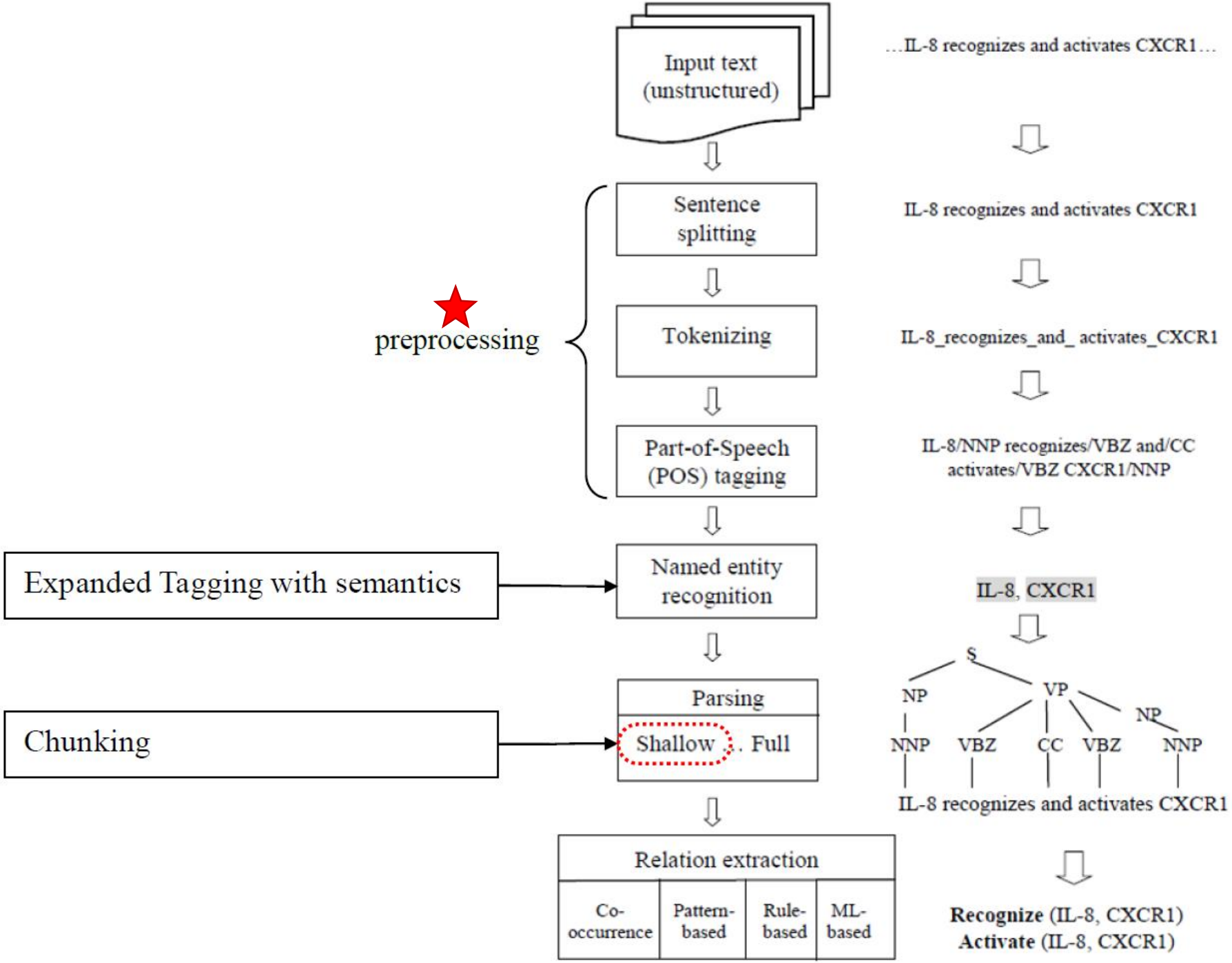
Preprocessing(2)


Input text (unstructured)
…IL-8 recognizes and activates CXCR1…
Sentence splitting
IL-8 recognizes and activates CXCR1
preprocessing
Tokenizing
IL-8_recognizes_and_ activates_CXCR1
Part-of-Speech (POS) tagging
IL-8/NNP recognizes/VBZ and/CC activates/VBZ CXCR1/NNP
Expanded Tagging with semantics
Named entity recognition
IL-8, CXCR1
Parsing
Chunking
Shallow
Full
S
NP
VP
NP
NNP
VBZ
CC
VBZ
NNP
IL-8 recognizes and activates CXCR1
Relation extraction
Co-occurrence
Pattern-based
Rule-based
ML-based
Recognize (IL-8, CXCR1)
Activate (IL-8, CXCR1)

n-gram

n-gram

WHAT **Contiguous sequence** of ***n*** items from a given sample of text or speech

phonemes, ***syllables***, *letters*, ***words*** or *base pairs* according to the application

WHY **Predicts** a next letter

Given a sequence of letters, what is the likelihood of the next letter?

$x_{i-(n-1)}, \dots, x_{i-1}$ $\quad$ x_i

$$\boldsymbol{P(x_i | x_{i-(n-1)}, \dots, x_{i-1})}$$

WPM

Word Piece Model

하나의 단어를 내부 단어(Subword Unit)들로 분리하는 단어 분리 모델

Google's Neural Machine Translation System:
Bridging the Gap between Human and Machine Translation
Wo at el, 2016

System	Combined test set
Bing Translate	23.48
Google Translate	22.67
Word Model	18.19
Word-piece Model	26.58

BPE

Byte Pair Encoding (Digram Coding)

Simple form of data compression

The **most common pair** of consecutive bytes of data is replaced with a byte

Philip Gage, 1994

Neural Machine Translation of Rare Words with Subword Units
Sennrich at el, 2015

example

- learning
  - word:freq : {low:5, lowest:2, newer:6, wider:3}
  - marge & count
    1. 'r' '</w>' : 9 → marge'r</w>'
    2. 'e' 'r</w>' : 9 →marge'er</w>'
    3. 'l' 'o' : 7 →marge'lo'
    4. 'lo' 'w' : 7 →marge'low'

→ OOV : 'lower' segmented 'low er</w>'

Empirical Law

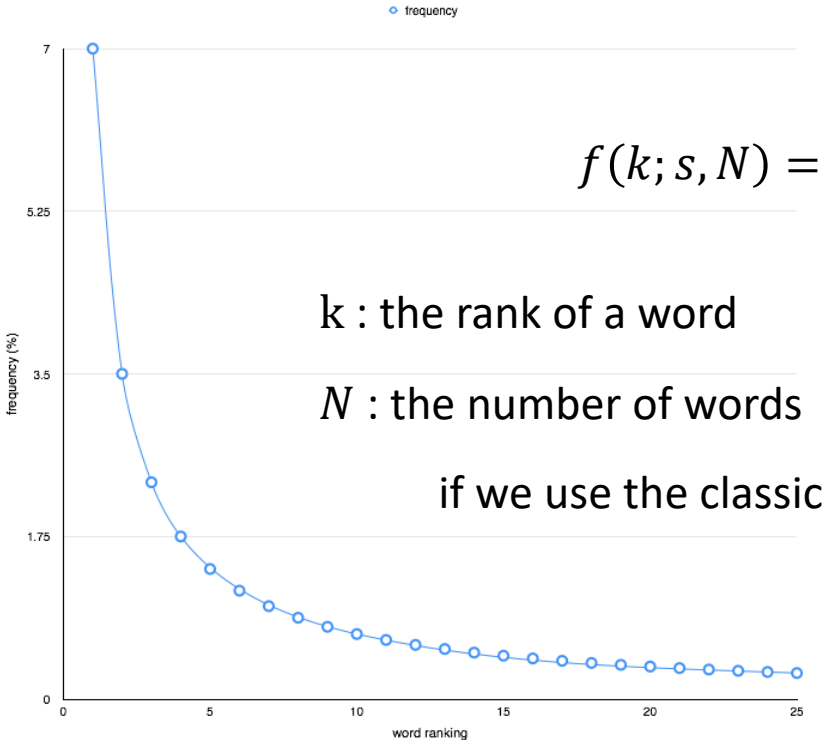
Zipf's Law

WHAT **An empirical law**

frequency of word is inversely proportional to its rank in frequency table

most frequent word will occur approximately twice

as often as the second most frequent word

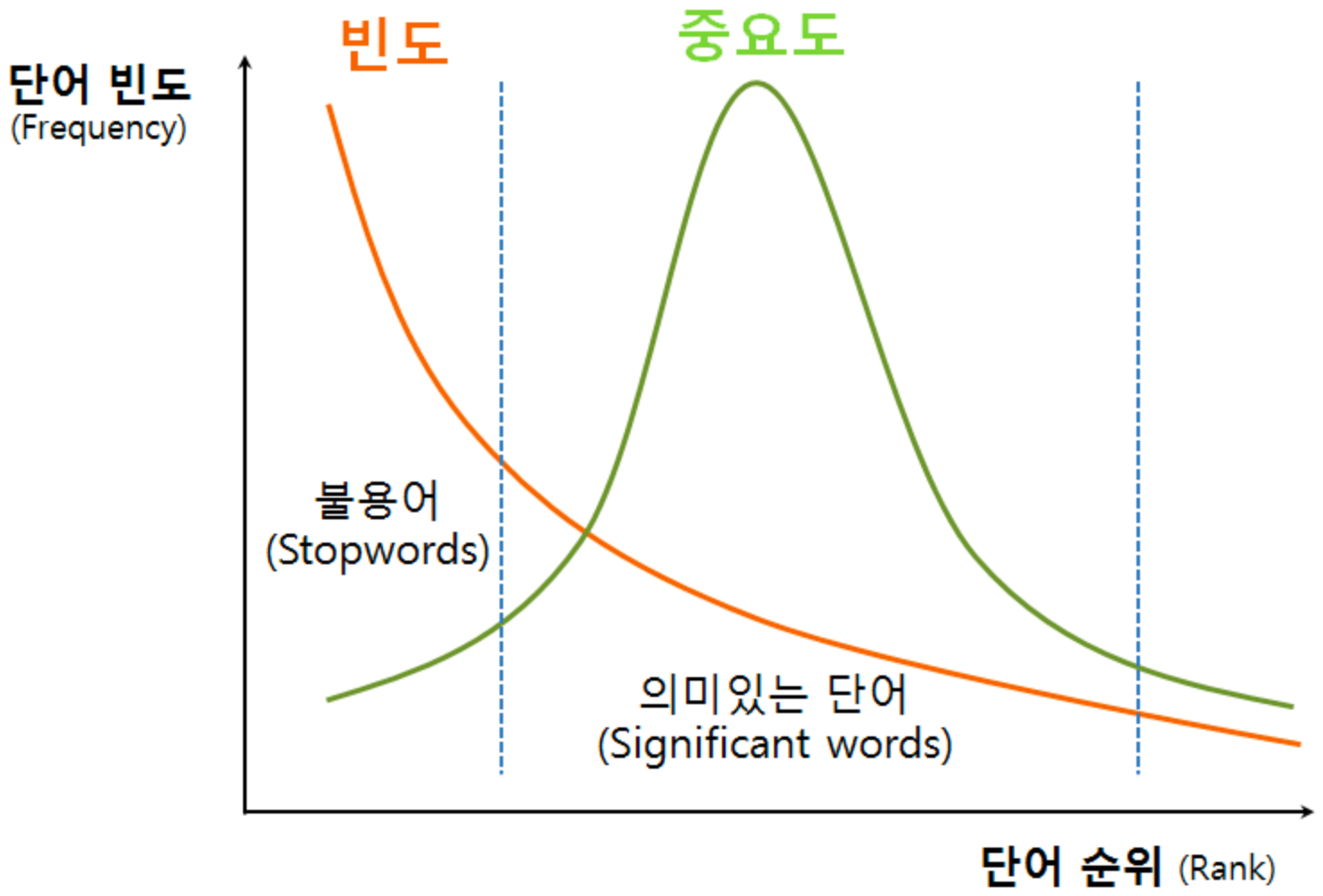


$$f(k;s,N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)} \Leftrightarrow \frac{1}{k^s H_{n,s}}$$

k : the rank of a word

N : the number of words

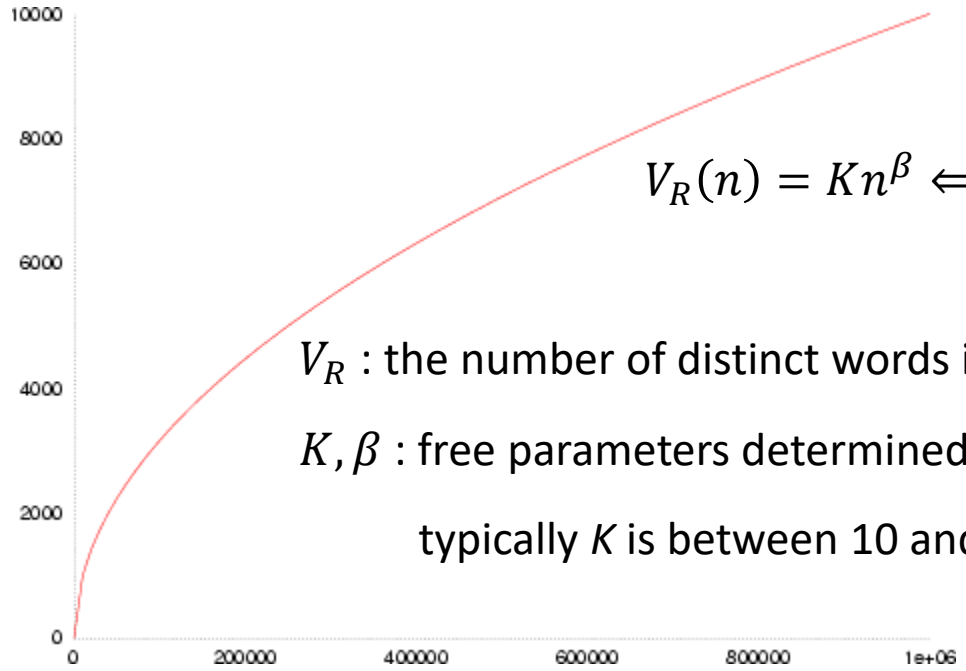
if we use the classic version of Zipf's law, the exponent s is 1


빈도
중요도
단어 빈도
(Frequency)
불용어
(Stopwords)
의미있는 단어
(Significant words)
단어 순위 (Rank)

Heaps' Law

WHAT **An empirical law**

# of distinct words in a document as a function of the document length



$$V_R(n) = Kn^{\beta} \Leftrightarrow M = kT^b$$

V_R : the number of distinct words in an instance text of size *n*

K, β : free parameters determined empirically

typically *K* is between 10 and 100, and β is between 0.4 and 0.6