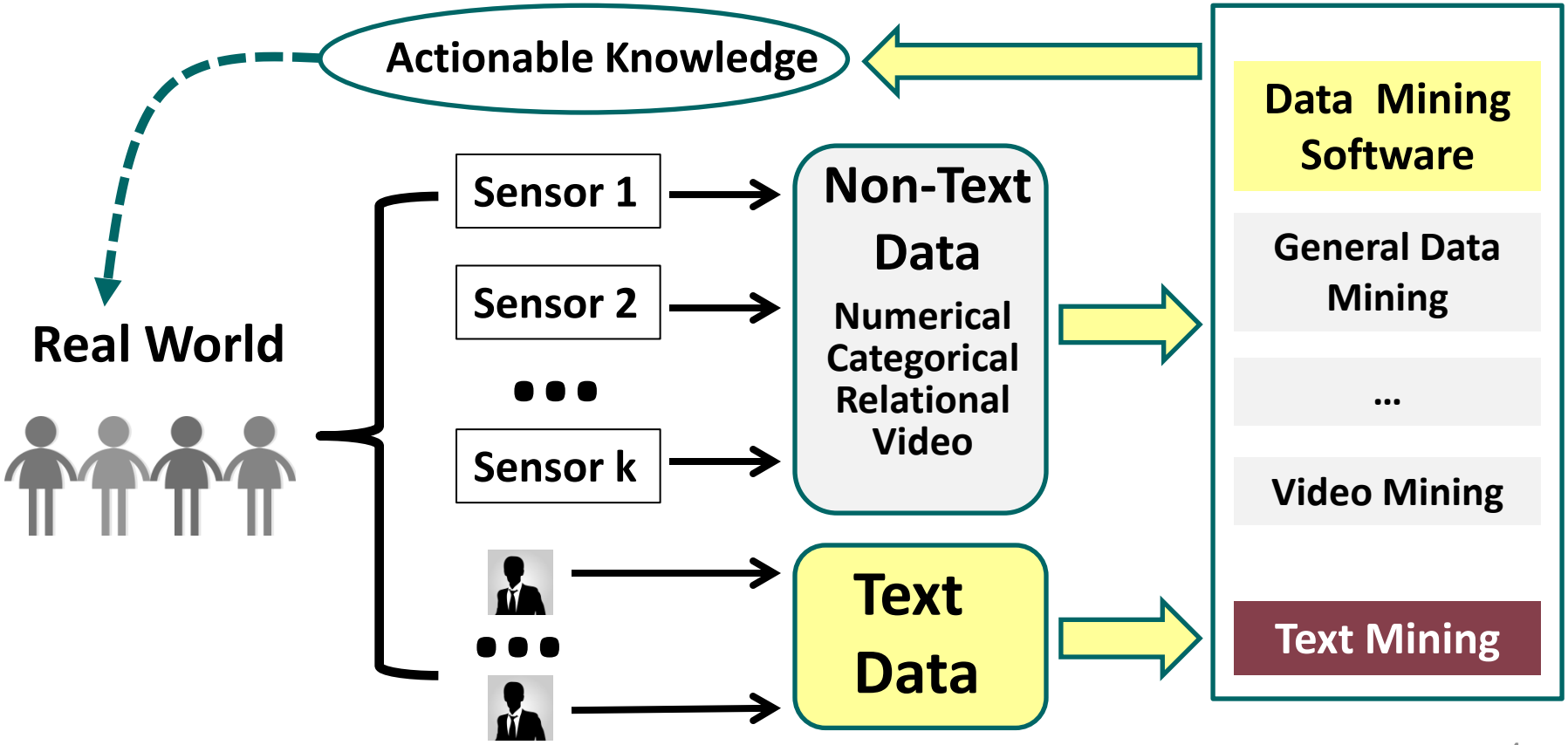
Networks  ⇢ **Network Sensor** ⇢ 01000100011100

Perceive →

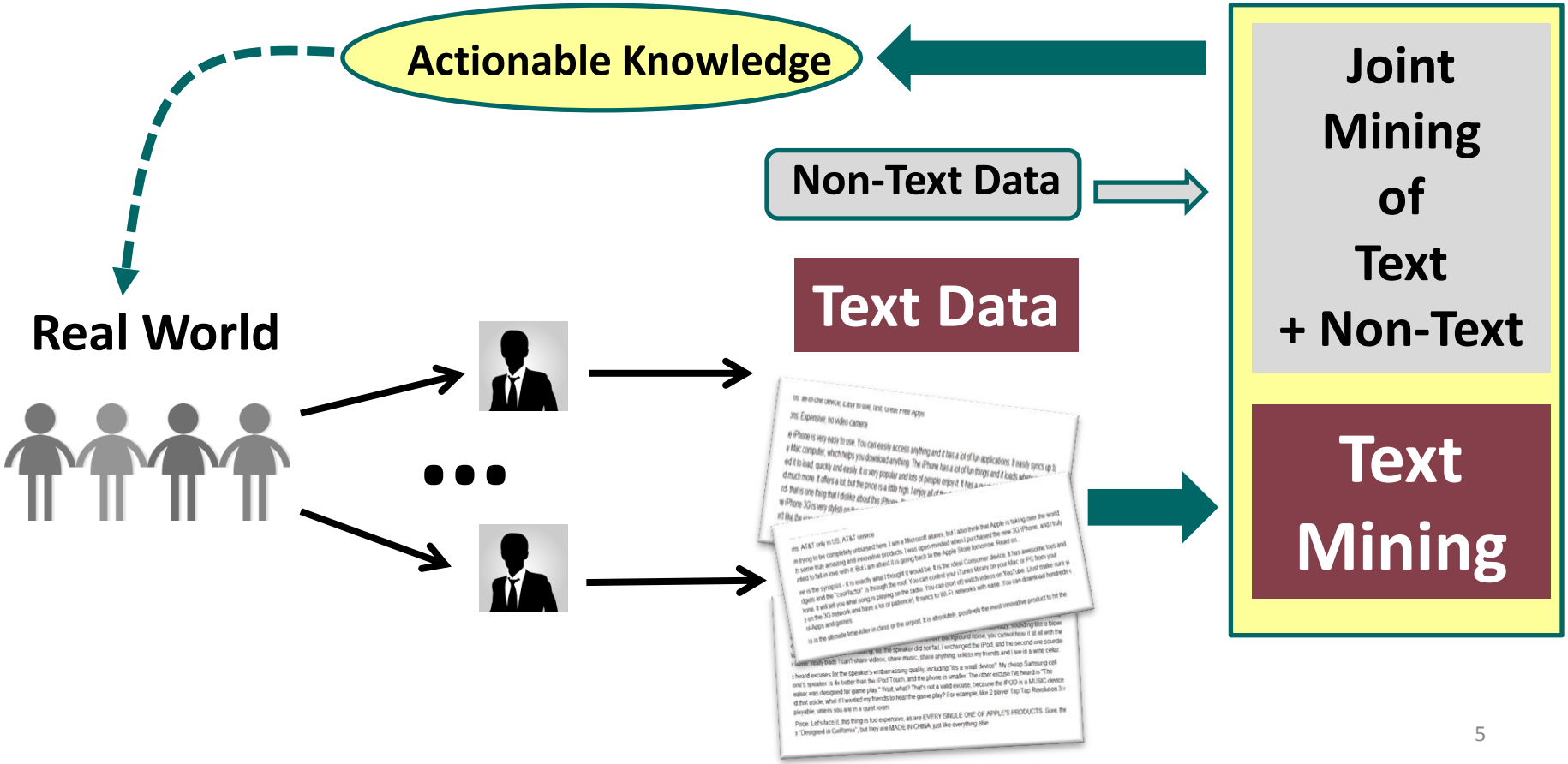
Express →

"Human Sensor"

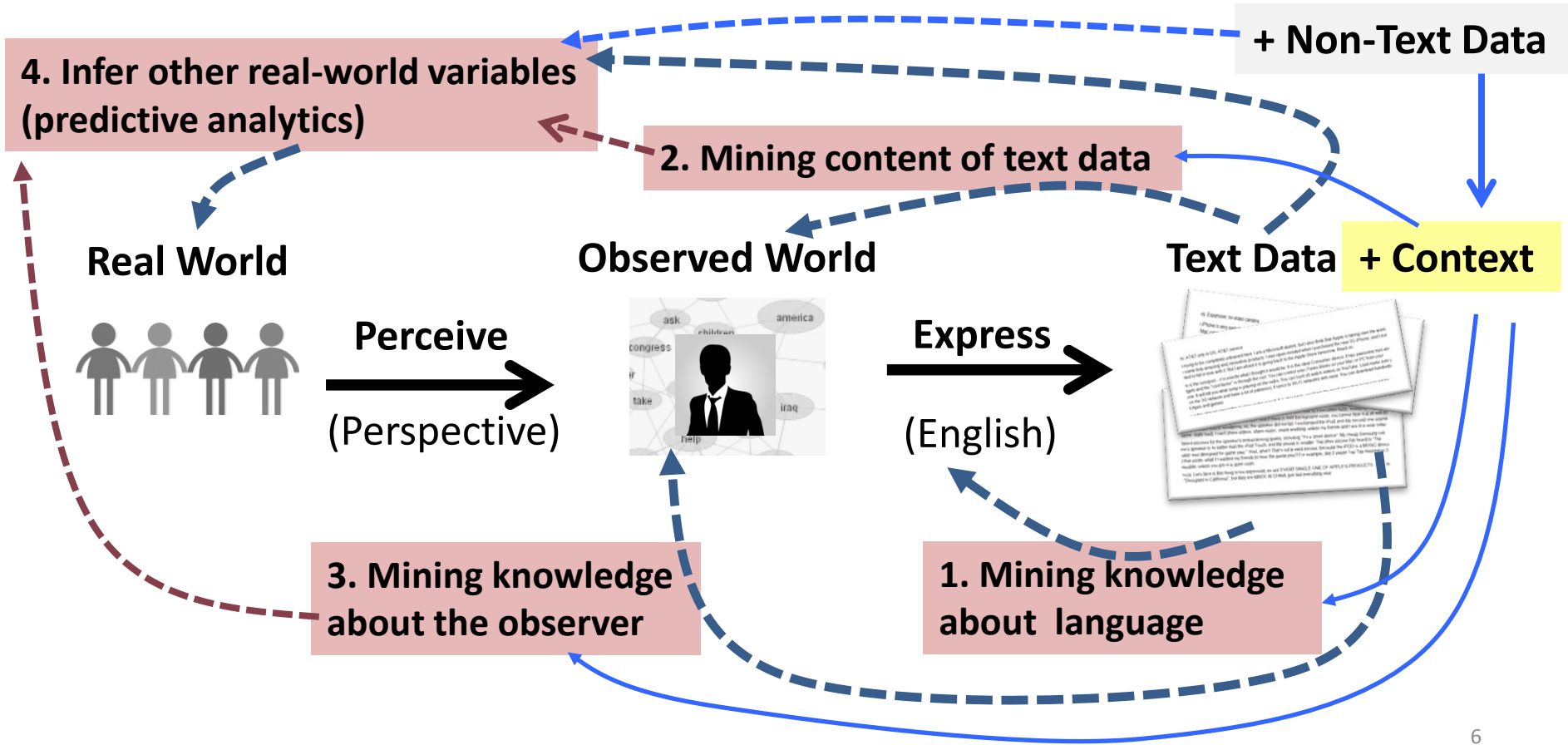
The General Problem of Data Mining



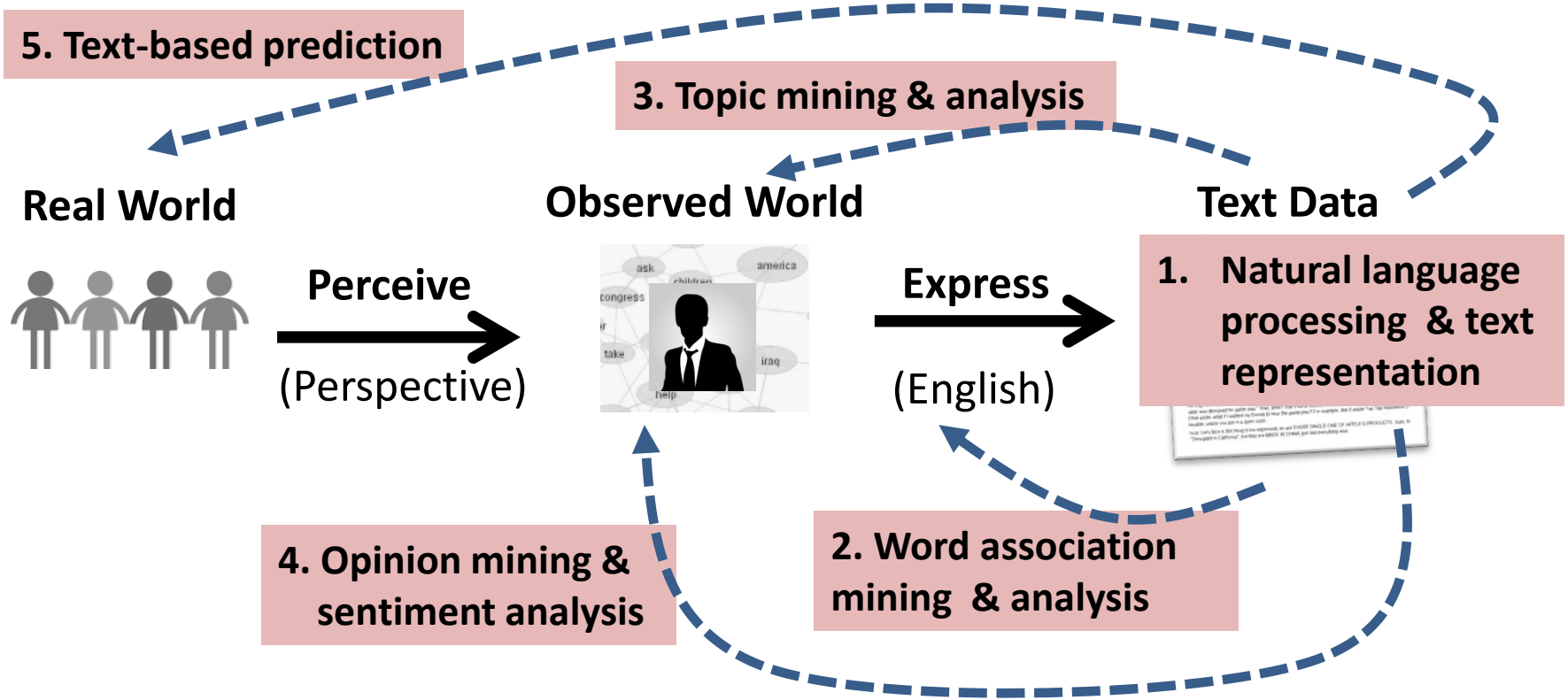
The Problem of Text Mining



Landscape of Text Mining and Analytics



Topics Covered in This Course



Overview Text Mining and Analytics

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

Text Mining and Analytics

- Text mining ≈ Text analytics
- Turn text data into **high-quality information** or **actionable knowledge**
  - **Minimizes human effort** (on consuming text data)
  - Supplies knowledge for **optimal decision making**
- Related to **text retrieval**, which is an essential component in any text mining system
  - Text retrieval can be a preprocessor for text mining
  - Text retrieval is needed for knowledge provenance

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

Real World —Sense→ **Sensor** —Report→ **Data**

Weather  ⇢ **Thermometer** ⇢ 3°C , 15°F, …

Locations  ⇢ **Geo Sensor** ⇢ 41°N and 120°W ….

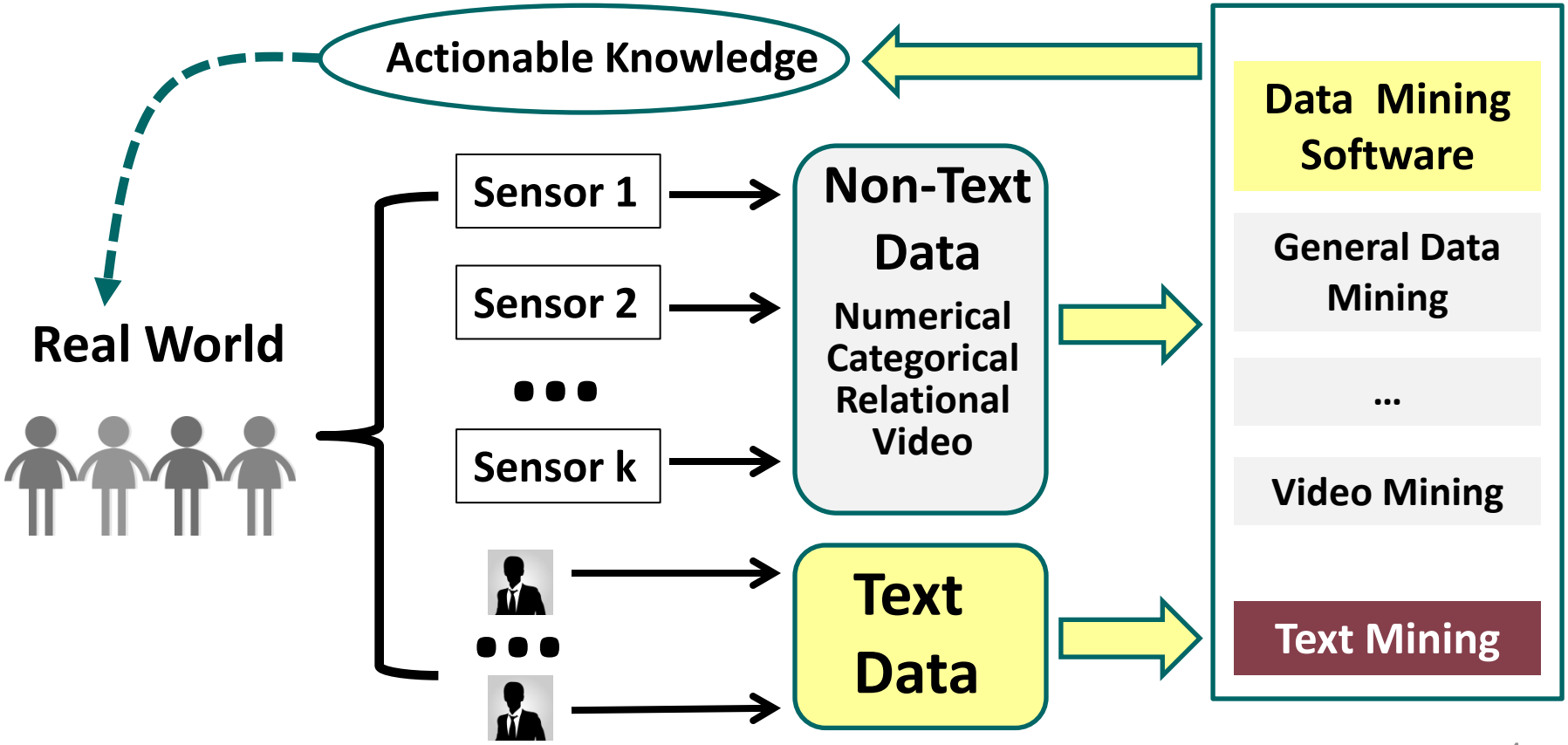
Networks  ⇢ **Network Sensor** ⇢ 01000100011100

Perceive →

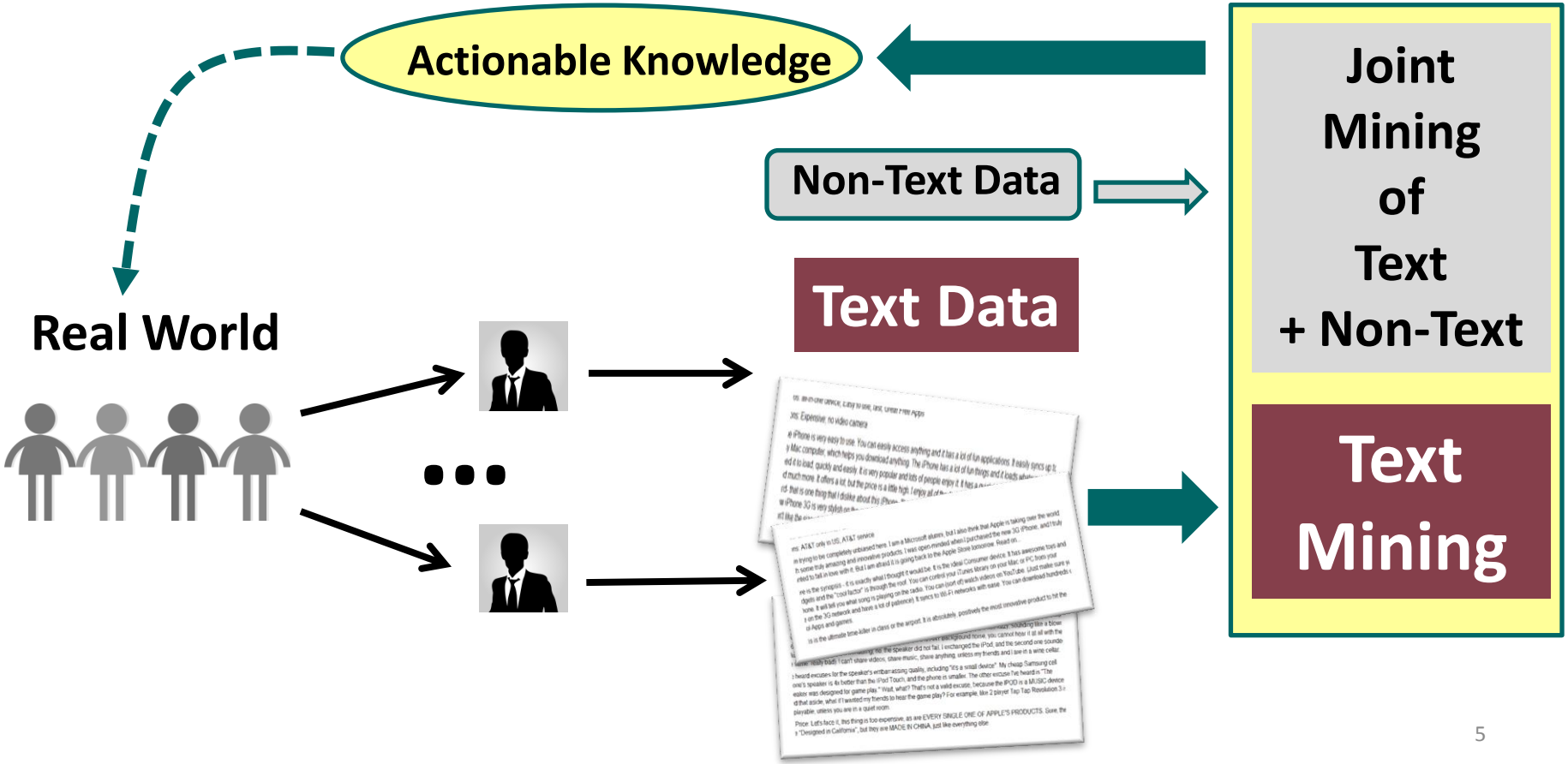
Express →

"Human Sensor"

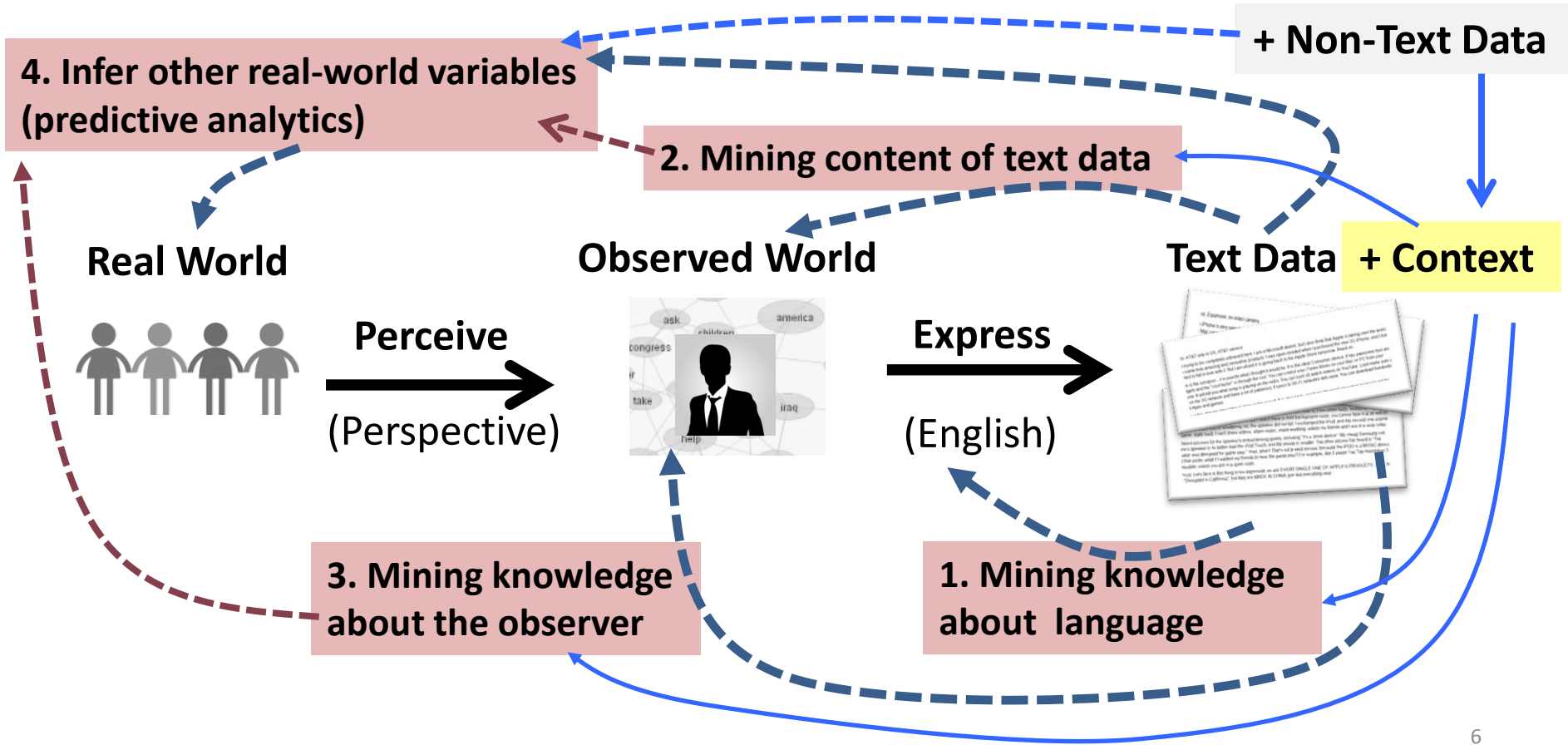
The General Problem of Data Mining



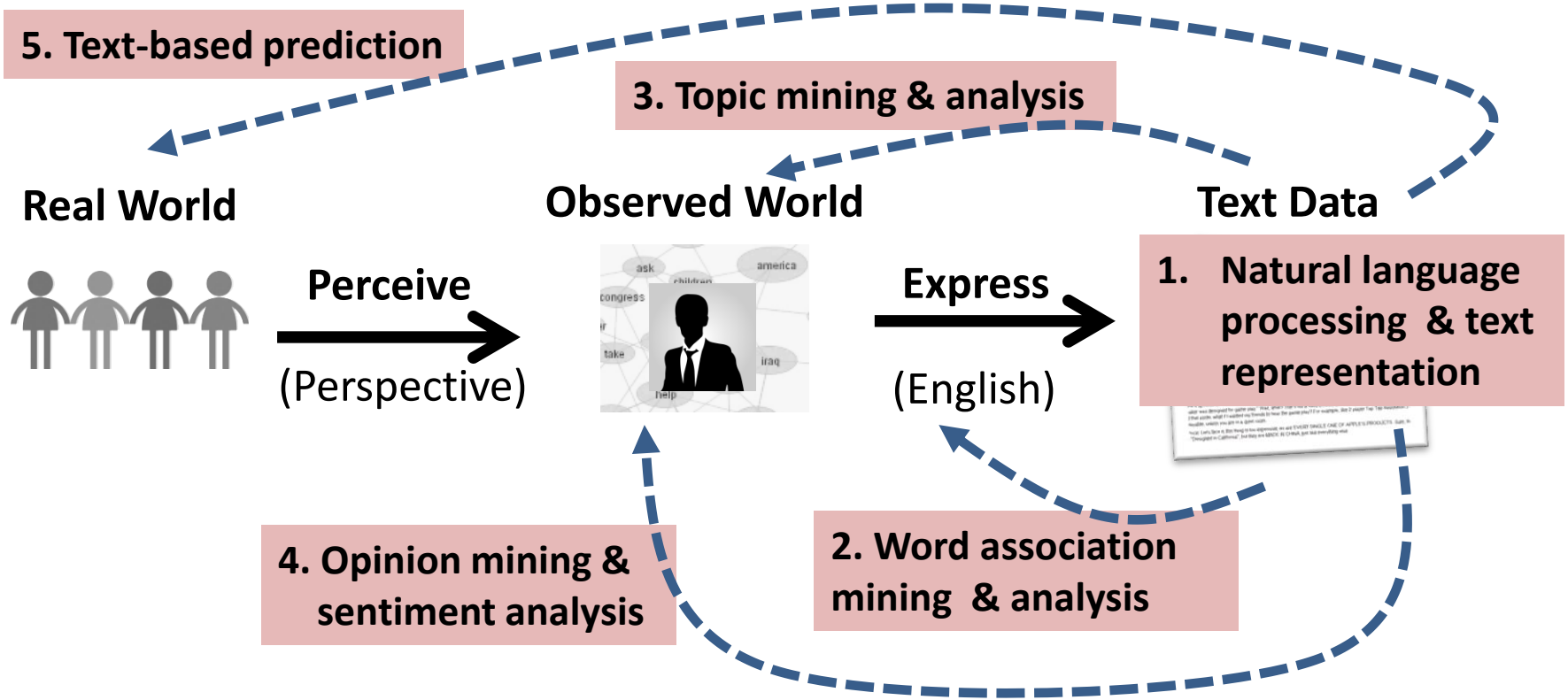
The Problem of Text Mining



Landscape of Text Mining and Analytics



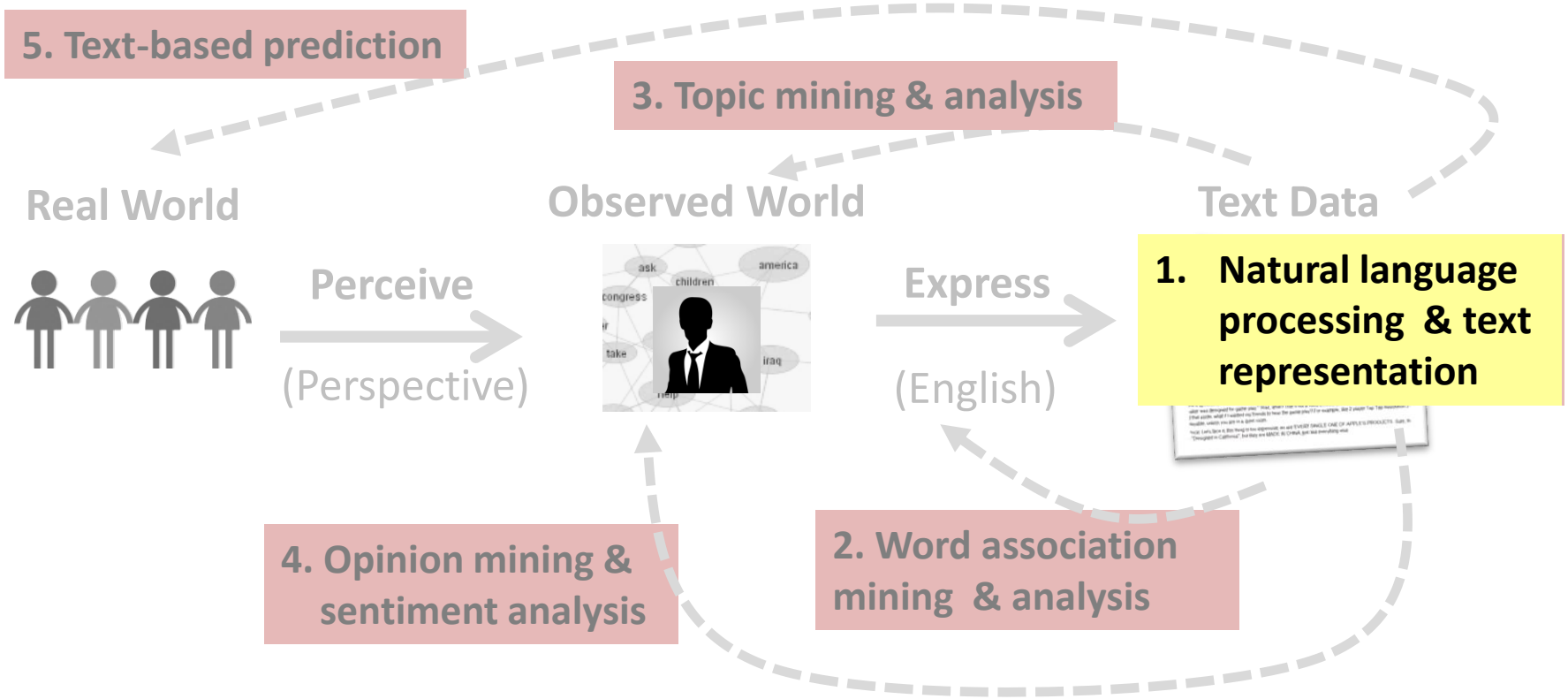
Topics Covered in This Course



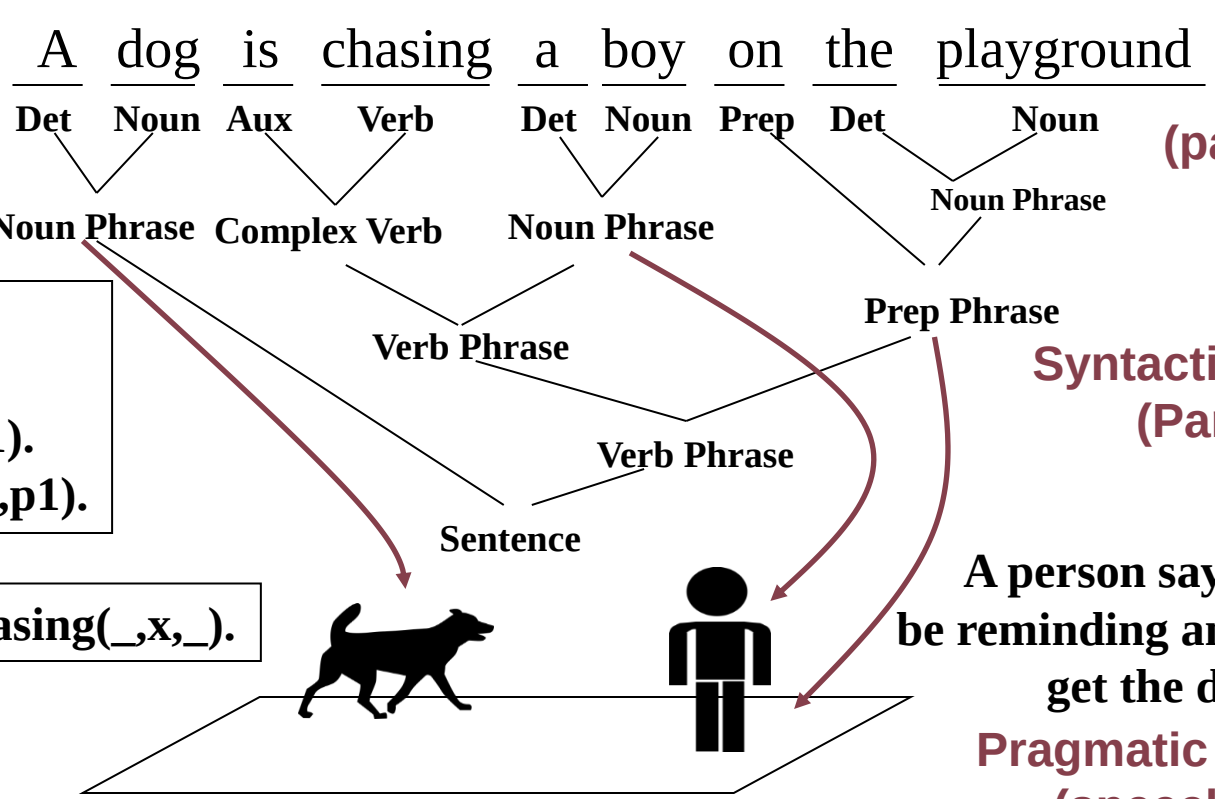
Natural Language Content Analysis

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

Natural Language Content Analysis



Basic Concepts in NLP



Lexical analysis (part-of-speech tagging)

Syntactic analysis (Parsing)

Semantic analysis

Dog(d1).
Boy(b1).
Playground(p1).
Chasing(d1,b1,p1).

+

Scared(x) if Chasing(_,x,_).

Scared(b1)
Inference

A person saying this may be reminding another person to get the dog back.

Pragmatic analysis (speech act)

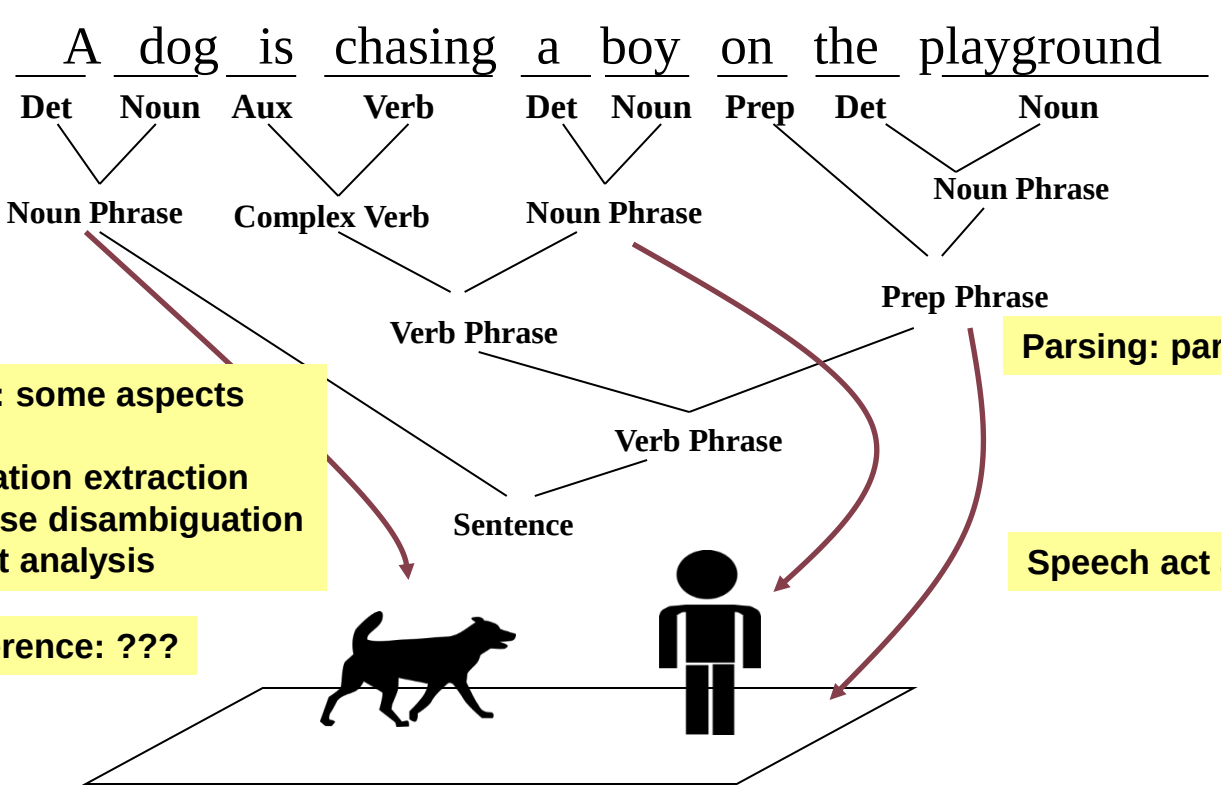
NLP Is Difficult!

- Natural language is designed to make human communication efficient. As a result,
  - we omit a lot of *common sense* knowledge, which we assume the hearer/reader possesses.
  - we keep a lot of ambiguities, which we assume the hearer/reader knows how to resolve.
- This makes EVERY step in NLP hard
  - Ambiguity is a *killer*!
  - Common sense reasoning is pre-required.

Examples of Challenges

- Word-level ambiguity:
  - "design" can be a noun or a verb (ambiguous POS)
  - "root" has multiple meanings (ambiguous sense)
- Syntactic ambiguity:
  - "natural language processing" (modification)
  - "A man saw a boy *with a telescope*." (PP Attachment)
- Anaphora resolution: "John persuaded Bill to buy a TV for *himself*." (himself = John or Bill?)
- Presupposition: "He has quit smoking" implies that he smoked before.

The State of the Art



POS Tagging: 97%

Parsing: partial >90%(?)

Semantics: some aspects

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

Speech act analysis: ???

Inference: ???

What We Can't Do

- 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
  - Will we ever be able to precisely define the meaning of "own" in "John owns a restaurant"?

Robust and general NLP tends to be *shallow* while *deep* understanding doesn't scale up.

Summary

- NLP is the foundation for text mining
- Computers are far from being able to understand natural language
  - Deep NLP requires common sense knowledge and inferences, thus only working for very limited domains
  - Shallow NLP based on statistical methods can be done in large scale and is thus more broadly applicable
- In practice: statistical NLP as the basis, while humans provide help as needed

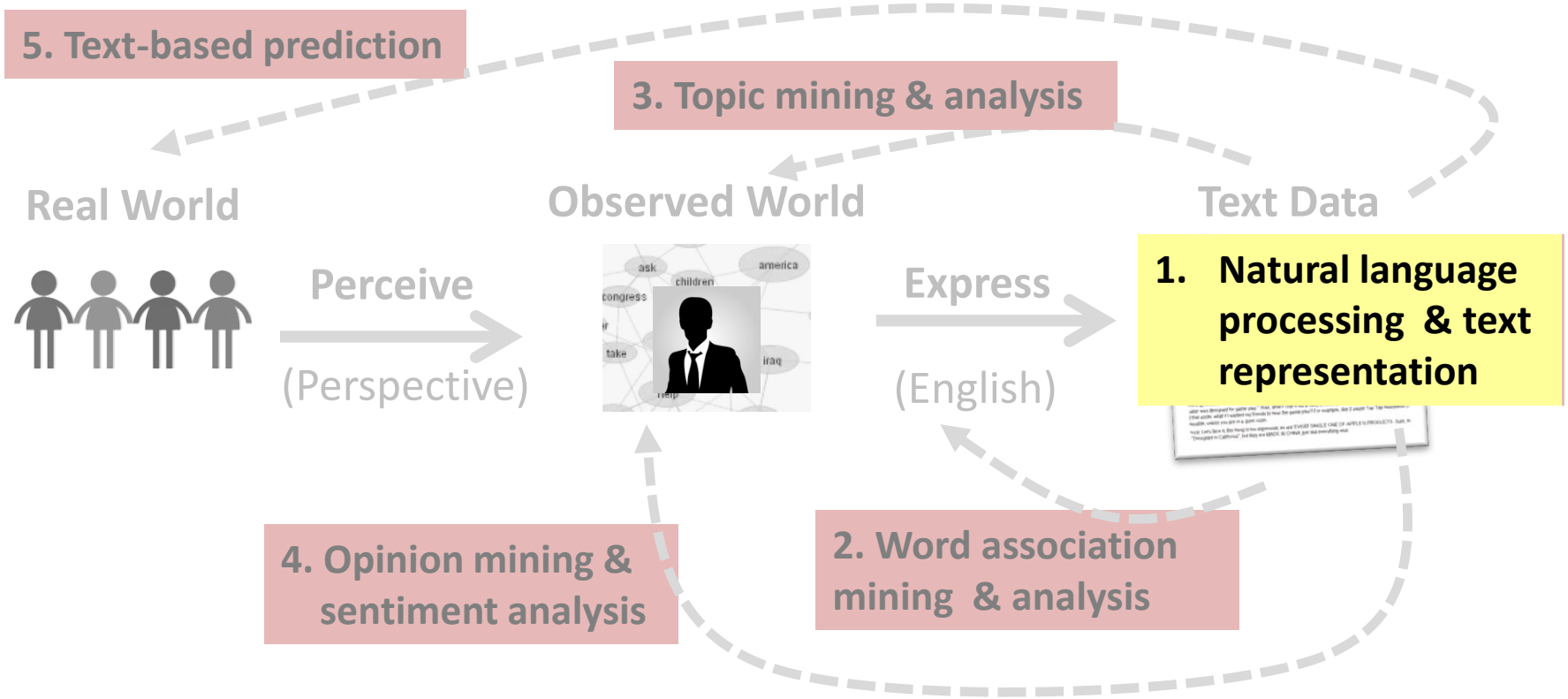
Additional Reading

Manning, Chris and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. Cambridge: MIT Press, 1999.

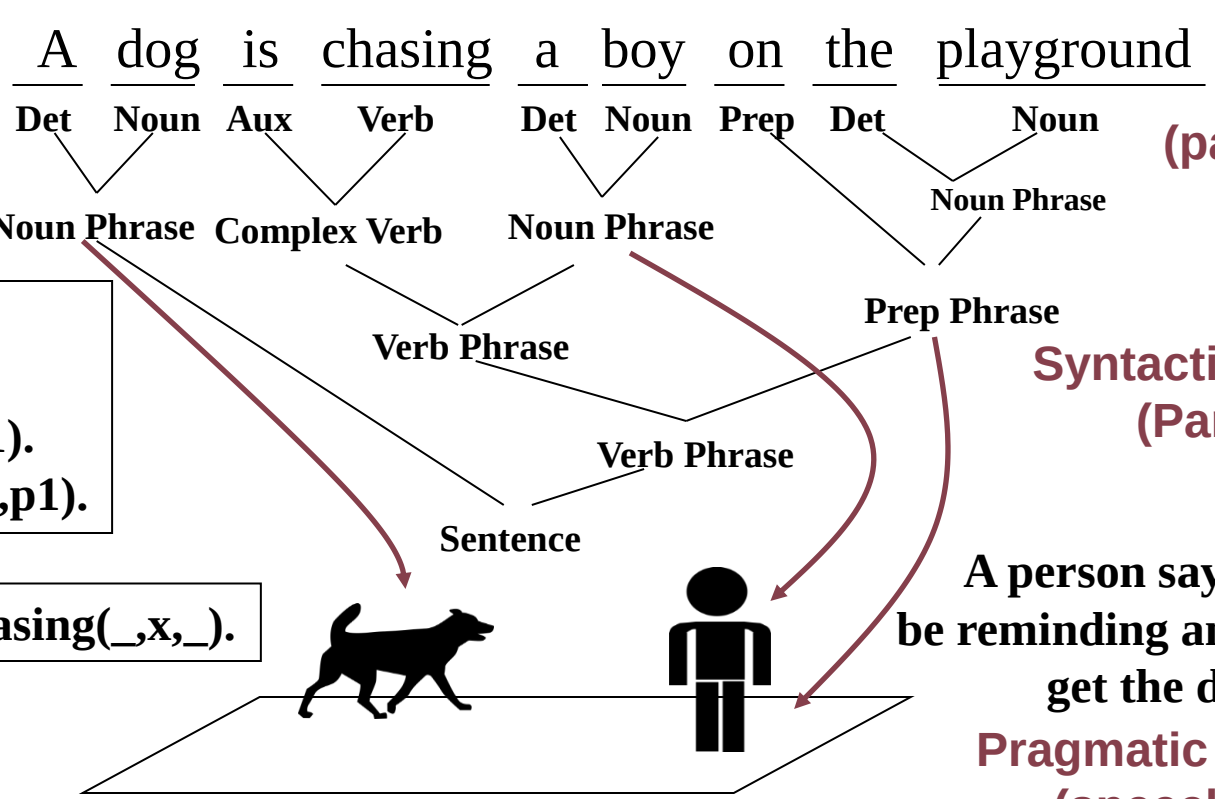
Natural Language Content Analysis

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

Natural Language Content Analysis



Basic Concepts in NLP



Lexical analysis (part-of-speech tagging)

Syntactic analysis (Parsing)

Semantic analysis

Dog(d1).
Boy(b1).
Playground(p1).
Chasing(d1,b1,p1).

+

Scared(x) if Chasing(_,x,_).

Scared(b1)
Inference

A person saying this may be reminding another person to get the dog back.

Pragmatic analysis (speech act)

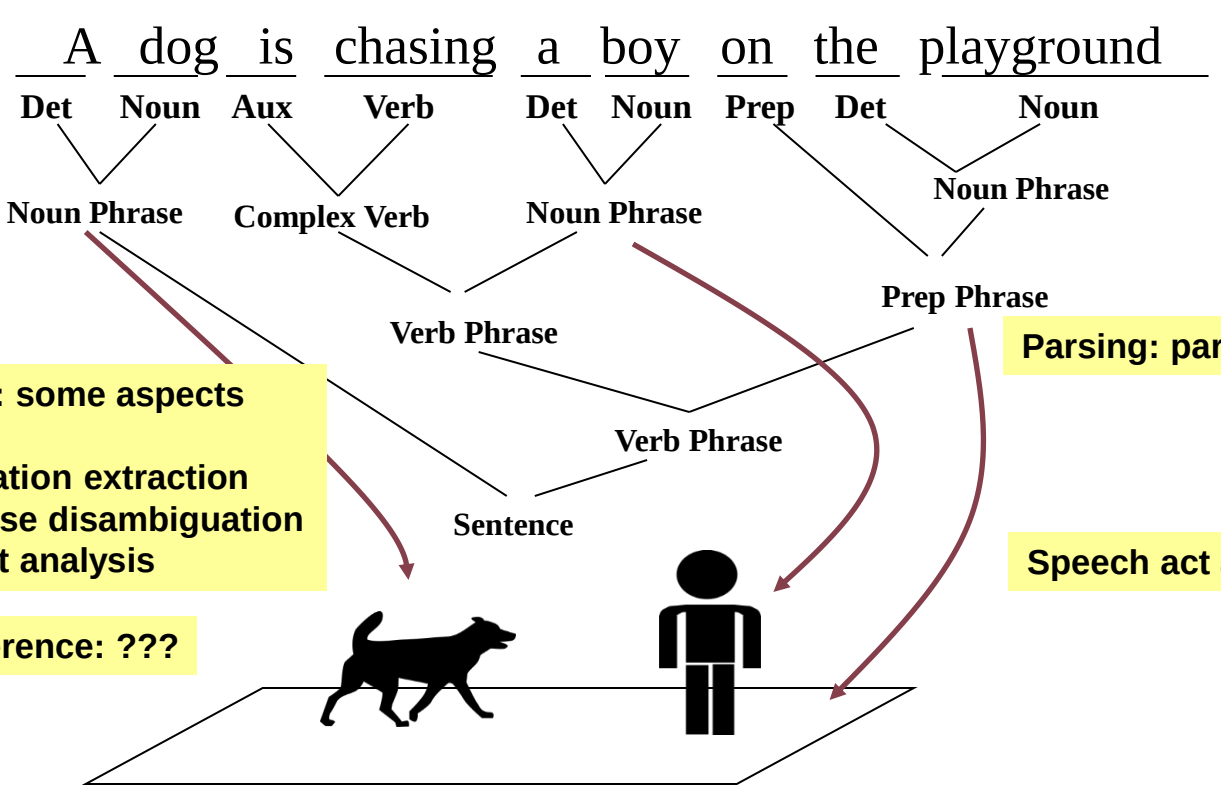
NLP Is Difficult!

- Natural language is designed to make human communication efficient. As a result,
  - we omit a lot of *common sense* knowledge, which we assume the hearer/reader possesses.
  - we keep a lot of ambiguities, which we assume the hearer/reader knows how to resolve.
- This makes EVERY step in NLP hard
  - Ambiguity is a *killer*!
  - Common sense reasoning is pre-required.

Examples of Challenges

- Word-level ambiguity:
  - "design" can be a noun or a verb (ambiguous POS)
  - "root" has multiple meanings (ambiguous sense)
- Syntactic ambiguity:
  - "natural language processing" (modification)
  - "A man saw a boy *with a telescope*." (PP Attachment)
- Anaphora resolution: "John persuaded Bill to buy a TV for *himself*." (himself = John or Bill?)
- Presupposition: "He has quit smoking" implies that he smoked before.

The State of the Art



POS Tagging: 97%

Parsing: partial >90%(?)

Semantics: some aspects

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

Speech act analysis: ???

Inference: ???

What We Can't Do

- 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
  - Will we ever be able to precisely define the meaning of "own" in "John owns a restaurant"?

Robust and general NLP tends to be *shallow* while *deep* understanding doesn't scale up.

Summary

- NLP is the foundation for text mining
- Computers are far from being able to understand natural language
  - Deep NLP requires common sense knowledge and inferences, thus only working for very limited domains
  - Shallow NLP based on statistical methods can be done in large scale and is thus more broadly applicable
- In practice: statistical NLP as the basis, while humans provide help as needed

Additional Reading

Manning, Chris and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. Cambridge: MIT Press, 1999.

Text Representation

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

Text Representation

5. Text-based prediction

3. Topic mining & analysis

Real World

Observed World

Text Data

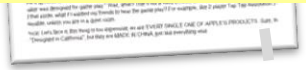
Perceive

(Perspective)

Express

(English)

1. **Natural language processing & text representation**



4. Opinion mining & sentiment analysis

2. Word association mining & analysis

A dog is chasing a boy on the playground

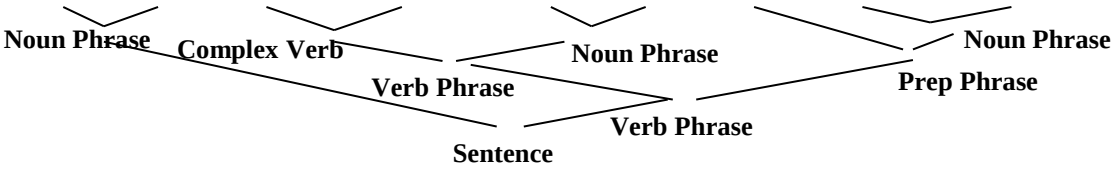
String of characters

A dog is chasing a boy on the playground

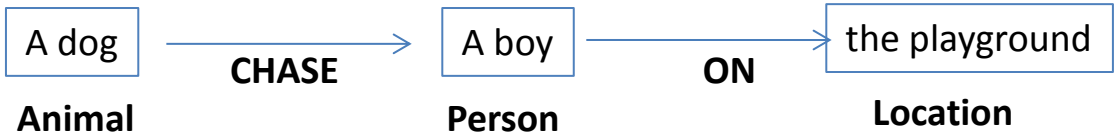
Sequence of words

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

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

Summary

- Text representation determines what kind of mining algorithms can be applied
- **Multiple ways** of representing text are possible
  - string, words, syntactic structures, entity-relation graphs, predicates…
  - can/should be **combined** in real applications
- This course focuses on **word-based representation**
  - **General and robust**: applicable to any natural language
  - **No/little manual effort**
  - **"Surprisingly" powerful** for many applications (not all!)
  - **Can be combined** with more sophisticated representations

Text Representation

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

Text Representation

5. Text-based prediction

3. Topic mining & analysis

Real World

Observed World

Text Data

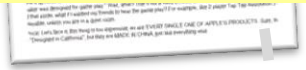
Perceive

(Perspective)

Express

(English)

1. **Natural language processing & text representation**



4. Opinion mining & sentiment analysis

2. Word association mining & analysis

A dog is chasing a boy on the playground

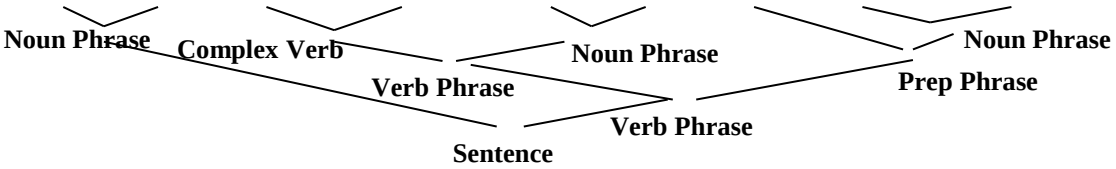
String of characters

A dog is chasing a boy on the playground

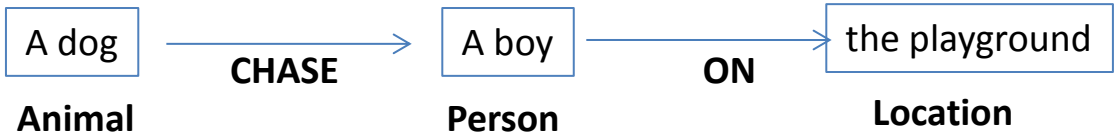
Sequence of words

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

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

Summary

- Text representation determines what kind of mining algorithms can be applied
- **Multiple ways** of representing text are possible
  - string, words, syntactic structures, entity-relation graphs, predicates…
  - can/should be **combined** in real applications
- This course focuses on **word-based representation**
  - **General and robust**: applicable to any natural language
  - **No/little manual effort**
  - **"Surprisingly" powerful** for many applications (not all!)
  - **Can be combined** with more sophisticated representations

Word Association Mining and Analysis

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

Word Association Mining & Analysis

5. Text-based prediction

3. Topic mining and analysis

Real World

Observed World

Text Data

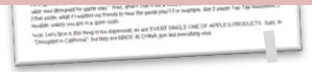
Perceive

(Perspective)

Express

(English)

1. Natural language processing and text representation



4. Opinion mining and sentiment analysis

2. Word association mining and analysis

Outline

- What is a word association?
- Why mine word associations?
- How to mine word associations?

Basic Word Relations: Paradigmatic vs. Syntagmatic

- Paradigmatic: A & B have paradigmatic relation if they can be substituted for each other (i.e., A & B are in the same class)
  - E.g., “cat” and “dog”; “Monday” and “Tuesday”
- Syntagmatic: A & B have syntagmatic relation if they can be combined with each other (i.e., A & B are related semantically)
  - E.g., “cat” and “sit”; “car” and “drive”
- These two basic and complementary relations can be generalized to describe relations of any items in a language

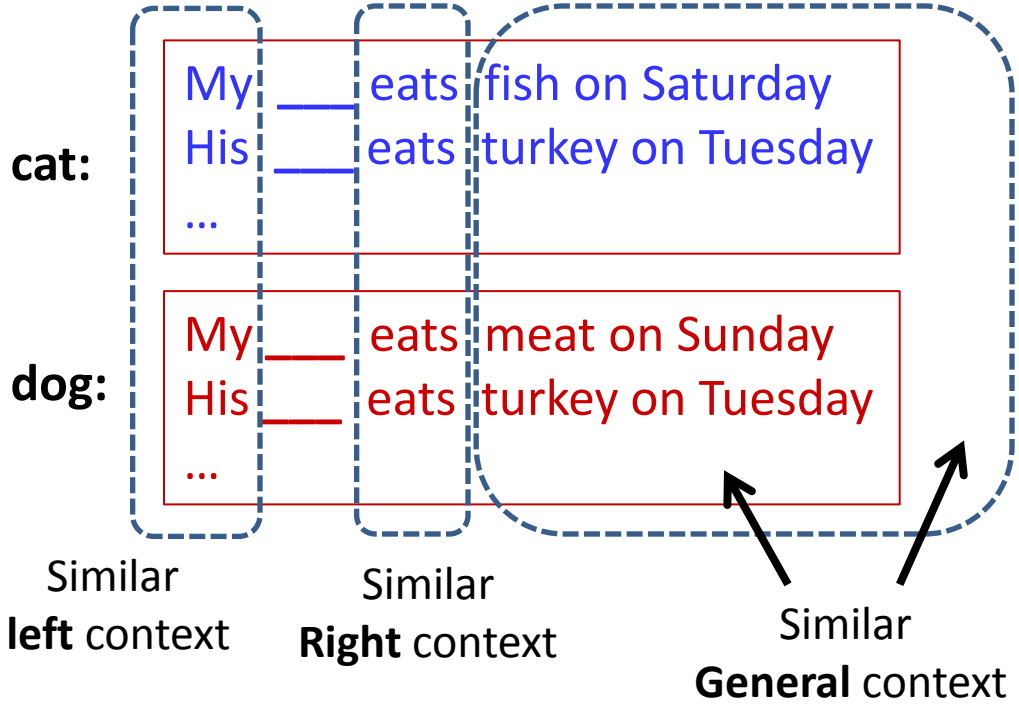
Why Mine Word Associations?

- They are useful for improving accuracy of many NLP tasks
  - POS tagging, parsing, entity recognition, acronym expansion
  - Grammar learning
- They are directly useful for many applications in text retrieval and mining
  - Text retrieval (e.g., use word associations to suggest a variation of a query)
  - Automatic construction of topic map for browsing: words as nodes and associations as edges
  - Compare and summarize opinions (e.g., what words are most strongly associated with “battery” in positive and negative reviews about iPhone 6, respectively?)

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
…

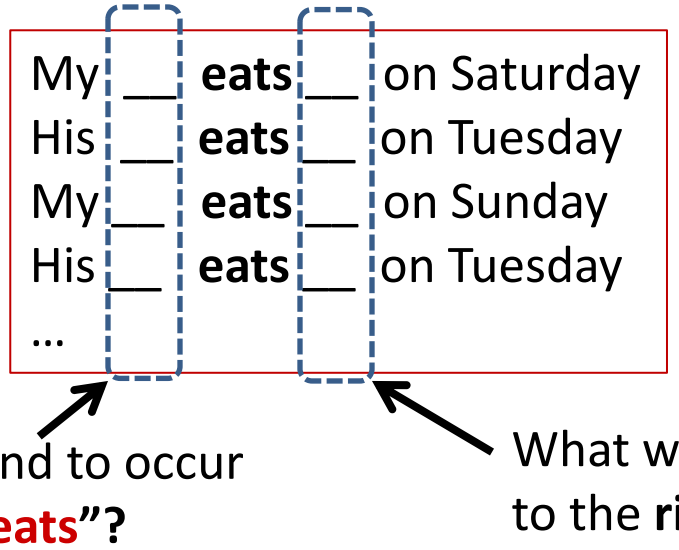


How similar are context ("cat") and context ("dog")?
How similar are context ("cat") and context ("computer")?

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
…



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: General Ideas

- **Paradigmatic**
  - Represent each word by its context
  - Compute context similarity
  - Words with **high context similarity** likely have paradigmatic relation
- **Syntagmatic**
  - Count how many times two words occur together in a context (e.g., sentence or paragraph)
  - Compare their co-occurrences with their individual occurrences
  - Words with **high co-occurrences but relatively low individual occurrences** likely have syntagmatic relation
- Paradigmatically related words tend to have syntagmatic relation with the same word ➔ **joint discovery** of the two relations
- These ideas can be implemented in many different ways!

Paradigmatic Relation Discovery

Parts 1-3

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

Paradigmatic Relation Discovery

5. Text-based prediction

3. Topic mining & analysis

Real World

Observed World

Text Data

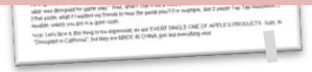
Perceive

(Perspective)

Express

(English)

1. Natural language processing and text representation

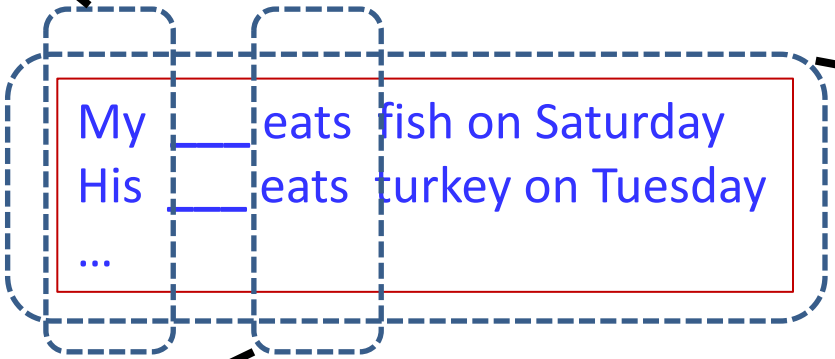


4. Opinion mining and sentiment analysis

2. Word association mining and analysis

Word Context as “Pseudo Document”

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



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”)) +

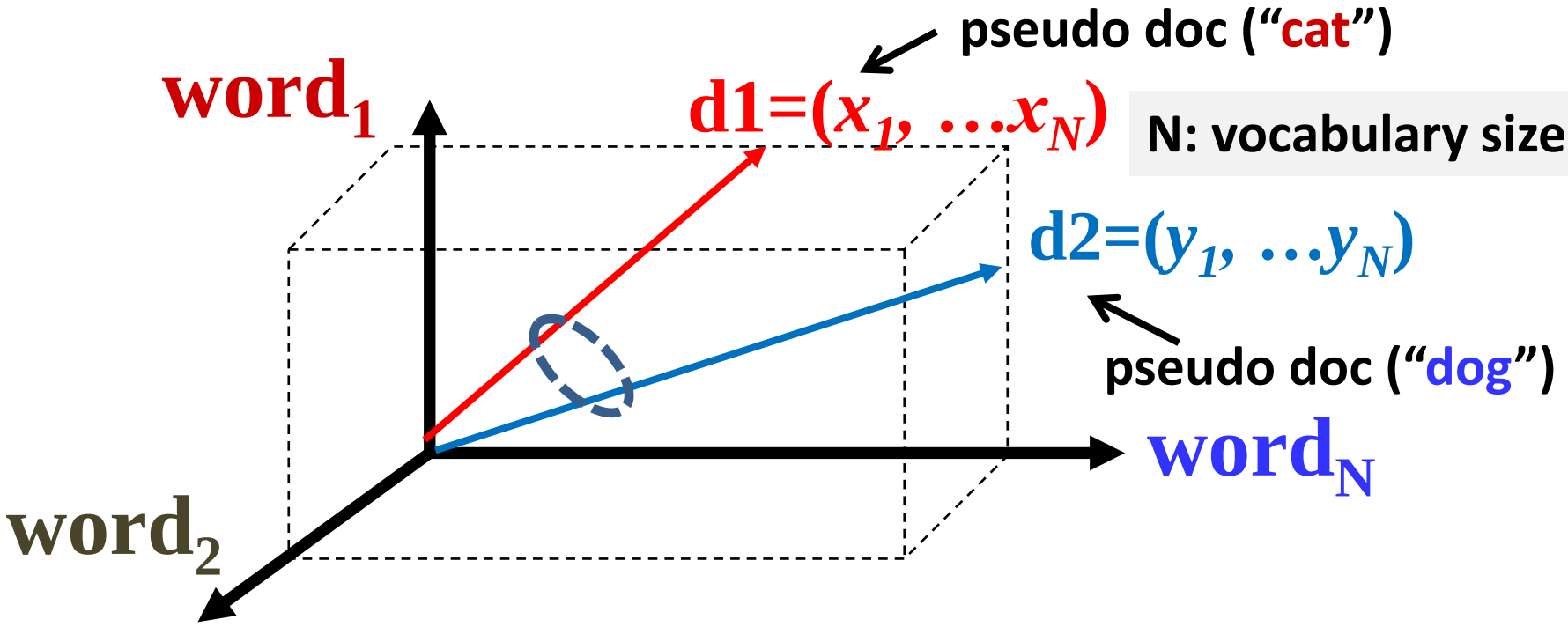
…

+ Sim(Window8(“cat”), Window8(“dog”))=?

High sim(word1, word2)

➔ word1 and word2 are **paradigmatically related**

Bag of Words ➔ Vector Space Model (VSM)

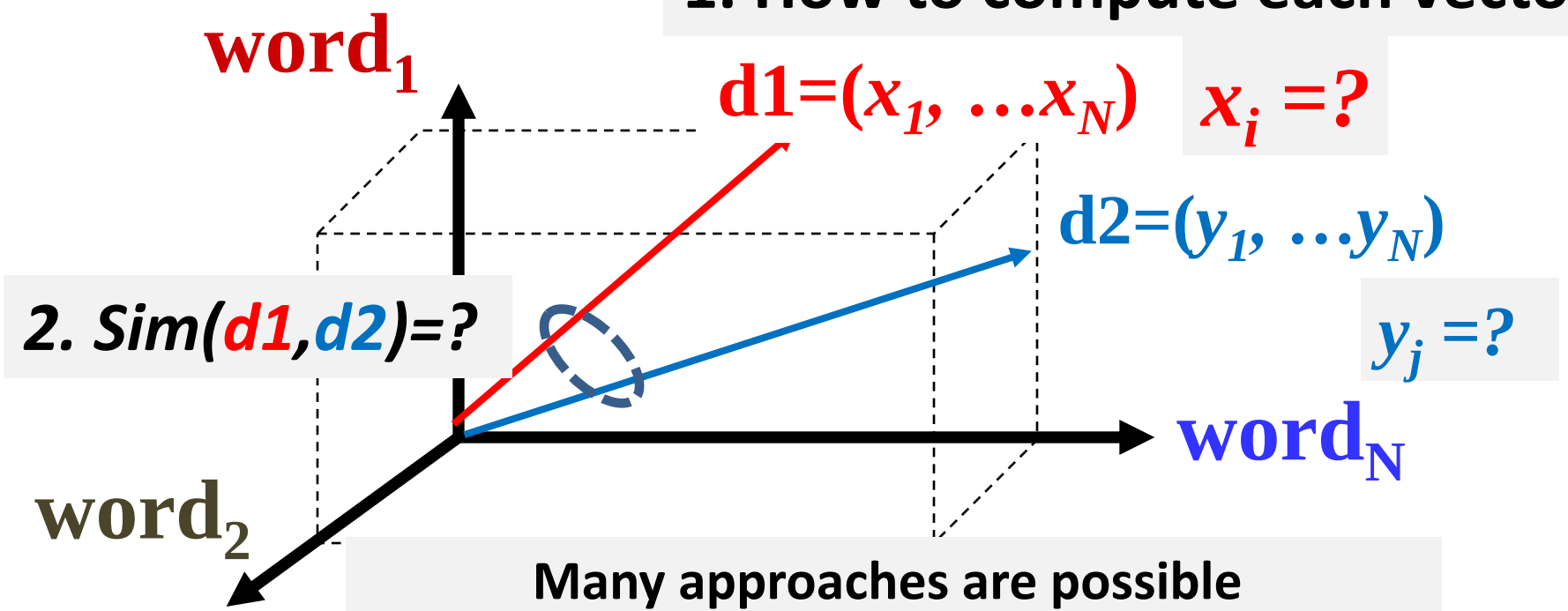


Terms: "eats" "ate" "is" "has" ….

Vector: (5, 3, 10, 3 ….)

VSM for Paradigmatic Relation Mining

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

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

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

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:
  - 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

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

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

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

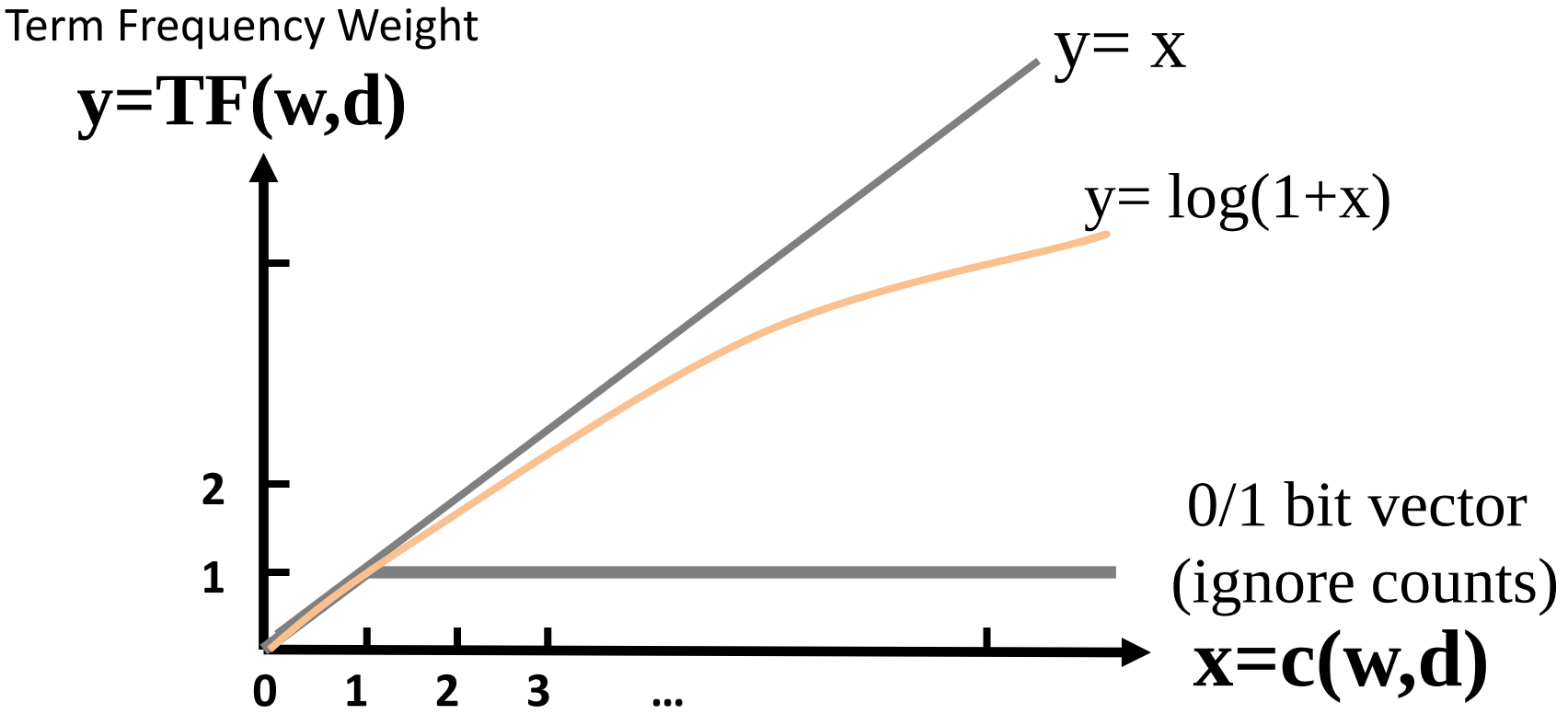
- It favors matching one frequent term very well over matching more distinct terms.

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

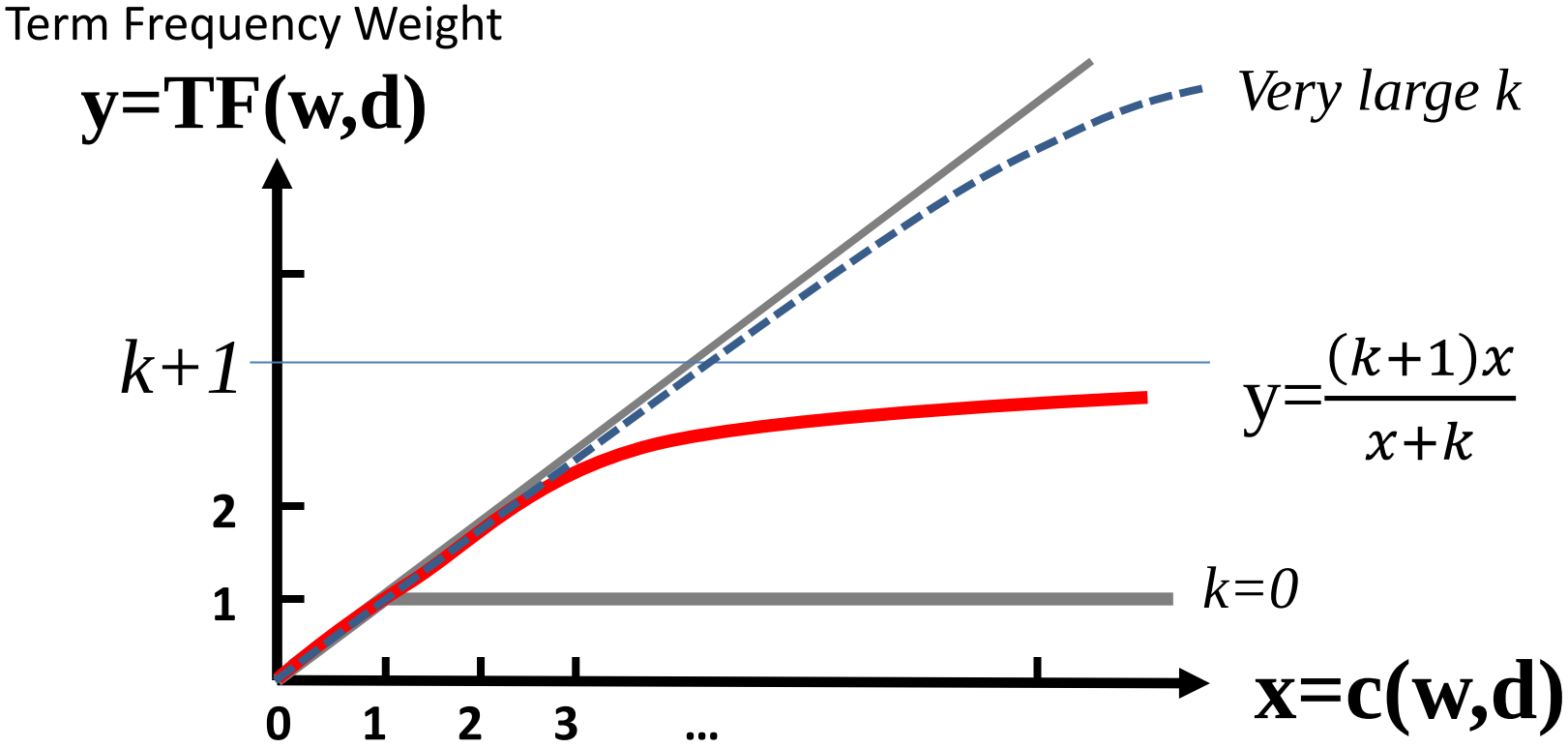
- It treats every word equally (overlap on “the” isn’t as so meaningful as overlap on “eats”).

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

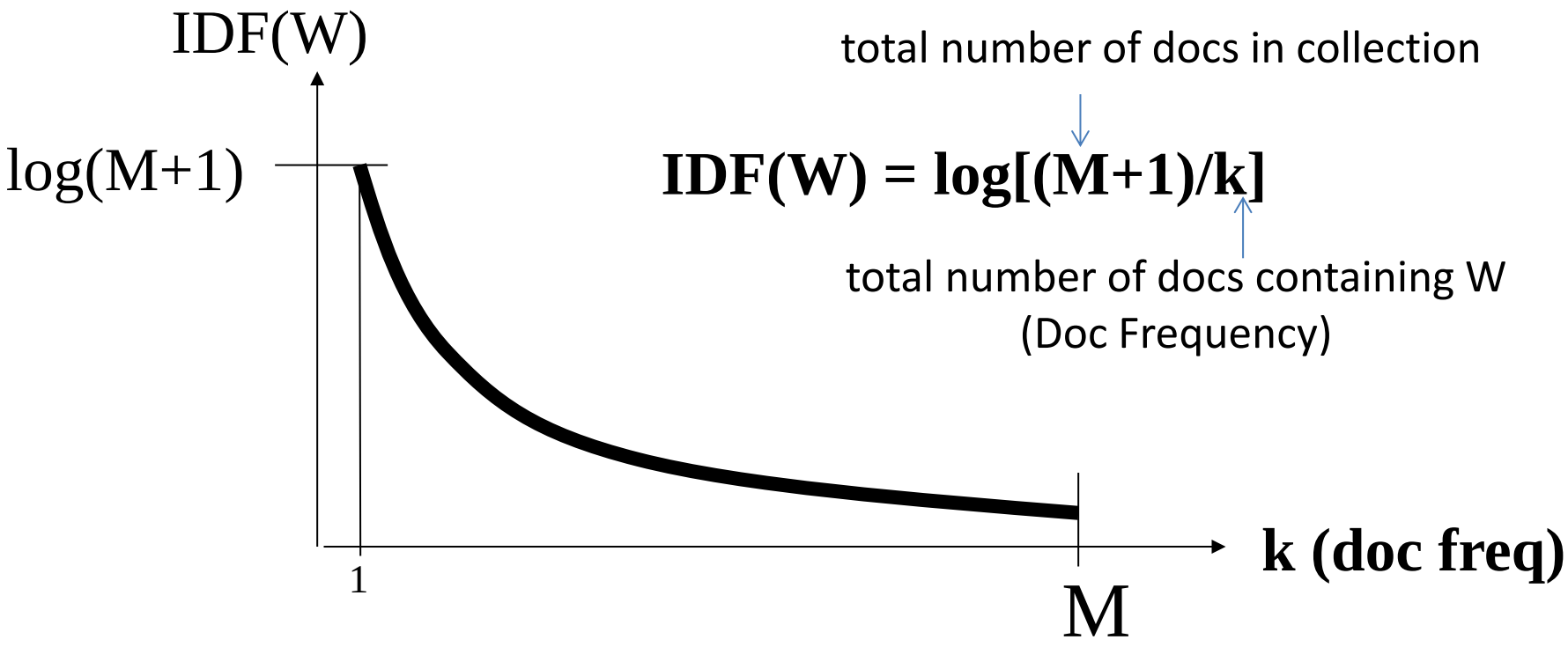
TF Transformation: c(w,d)➔TF(w,d)



TF Transformation: BM25 Transformation



IDF Weighting: Penalizing Popular Terms



total number of docs in collection

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

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

- 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”

Paradigmatic Relation Discovery

Parts 1-3

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

Paradigmatic Relation Discovery

5. Text-based prediction

3. Topic mining & analysis

Real World

Observed World

Text Data

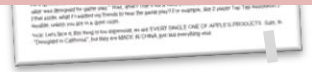
Perceive

(Perspective)

Express

(English)

1. Natural language processing and text representation

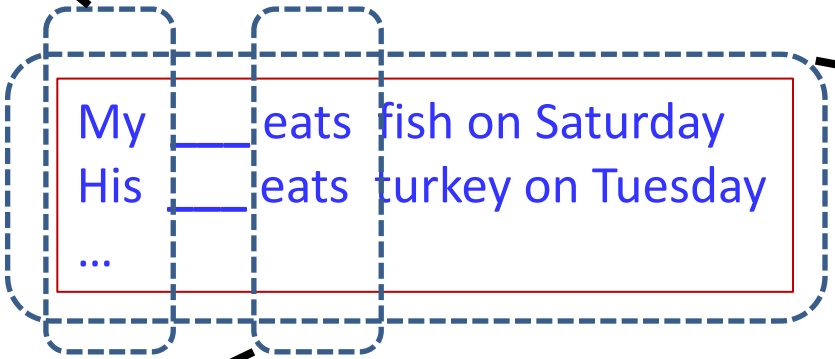


4. Opinion mining and sentiment analysis

2. Word association mining and analysis

Word Context as “Pseudo Document”

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



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”)) +

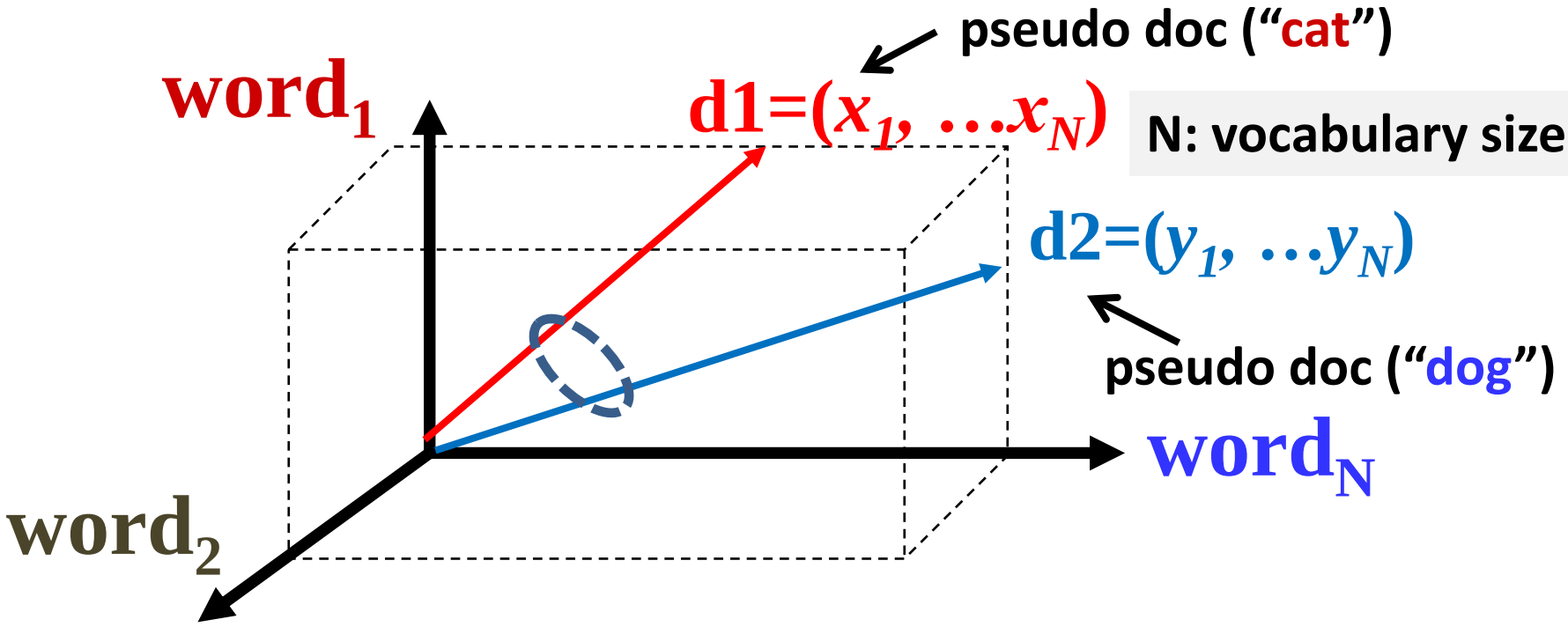
…

+ Sim(Window8(“cat”), Window8(“dog”))=?

High sim(word1, word2)

➔ word1 and word2 are **paradigmatically related**

Bag of Words ➔ Vector Space Model (VSM)

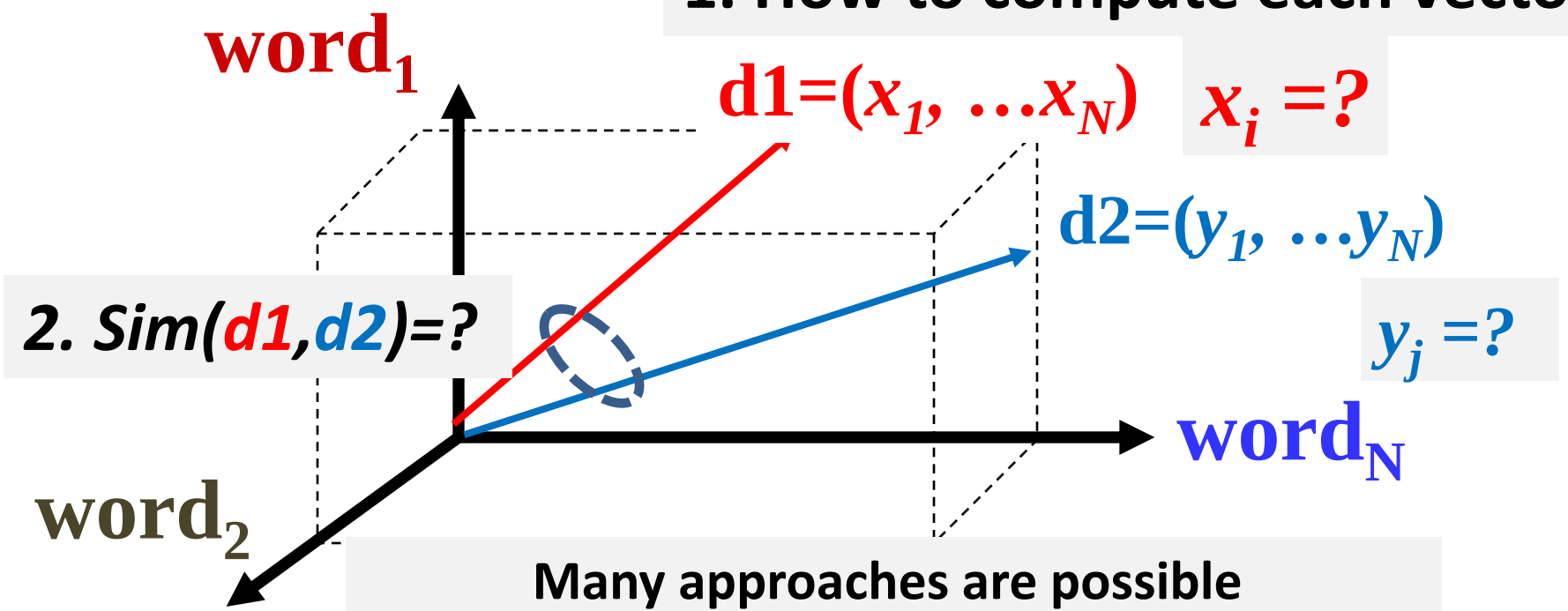


Terms: "eats" "ate" "is" "has" ….

Vector: (5, 3, 10, 3 ….)

VSM for Paradigmatic Relation Mining

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

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

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

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:
  - 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

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

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

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

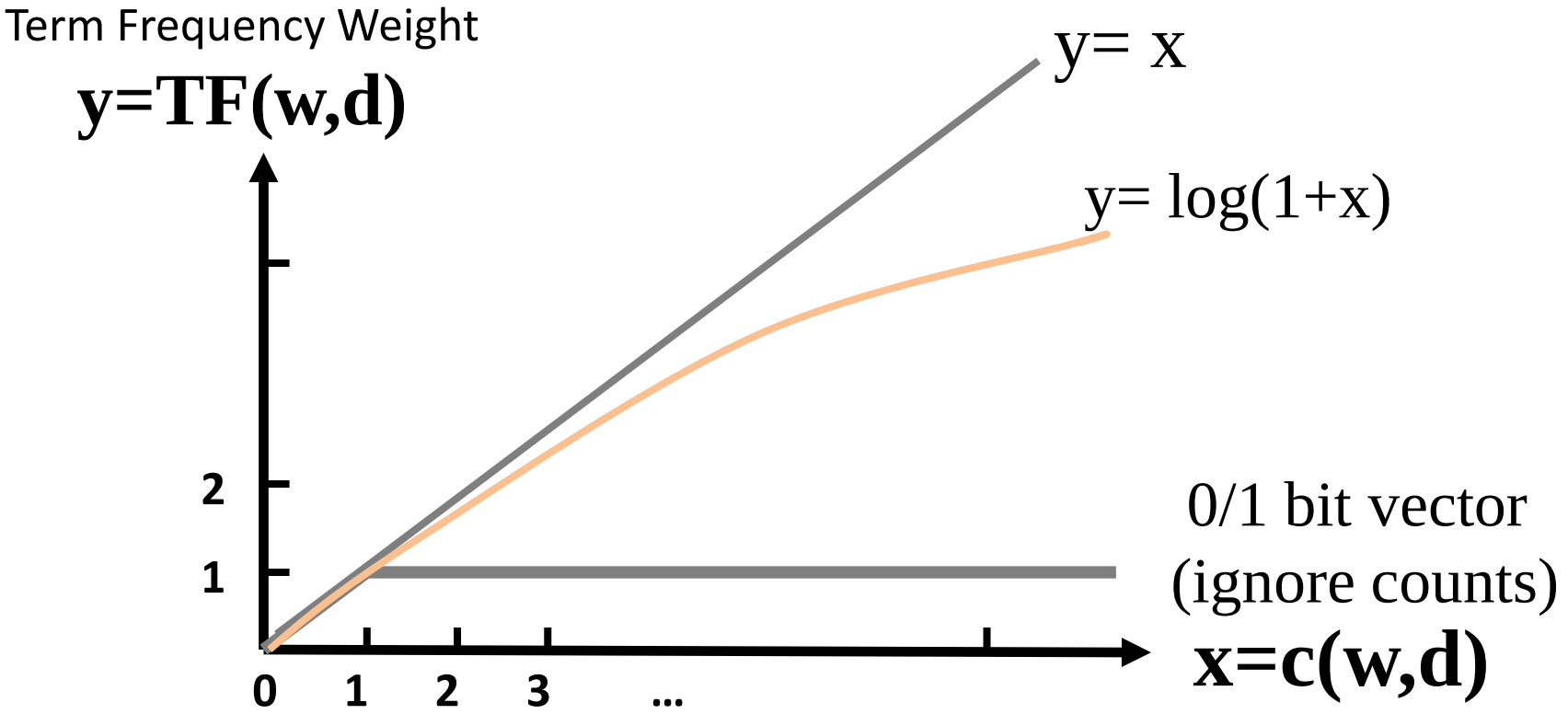
- It favors matching one frequent term very well over matching more distinct terms.

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

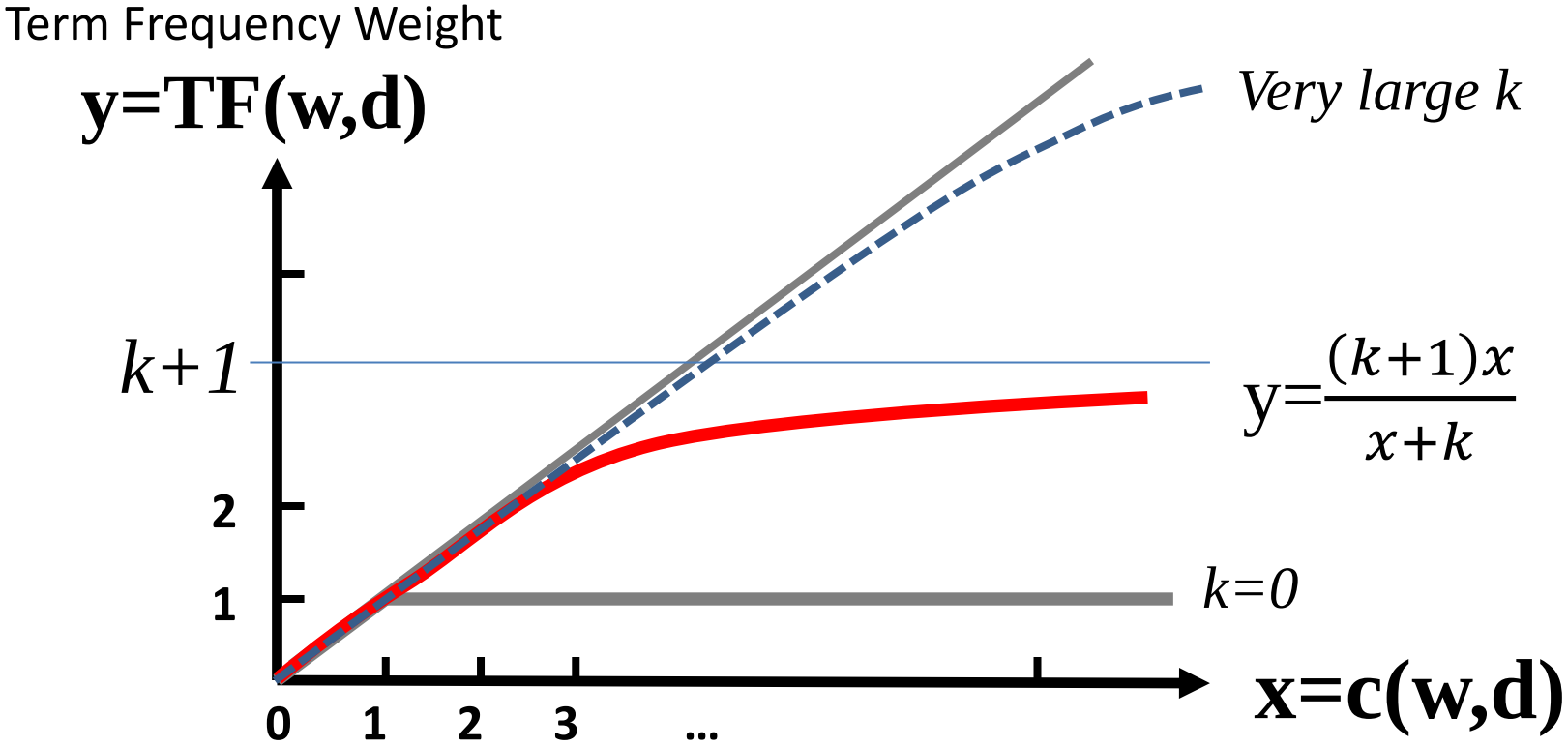
- It treats every word equally (overlap on “the” isn’t as so meaningful as overlap on “eats”).

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

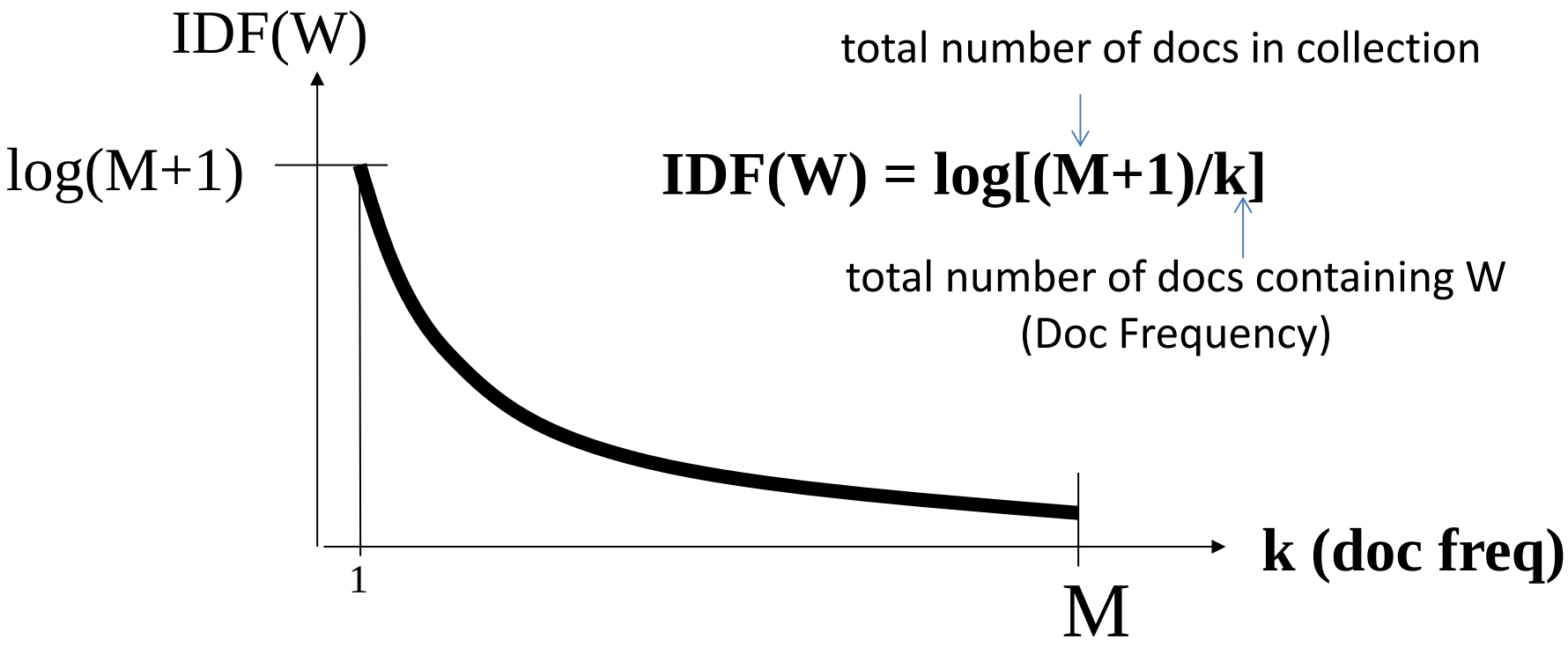
TF Transformation: c(w,d)➔TF(w,d)



TF Transformation: BM25 Transformation



IDF Weighting: Penalizing Popular Terms



total number of docs in collection

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

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

- 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”