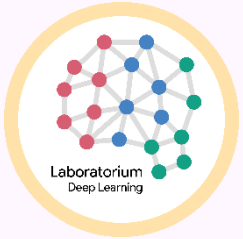
NLP (Natural language processing)

Dr. Andreas Hadiyono, ST, MMSI

Machine Learning Course



23/02/2019

NLP

NLP

- "Natural" languages
  - English, Mandarin, French, Swahili, Arabic, Nahuatl, ...
  - NOT Java, C++, Perl, ...
- Tujuan Utama: Natural human-to-computer communication
- Sub-field of Artificial Intelligence, but very interdisciplinary
  - Computer science, human-computer interaction (HCI), linguistics, cognitive psychology, speech signal processing (EE), ...

Tujuan NLP

- Understand language analysis & generation
- Communication
- Language is a window to the mind
- Data is in linguistic form
- Data can be in Structured (table form), Semi structured (XML form), Unstructured (sentence form).

Language Processing

- *Level 1* – Speech sound (*Phonetics & Phonology*)
- *Level 2* – Words & their forms (*Morphology, Lexicon*)
- *Level 3* – Structure of sentences (*Syntax, Parsing*)
- *Level 4* – Meaning of sentences (*Semantics*)
- *Level 5* – Meaning in context & for a purpose (*Pragmatics*)
- *Level 6* – Connected sentence processing in a larger body of text (*Discourse*)

Language Processing

- Morphology: What is a word?
- **奧林匹克運動會**（希臘語：**Ολυμπιακοί Αγώνες**，簡稱**奧運會**或**奧運**）是國際奧林匹克委員會主辦的包含多種體育運動項目的國際性運動會，**每四年舉行一次。**
- بيوتهك = “to her houses”

- Lexicography: What does each word mean?
  - He plays bass guitar.
  - That bass was delicious!

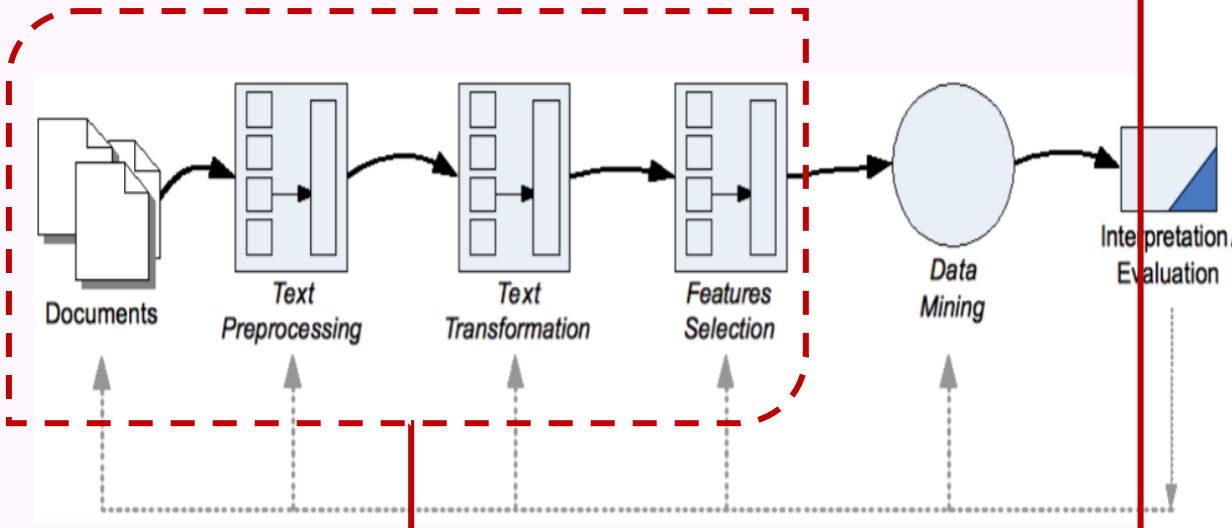
- Syntax: How do the words relate to each other?
  - The dog bit the man. ≠ The man bit the dog.
  - But in Russian: человек собаку съел = человек съел собаку

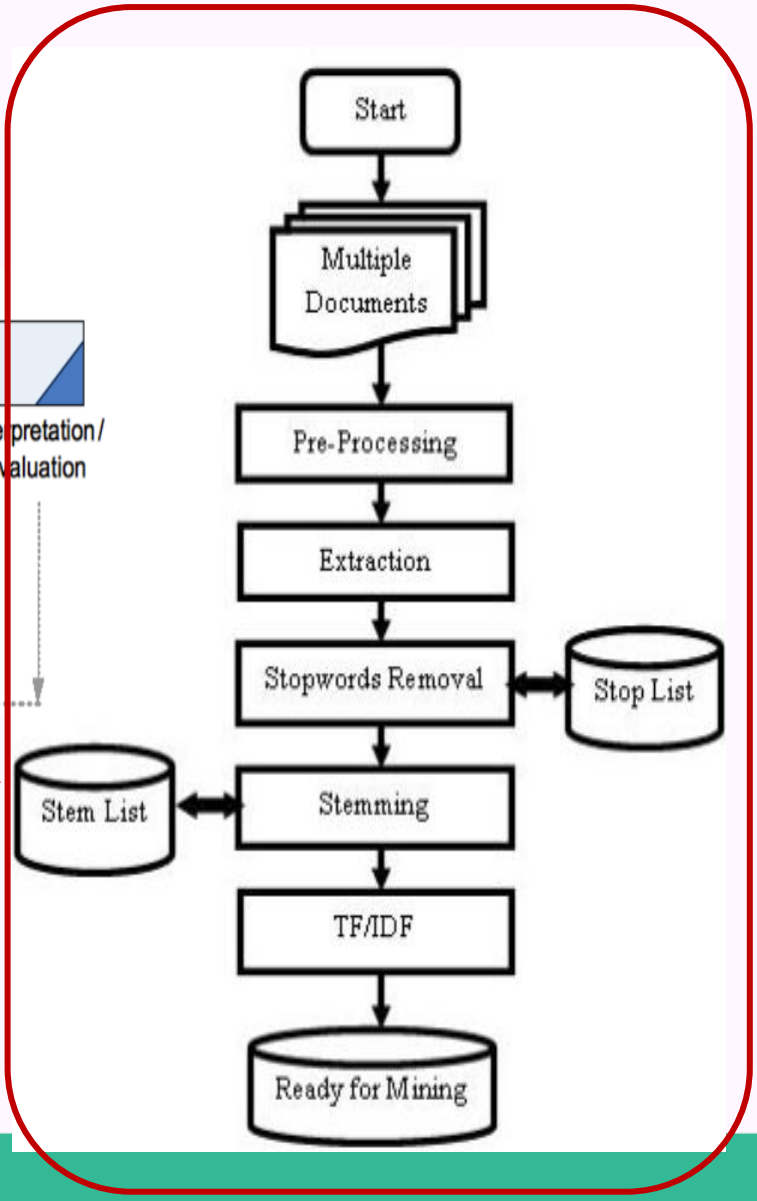
Language Processing

- Semantics: How can we infer meaning from sentences?
  - I saw the man on the hill with the telescope.
  - The ipod is so small!
  - The monitor is so small!
- Discourse: How about across many sentences?
  - President Bush met with President-Elect Obama today at the White House. He welcomed him, and showed him around.
  - Who is “he”? Who is “him”? How would a computer figure that out?

Preprosessing

Tahapan (bagian dari data/teks mining)





Pra pemrosesan Teks

- merubah data tekstual orisinal dalam bentuks struktur yang data-mining-ready
- fase paling kritis dan paling kompleks
- tujuan: mendapatkan fitur kunci dan term kunci dari dokumen teks untuk memperkuat relevansi antara kata dan dokumen , dan relevansi kata dan kategori
- merepresentasikan setiap dokumen sebagai vektor fitur, dengan membagi menjadi unsur terkecil dari dokumen teks --> kata

Aktifitas Pra pemrosesan

- Stopword Removal:
  - tidak semua kata memiliki “informasi” yang dibutuhkan
- Stemming (dan atau lemmatization):
  - teknik untuk mencari akar (root)/stem dari kata,
  - contoh kata user, users, using, used memiliki akar/stem: “USE”
  - stemmer: Porter stemmer (Inggris), Indonesia stemmer (Algoritma Nazief & Adriani)
  - lemmatization: am, is, was, are --> be
- Document indexing:
  - tujuan utama: meningkatkan efisiensi dgn mengekstrak dokumen hasil suatu term terpilih yg digunakan untuk mengindeks
  - dengan memilih himpunan kata kunci yang pantas berdasarkan corpus document, dan memberikan bobot tiap kata kunci tsb ke dokumen tertentu, yang kemudian
  - mentransformasikan setiap dokumen menjadi vektor bobot kata kunci
  - bobotnya berelasi dgn frekuensi kemunculan term dalam dokumen, dan banyak dokumen yg menggunakan term tsb

Aktifitas Pra pemrosesan

- Document Indexing: Term Weighting --> TF-IDF
- Dimensionality Reduction (DF)
  - banyaknya dokumen dimana suatu term muncul
  - cara paling simpel untuk reduksi kosa kata (vocabulary)

Stop word removal

- stop word (SW) perlu dihilangkan dalam dokumen teks karena membuat dokumen lebih berat dan kurang penting utk dianalisis
- penghilangan SW mengurangi dimensi dari ruang term.
- umumnya penghilangan atas preposisi dan pro-noun yang tidak memberikan arti thdp dokumen
- contoh:
  - inggris: *the, in, an, with*, dll
  - indonesia: *ini, itu, kami, di, ke, dari, pada, dengan, si*, dll

stop word removal tidak sesuai diterapkan untuk dokumen yg sensitive, misal: kitab suci

(rio bagus prakoso, 2017)

Stop word removal

Jenis metode SW:

- metode klasik: SW didapat dari daftar pre-compiled
- Zipf-law (Z-Method):
- Mutual Information Methode (MI)
- Term Based Random Sampling (TBRS) --> thn 2005:
  - secara manual mendeteksi stop word dari dokumen web. menggunakan pengukuran divergensi Kull :

$$d_x(t) = P_x(t).\log_2 \frac{Px(t)}{p(t)}$$

dimana Px(t) adalah normalisasi frekuensi term t dalam mass x,

P(t) adalah normalisasi frekuensi term t dalam koleksi keseluruhan.

stop list dibentuk dgn mengambil least informative term dalam seluruh chunks, menghapus seluruh duplikasi

Zipf law: didasari observasi/empiris, bukan teori

- Zipf law:
  - semakin sering suatu kata yang digunakan pada suatu bahasa tertentu akan menyebabkan kata tersebut semakin sering untuk digunakan kembali, namun semakin jarang suatu kata yang digunakan pada suatu bahasa tertentu akan menyebabkan kata tersebut semakin jarang untuk digunakan kembali.

$$P(r) \approx \frac{1}{r \ln(1.78\,R)},$$

where R is the number of different words.

  - terdapat suatu rasio penyebab 20:80 dan akibat 80:20.

Contoh 1, 20% pelanggan dari suatu perusahaan akan menghasilkan 80% dari total pemasukan perusahaan tersebut.

Contoh 2, 20% pelanggan dari suatu perusahaan akan mengakibatkan permasalahan sebesar 80% pada perusahaan tersebut. Hal ini menyebabkan fokus untuk menyelesaikan suatu permasalahan dapat diarahkan kepada 20% tersebut, maka permasalahan akan selesai.

Stemming vs Lemmatization

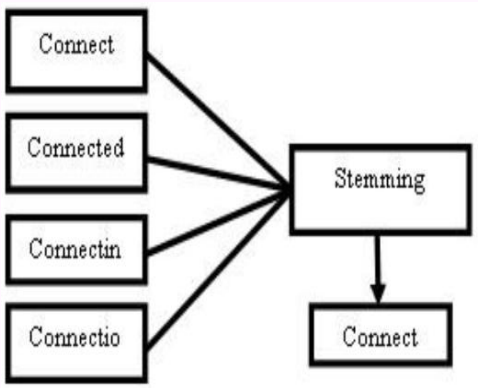
Lemmatization

- proses merubah kata dari kalimat ke dalam bentuk (rujukan) kamus
- analisis morphologi
- perlu kamus lengkap
- misal, kata *amusing*, *amusement*, and *amused above*, lemma untuk tiap kata --> *amuse*

Stemming

- proses merubah kata dari kalimat ke dalam porsi tidak berubah
- biasanya mereduksi berbagai suffixes (akhiran), jika perlu preffix (awalan)
- misal, kata *amusing*, *amusement*, and *amused above*, di stem --> *amus*

Stemming



sama2 mencari root/stem dari kata kalimat

Stemming dalam NLTK

- Misal dgn stemmer Lancaster

```
lancaster = nltk.LancasterStemmer()
stems = [lancaster.stem(i) for i in tokens]

```

- Outputnya:

```
['omg', ',', 'nat', 'langu', 'process', 'is', 'so', 'cool', 'and', 'i', "'m", '
```

- Misal dgn stemmer Porter

```
porter = nltk.PorterStemmer()
stem = [porter.stem(i) for i in tokens]

```

- Outputnya

```
['omg', ',', 'natur', 'languag', 'process', 'is', 'so', 'cool', 'and', 'i', "'m
```

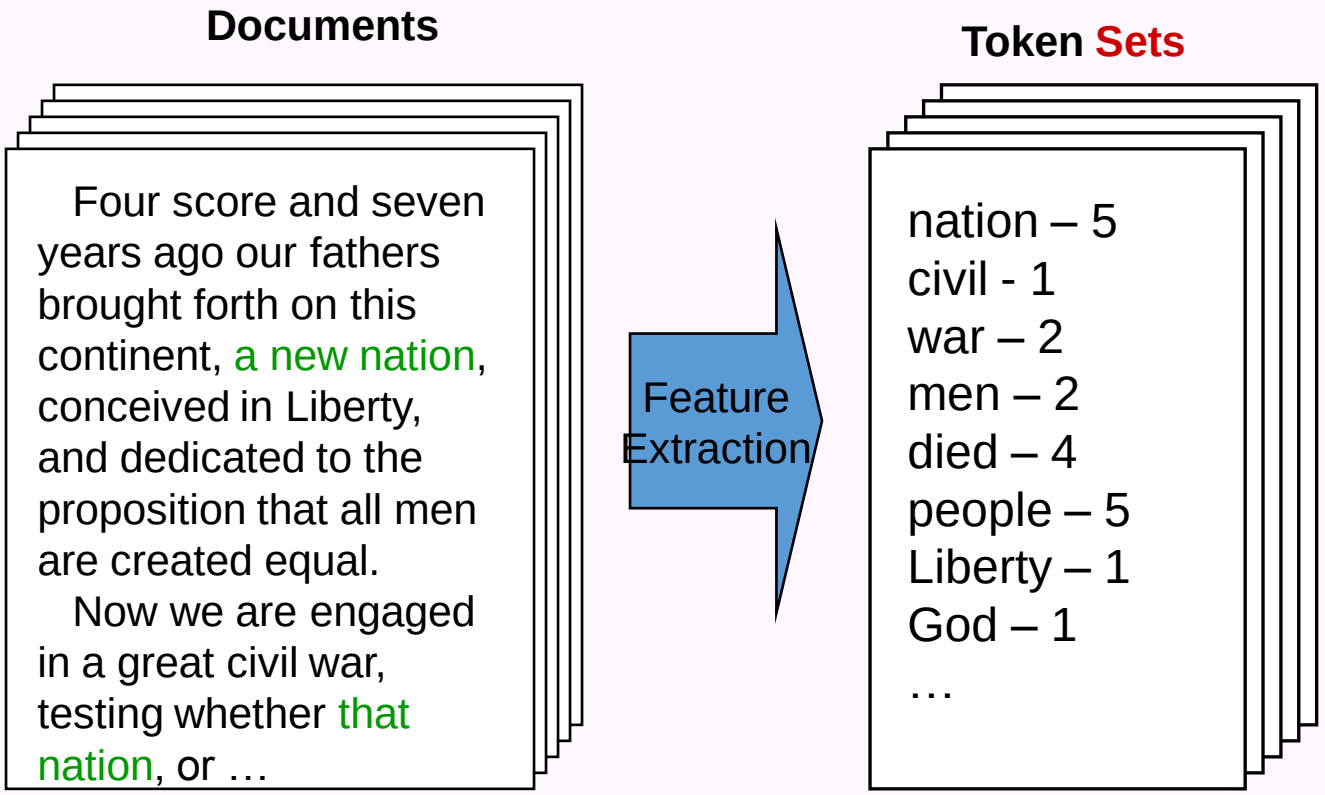
pada stemmer Porter, kata “natural” dipetakan ke “natur” dibanding “nat” pada lancaster.
dan kata “really” dipetakan ke realli dibanding “real”

Lemmatization dalam NLTK

```
1  from nltk import WordNetLemmatizer
2
3  lemma = nltk.WordNetLemmatizer()
4  text = "Women in technology are amazing at coding"
5  ex = [i.lower() for i in text.split()]
6
7  lemmas = [lemma.lemmatize(i) for i in ex]
8
9  ['woman', 'in', 'technology', 'are', 'amazing', 'at', 'coding']
10
```

- Kata “women” diubah/petakan ke “woman”

Pendekatan Bag-of-Tokens



Loses all order-specific information!
Severely limits context!

Chunking Teks

- Biasa disebut shallow/parsial parsing
- Identifikasi part-of-speech dan frase pendek (noun phrases) kata benda
- merupakan struktur non-rekursif yang dapat ditangani dengan metode stata hingga

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only $1.8 billion] [PP in] [NP September]

- Karena full parsing expensive dan tidak terlalu robust
- Contoh kita ingin tahu entitas nam,misal President Obama dalam dokumen berikut;

President Barack Obama criticized insurance companies and banks as he urged supporters to pressure Congress to back his moves to revamp the health-care system and overhaul financial regulations. (source)

Chunking Teks

- Chunking dapat dilakukan pada saat ingin mengetahui entitas bernama (named-entitas), seperti:
  - nama orang
  - organisasi
  - ekspresi waktu (tahun)
  - lokasi
  - nilai uang
  - persentase
  - dll

- misal, dokumen tak beranotasi/berlabel:

Jim bought 300 shares of Acme Corp. in 2006.

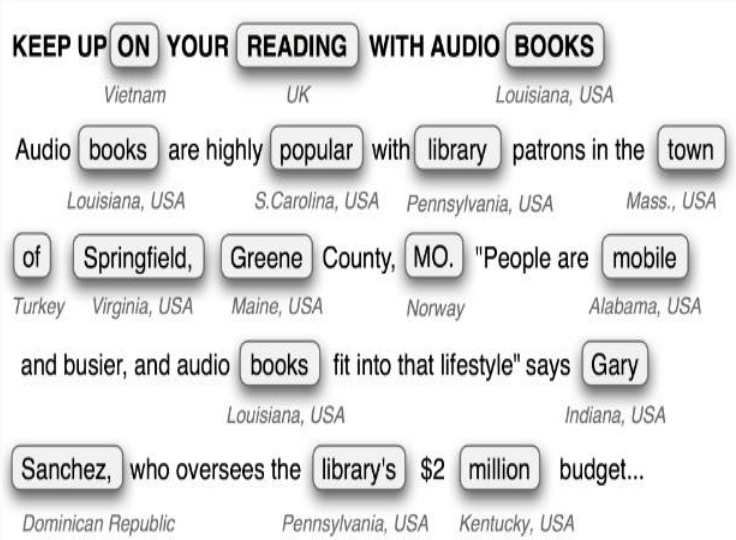
- setelah di-chunking:

$[\text{Jim}]_{\text{Person}}$ bought 300 shares of $[\text{Acme Corp.}]_{\text{Organization}}$ in $[\text{2006}]_{\text{Time}}$.

Proses ini disebut: **N**amed-**E**ntity **R**ecognition

Chunking Teks

- Deteksi Lokasi

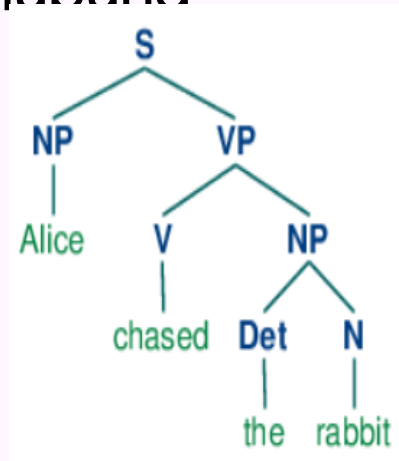


- NER

NE Type	Examples
ORGANIZATION	*Georgia-Pacific Corp., WHO*
PERSON	*Eddy Bonte, President Obama*
LOCATION	*Murray River, Mount Everest*
DATE	*June, 2008-06-29*
TIME	*two fifty a m, 1:30 p.m.*
MONEY	*175 million Canadian Dollars, GBP 10.40*
PERCENT	*twenty pct, 18.75 %*
FACILITY	*Washington Monument, Stonehenge*
GPE	*South East Asia, Midlothian*

(Parsing) Tree

- Kumpulan titik berlabel yang terhubung



- format teks dari Tree

```
(S
   (NP Alice)
   (VP
      (V chased)
      (NP
         (Det the)
         (N rabbit))))
```

- Tree dalam NLTK

```
>>> tree1 = nltk.Tree('NP', ['Alice'])
>>> print(tree1)
(NP Alice)
>>> tree2 = nltk.Tree('NP', ['the', 'rabbit'])
>>> print(tree2)
(NP the rabbit)
```

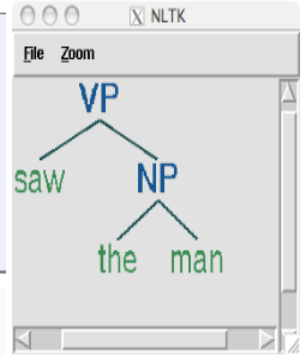
We can incorporate these into successively larger trees as follows:

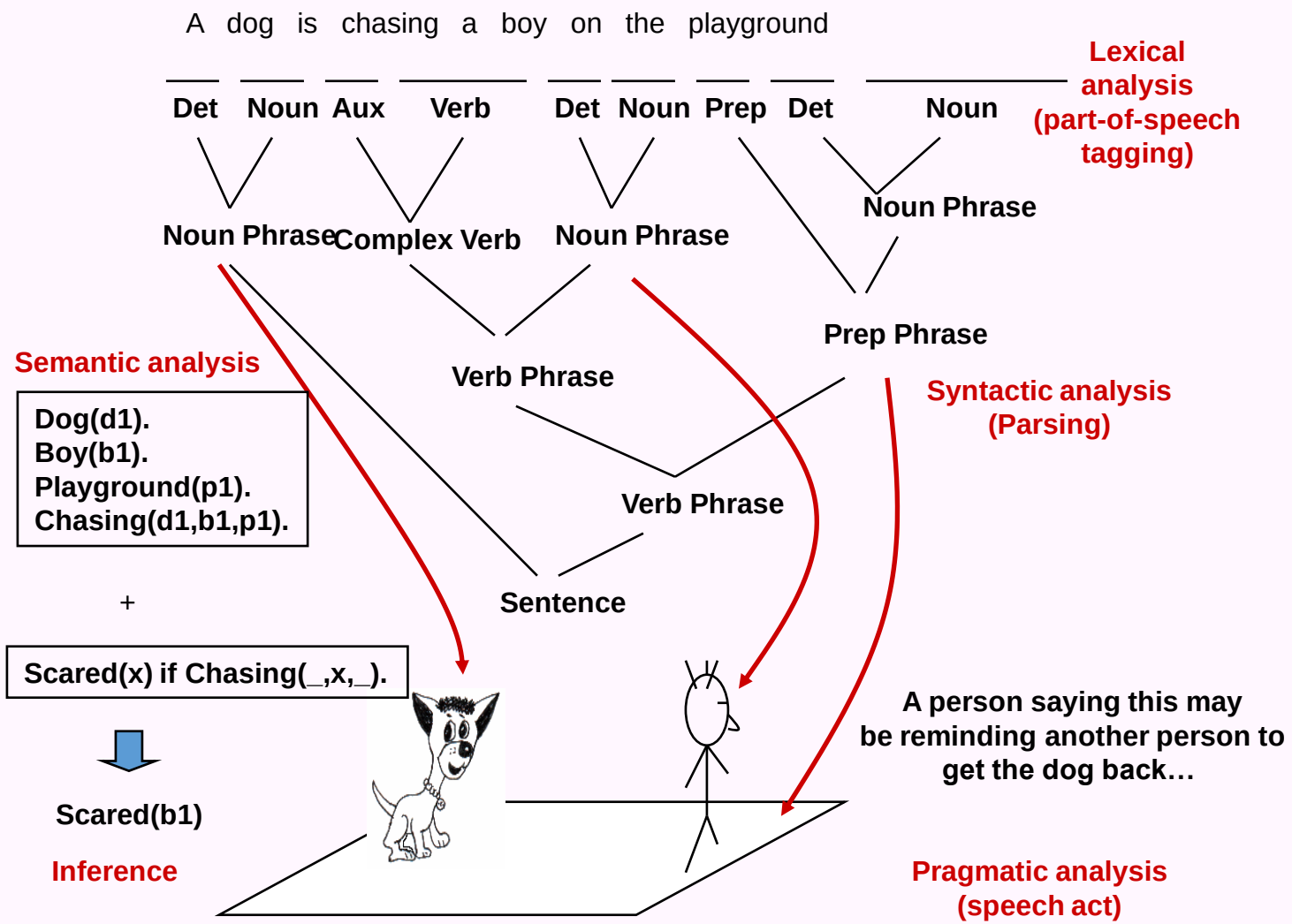
```
>>> tree3 = nltk.Tree('VP', ['chased', tree2])
>>> tree4 = nltk.Tree('S', [tree1, tree3])
>>> print(tree4)
(S (NP Alice) (VP chased (NP the rabbit)))
```

Here are some of the methods available for tree objects:

```
>>> print(tree4[1])
(VP chased (NP the rabbit))
>>> tree4[1].label()
'VP'
>>> tree4.leaves()
['Alice', 'chased', 'the', 'rabbit']
>>> tree4[1][1][1]
'rabbit'
```

```
>>> tree3.draw()
```





(Taken from ChengXiang Zhai, CS 397cxz – Fall 2003)

Parsing Struktur Frase

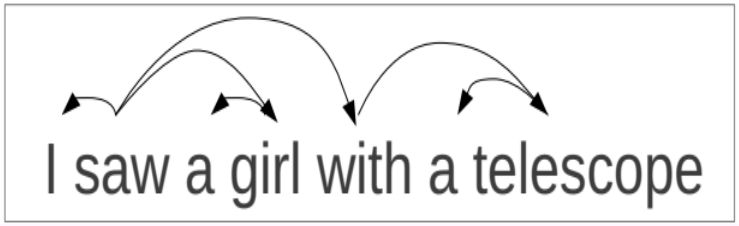
Interpreting Language is Hard!

I saw a girl with a telescope

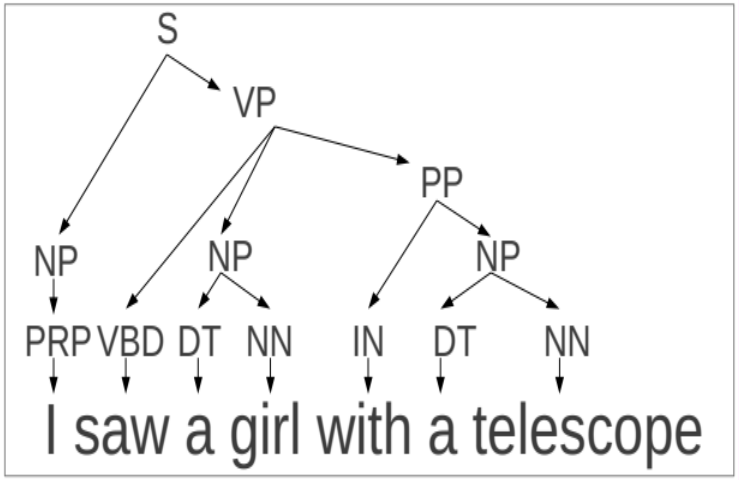
- "Parsing" resolves structural ambiguity in a formal way

Parsing Struktur Frase: 2 Jenis Parsing

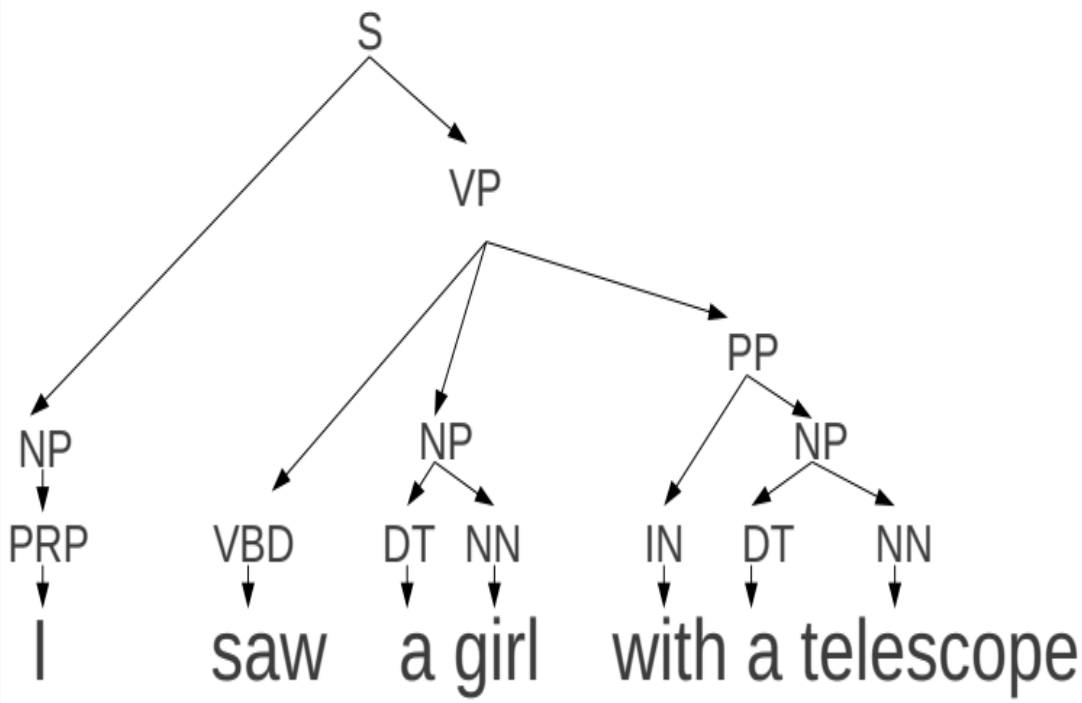
- Dependensi: berfokus pada relasi antara dua kata



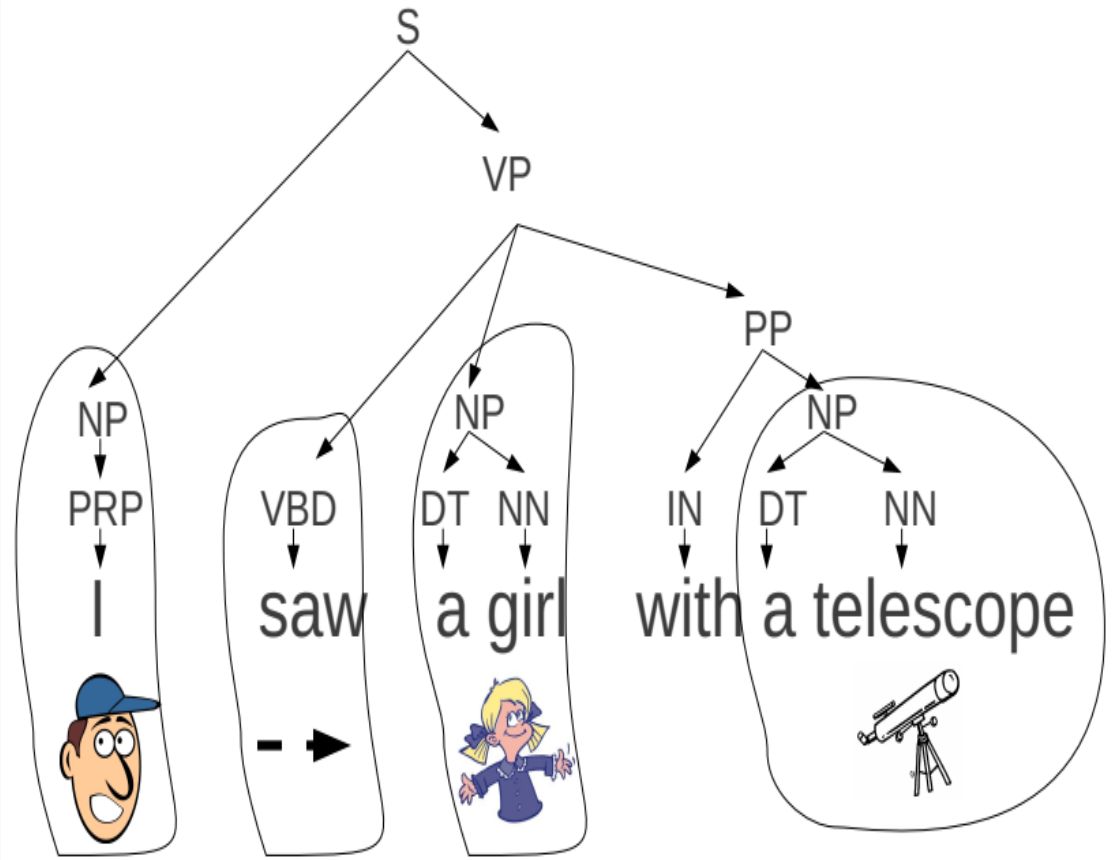
- Struktur frase: berfokus pada identifikasi frase dan struktur rekursifnya



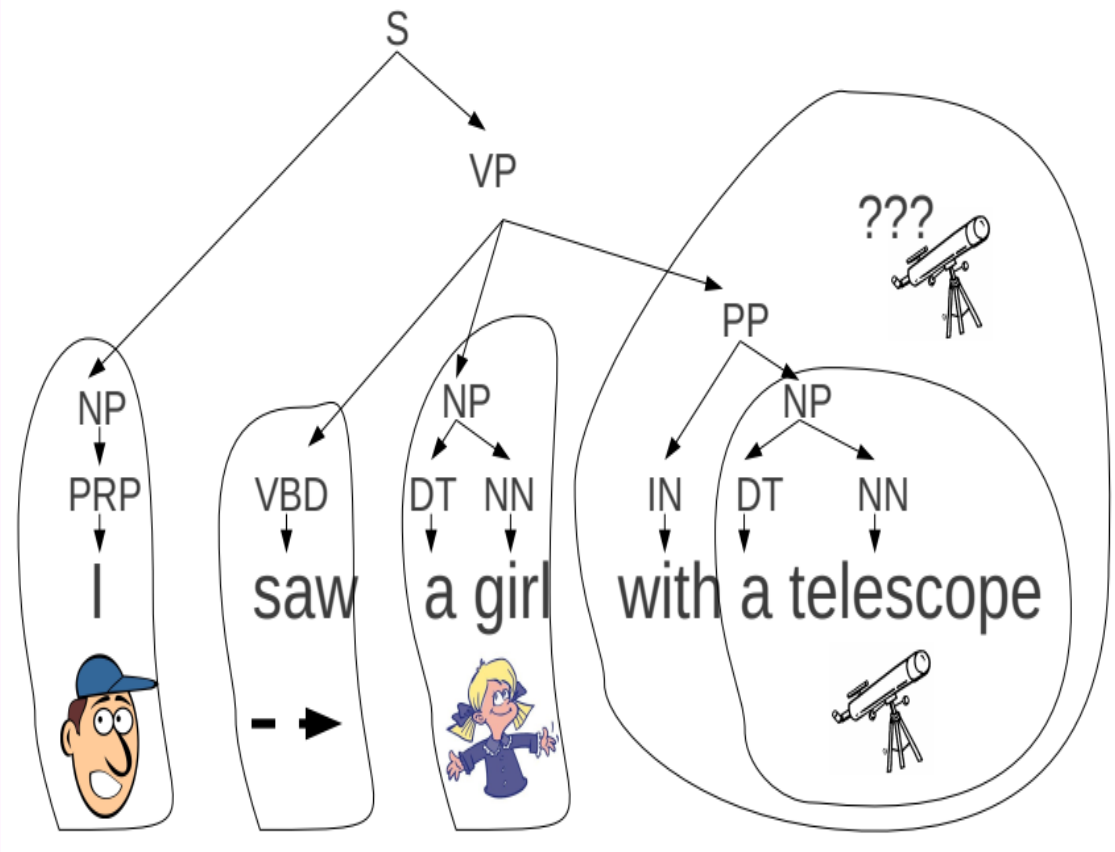
Parsing Struktur Frase: Struktur Rekursif



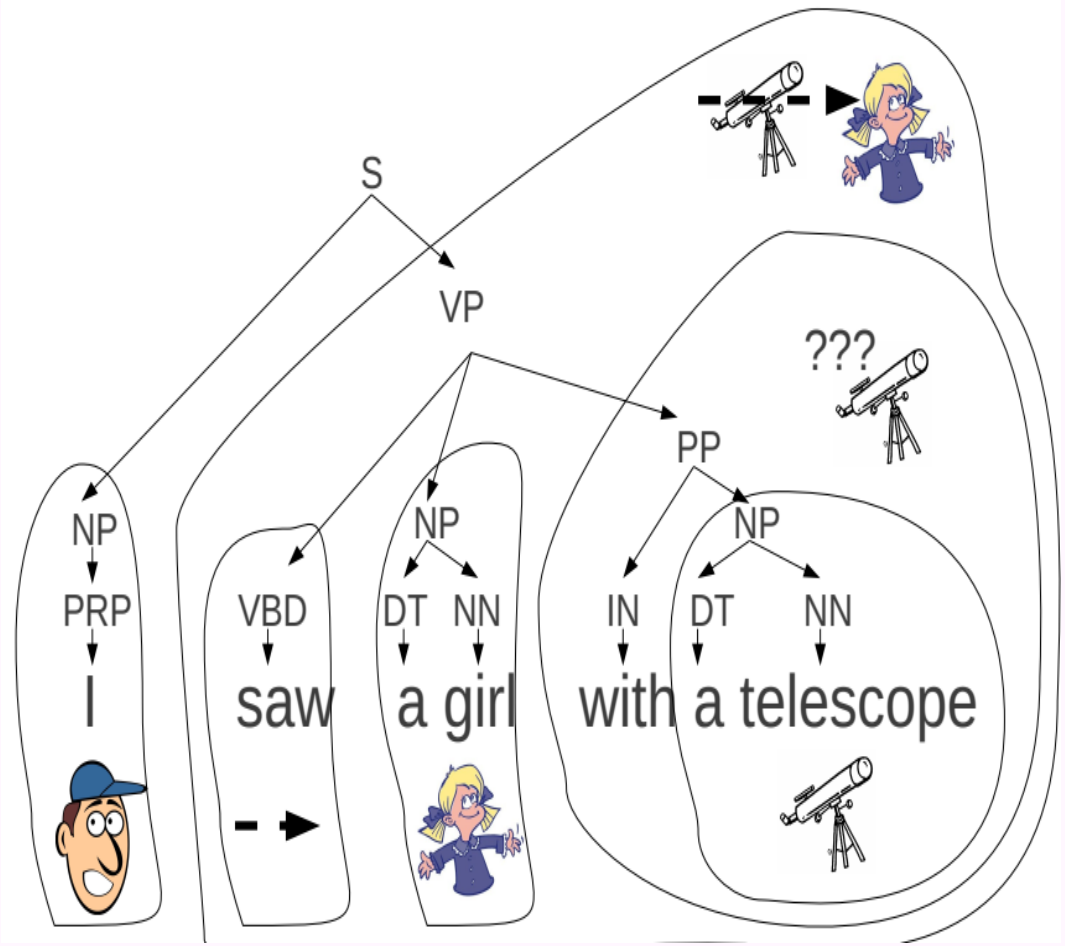
Parsing Struktur Frase: Struktur Rekursif



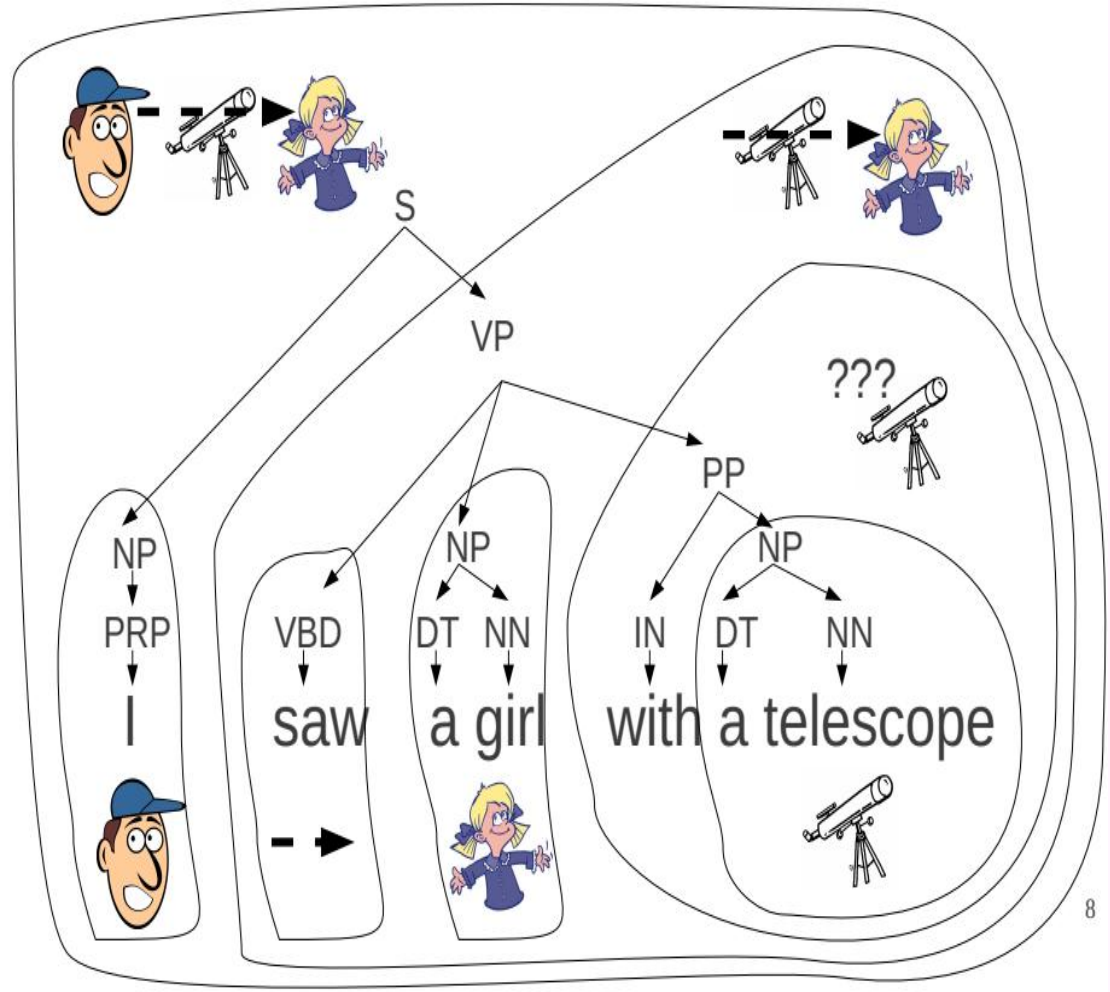
Parsing Struktur Frase: Struktur Rekursif



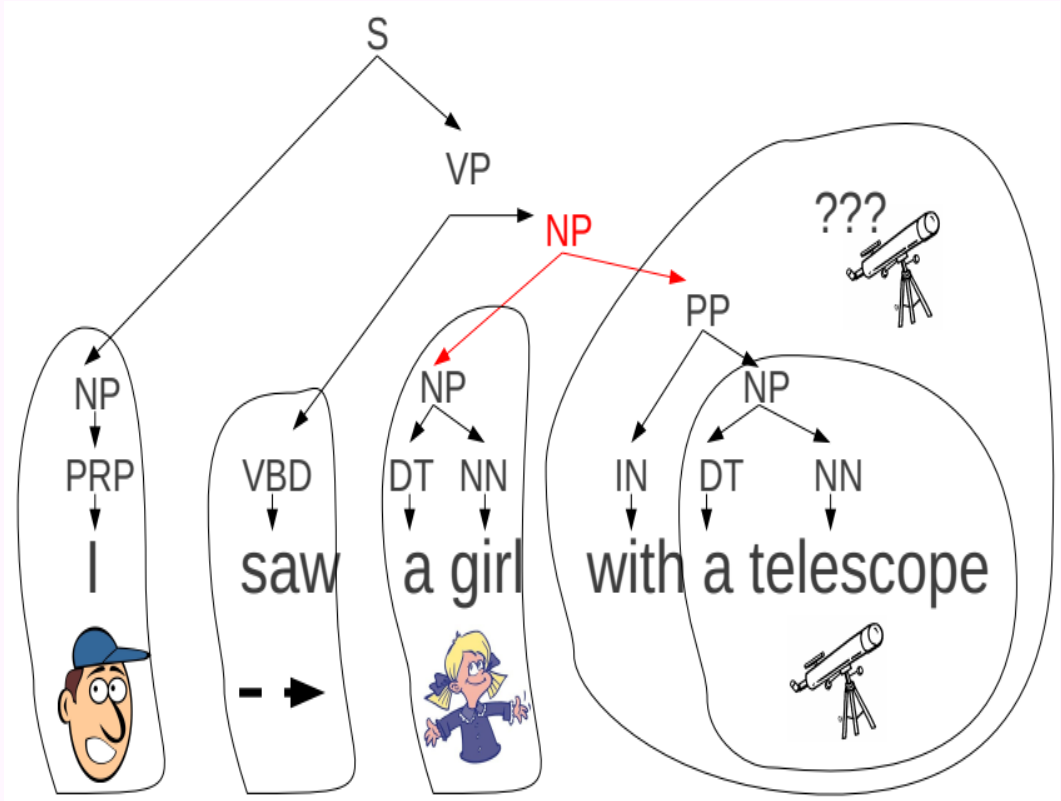
Parsing Struktur Frase: Struktur Rekursif



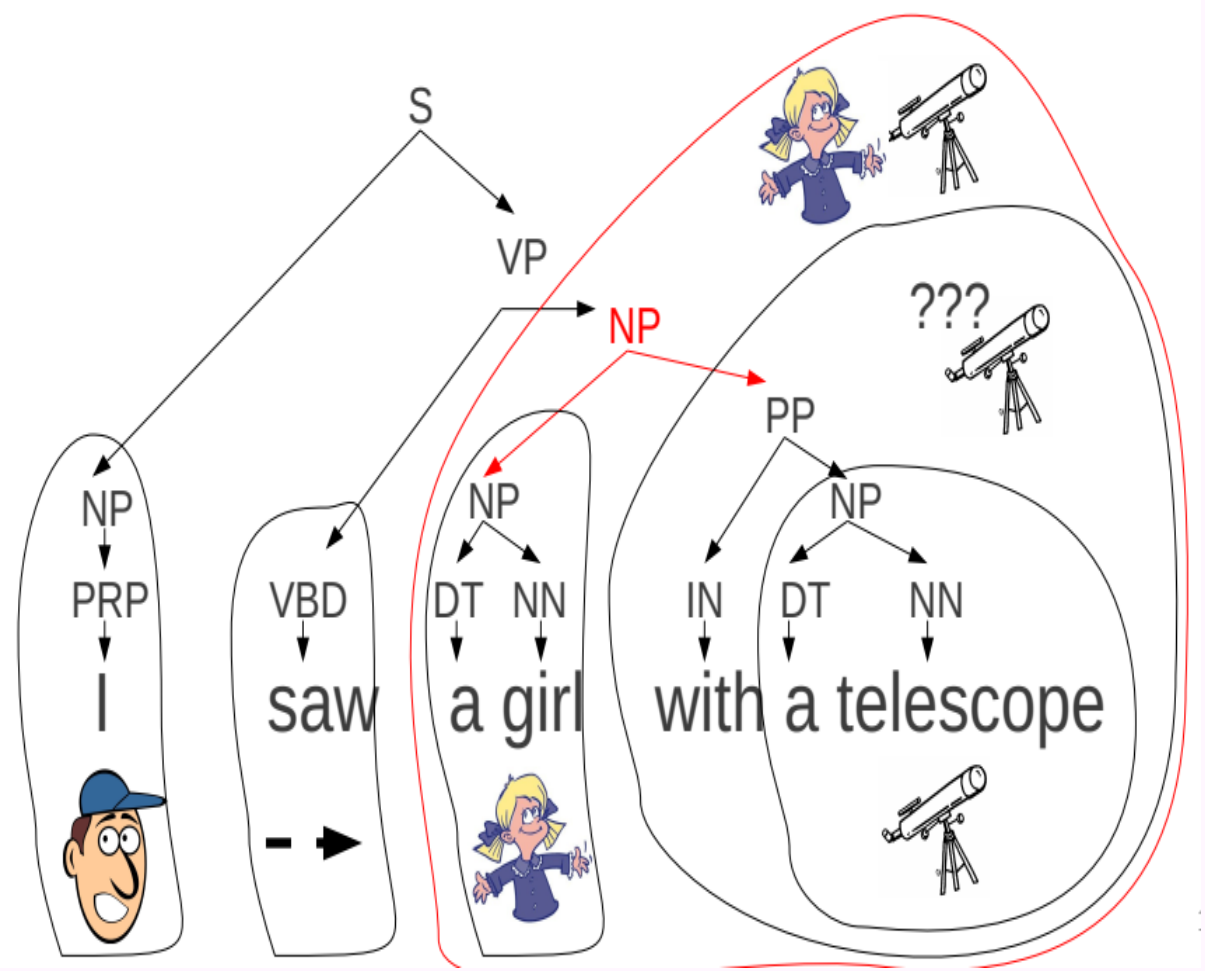
Parsing Struktur Frase: Struktur Rekursif



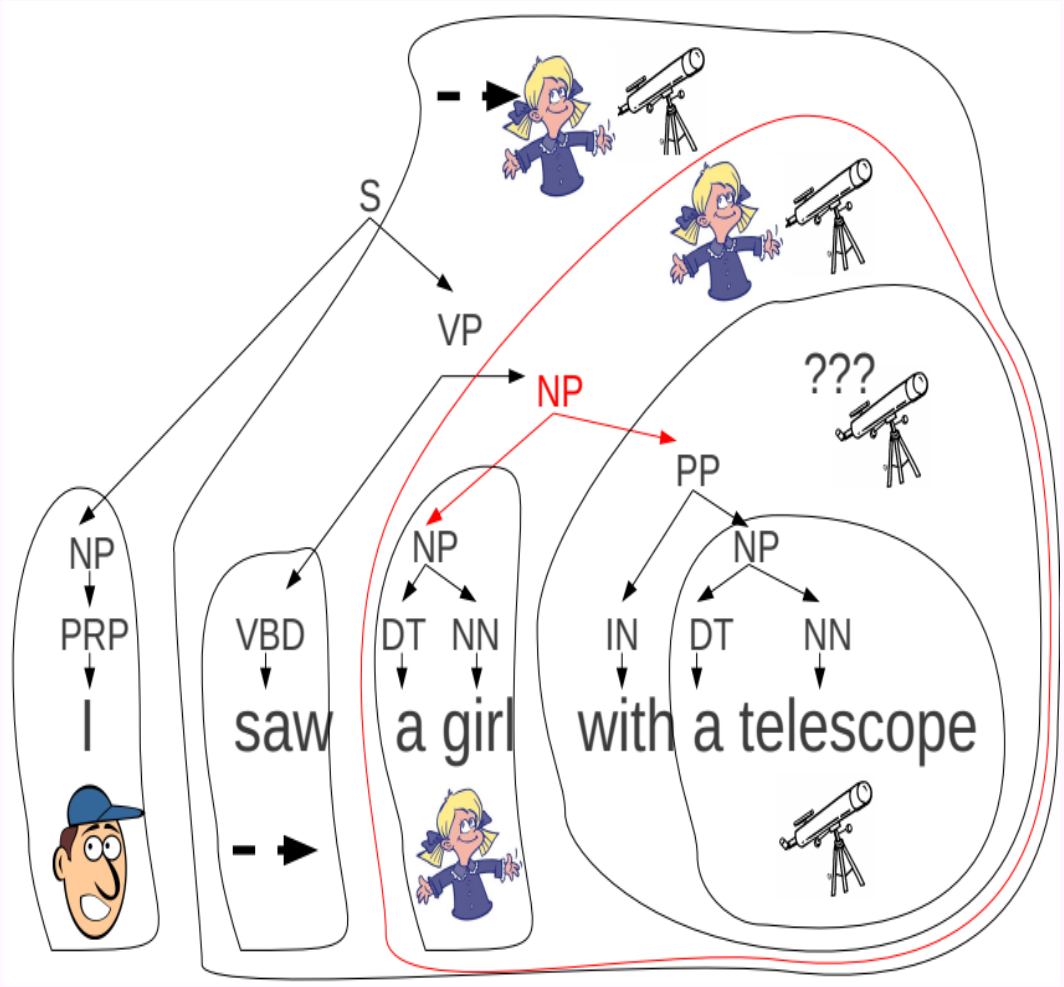
Strukur Beda, Interpretasi Beda



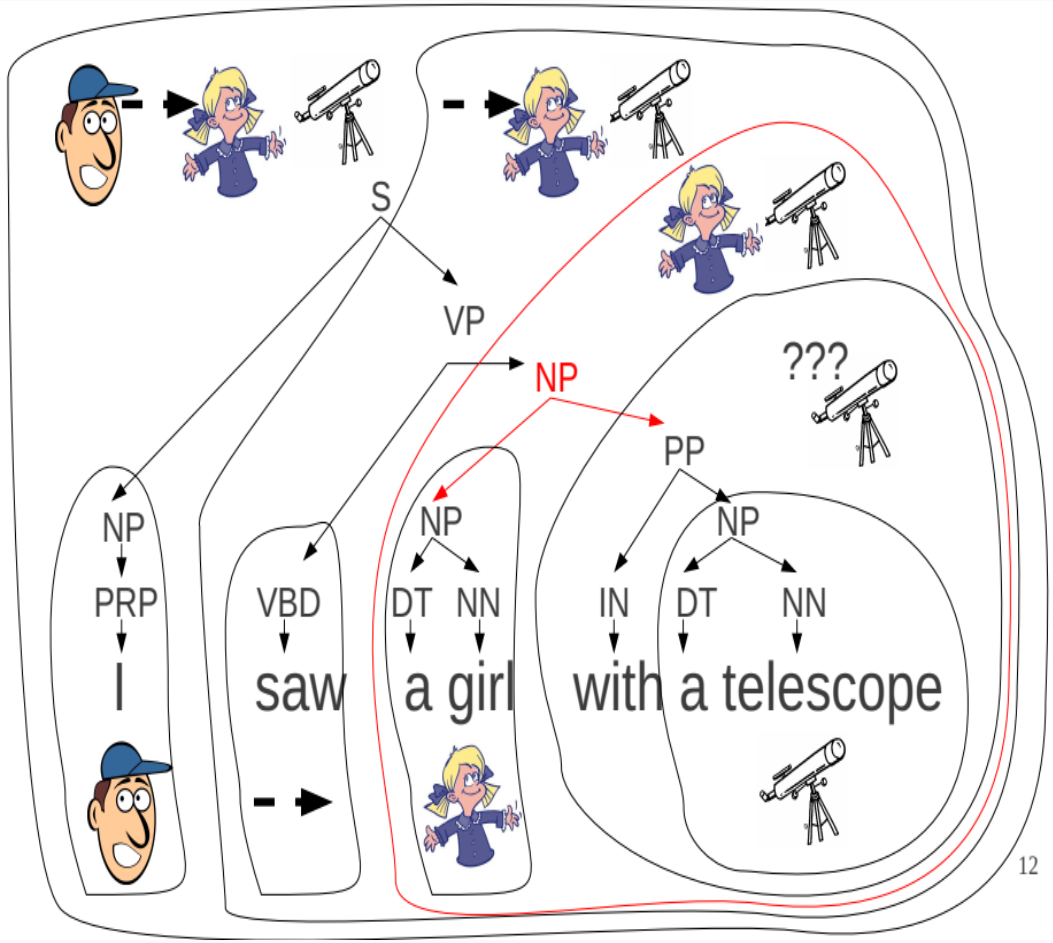
Strukur Beda, Interpretasi Beda



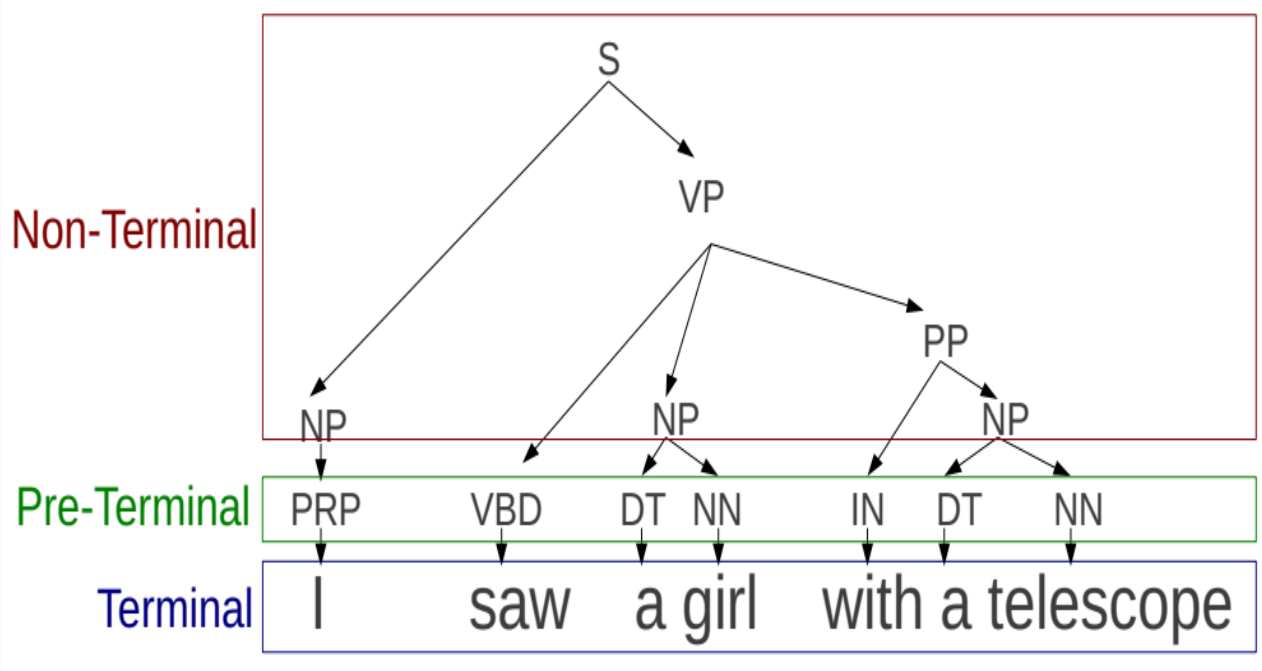
Strukur Beda, Interpretasi Beda



Strukur Beda, Interpretasi Beda

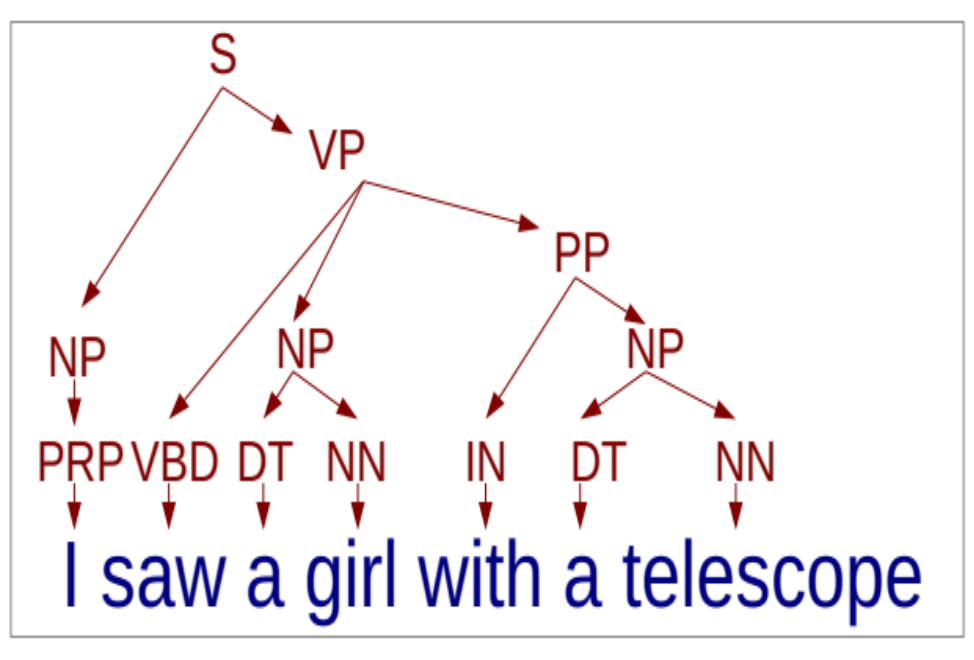


Non Terminal, Pre Terminal dan Terminal



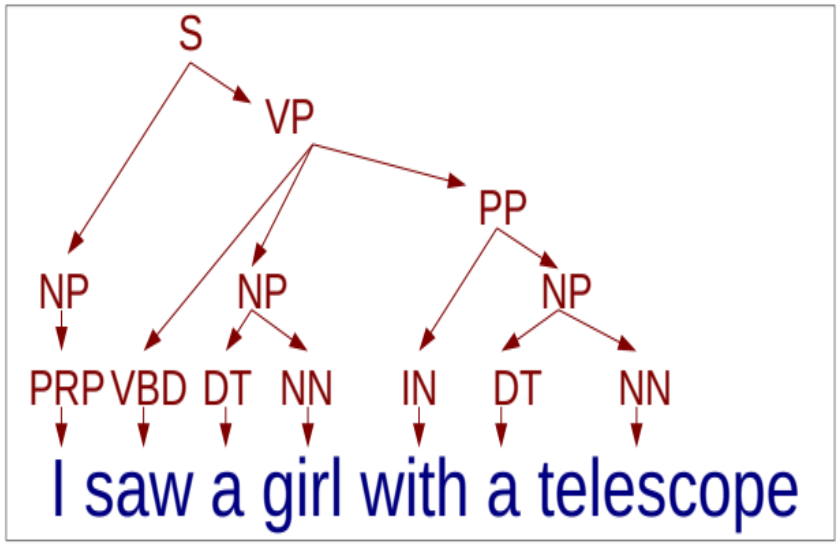
Parsing sebagai suatu Prediksi

- Diberikan kalimat X, prediksi parse tree Y



Parsing Model Probabilistik

- Diberikan/diketahui kalimat X, prediksi probabilitas terbesar dari parse tree Y

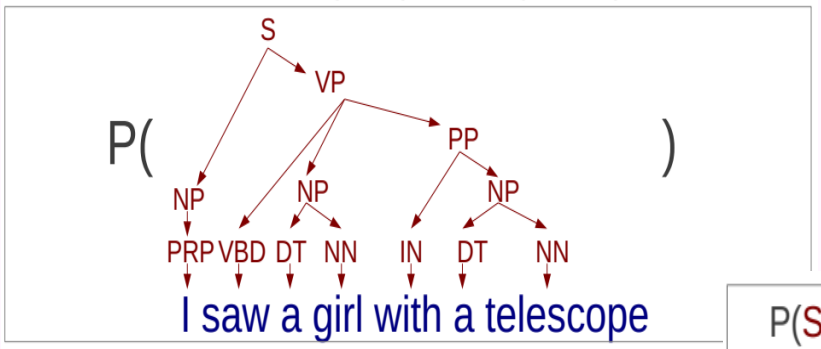


$$\underset{Y}{\operatorname{argmax}} P(Y|X) = \underset{Y}{\operatorname{argmax}} P(Y, X)$$

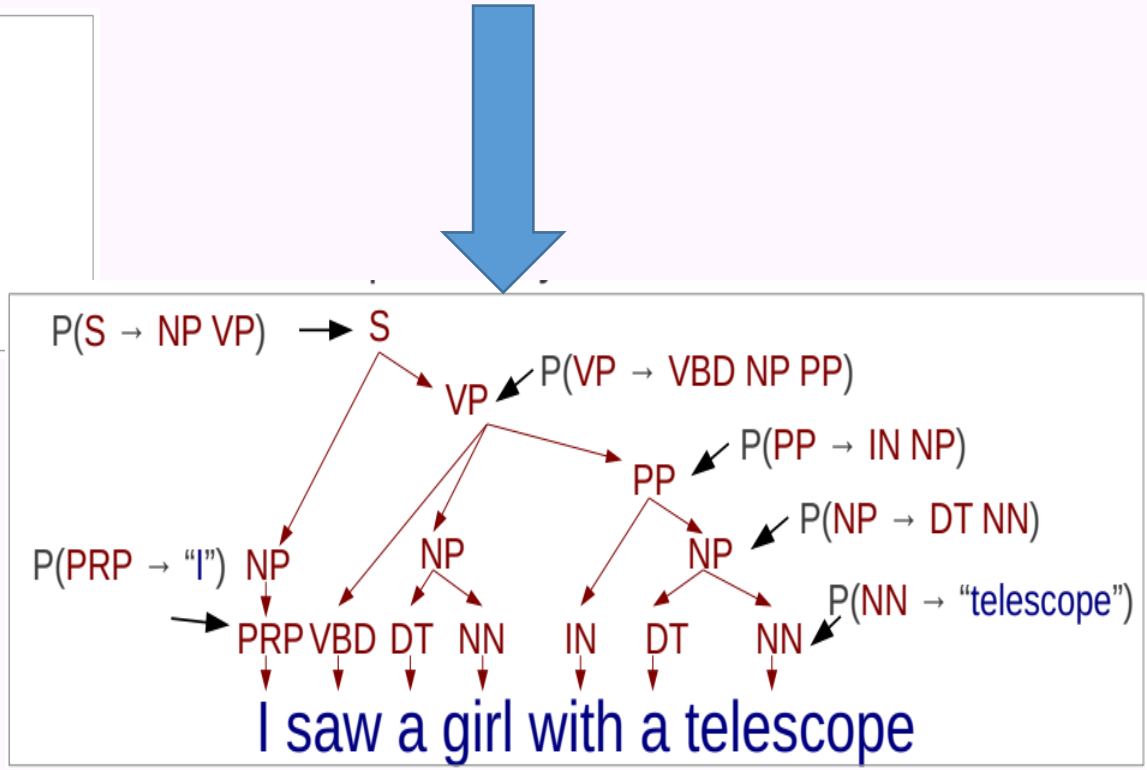
$$\underset{Y}{\operatorname{argmax}} P(Y|X)$$

Probabilistic Context Free Grammar (PCFG)

- Bagaimana mendefinisikan probabilitas gabungan untuk parse tree brkt:

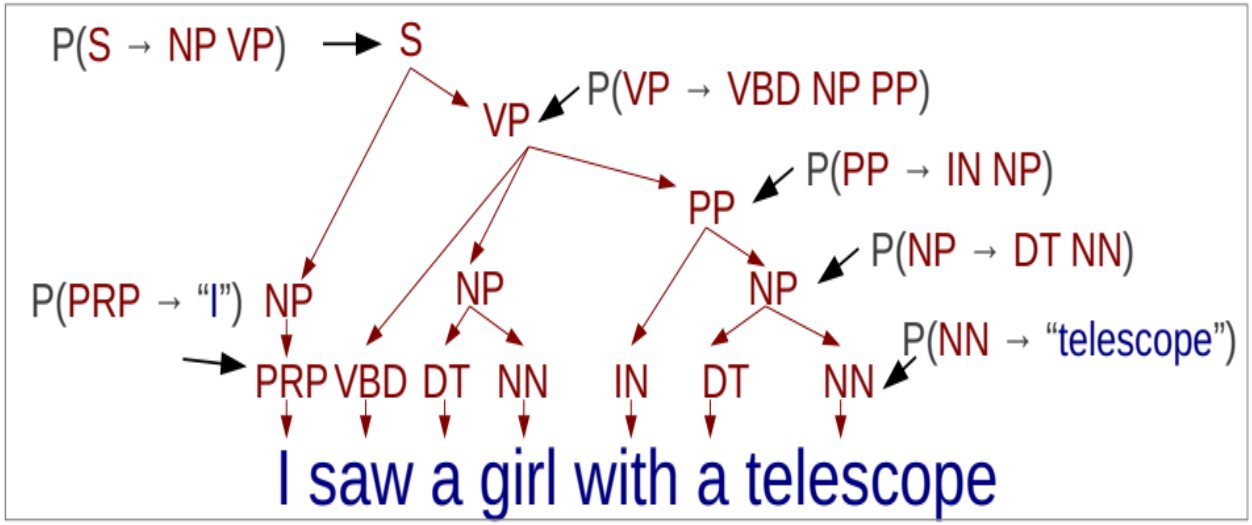


- PCFG: Definisikan probabilitas utk tiap node



Probabilistic Context Free Grammar (PCFG)

- PCFG: Define probability for each node



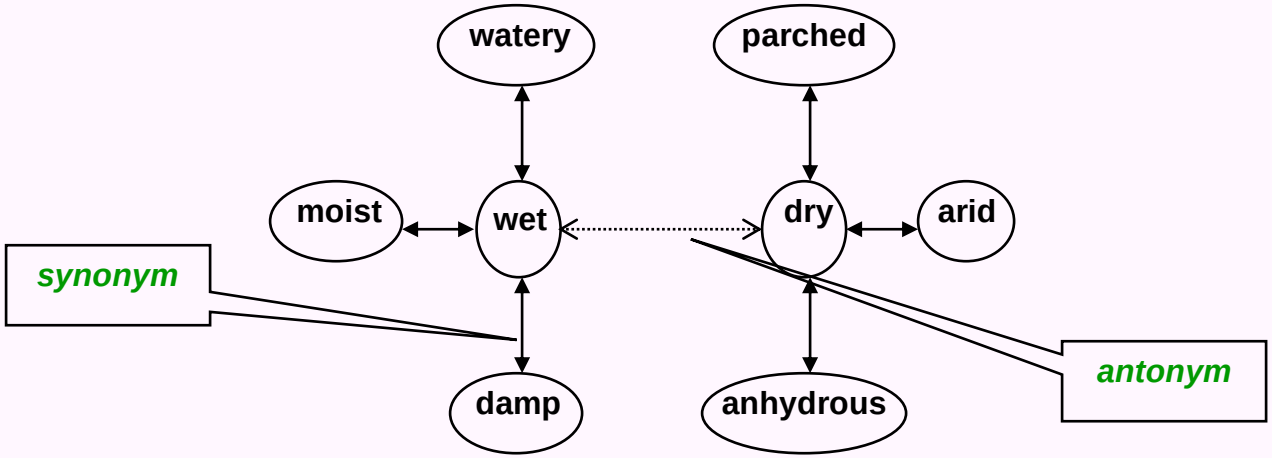
- Parse tree probability is product of node probabilities

P(S → NP VP) * P(NP → PRP) * P(PRP → "I")
* P(VP → VBD NP PP) * P(VBD → "saw") * P(NP → DT NN)
* P(DT → "a") * P(NN → "girl") * P(PP → IN NP) * P(IN → "with")
* P(NP → DT NN) * P(DT → "a") * P(NN → "telescope")

Linguistik

- Progress on Useful Sub-Goals:
  - English Lexicon
  - Part-of-Speech Tagging
  - Word Sense Disambiguation
  - Phrase Detection / Parsing

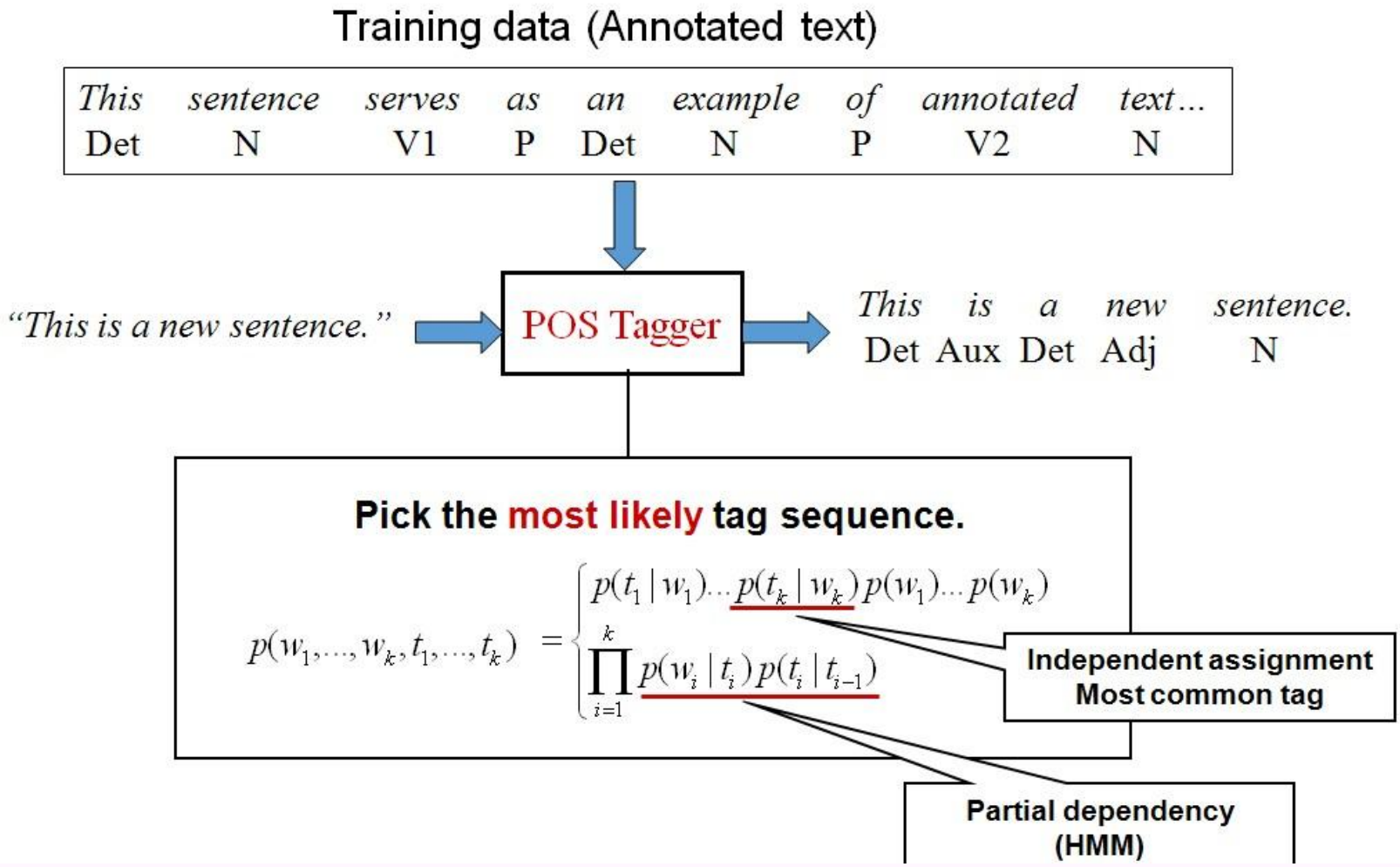
Wordnet



An extensive lexical network for the English language

- Contains over 138,838 words.
- Several graphs, one for each part-of-speech.
- *Synsets* (synonym sets), each defining a semantic sense.
- Relationship information (antonym, hyponym, meronym …)
- Downloadable for free (UNIX, Windows)
- Expanding to other languages (Global WordNet Association)
- Funded >$3 million, mainly government (translation interest)
- Founder George Miller, National Medal of Science, 1991.

Part of Speech Tagging



Parsing

Choose most likely parse tree…

Probabilistic CFG

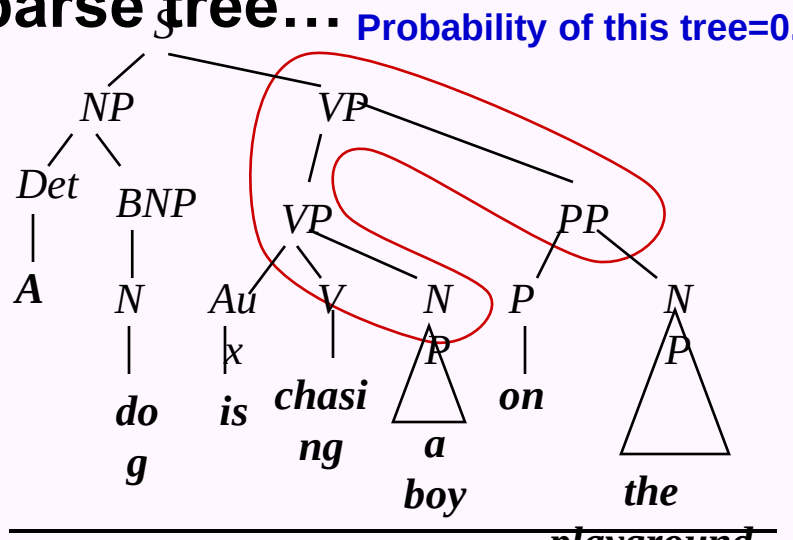
Grammar

$S \rightarrow NP\ VP$	1.0
$NP \rightarrow Det$	0.3
BNP	0.4
$NP \rightarrow BNP$	0.3
$NP \rightarrow NP\ PP$	…
$BNP \rightarrow N$	
$VP \rightarrow V$	
$VP \rightarrow Aux\ V$	…
NP	1.0
$VP \rightarrow VP\ PP$	
$PP \rightarrow P\ NP$	0.01

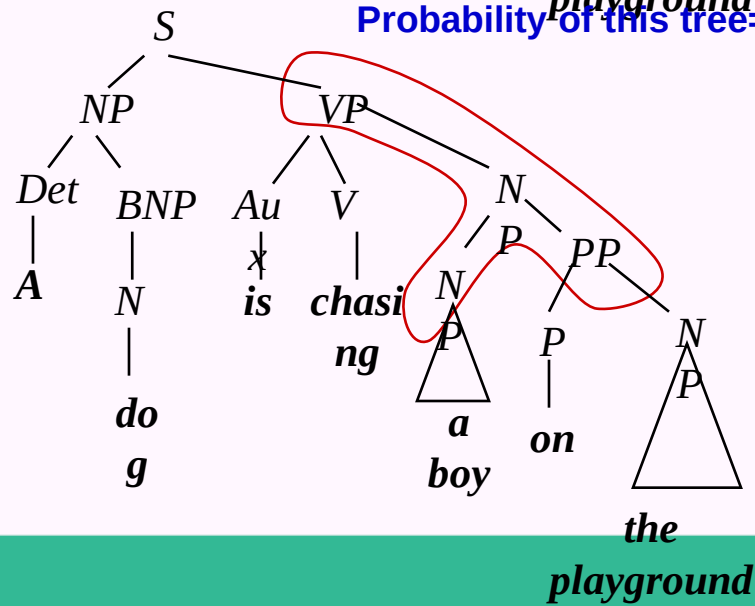
Lexicon

$V \rightarrow chasing$	0.003
$Aux \rightarrow is$	
$N \rightarrow dog$	…
$N \rightarrow boy$	
$N \rightarrow playground$	…
$Det \rightarrow the$	

Probability of this tree=0.000015

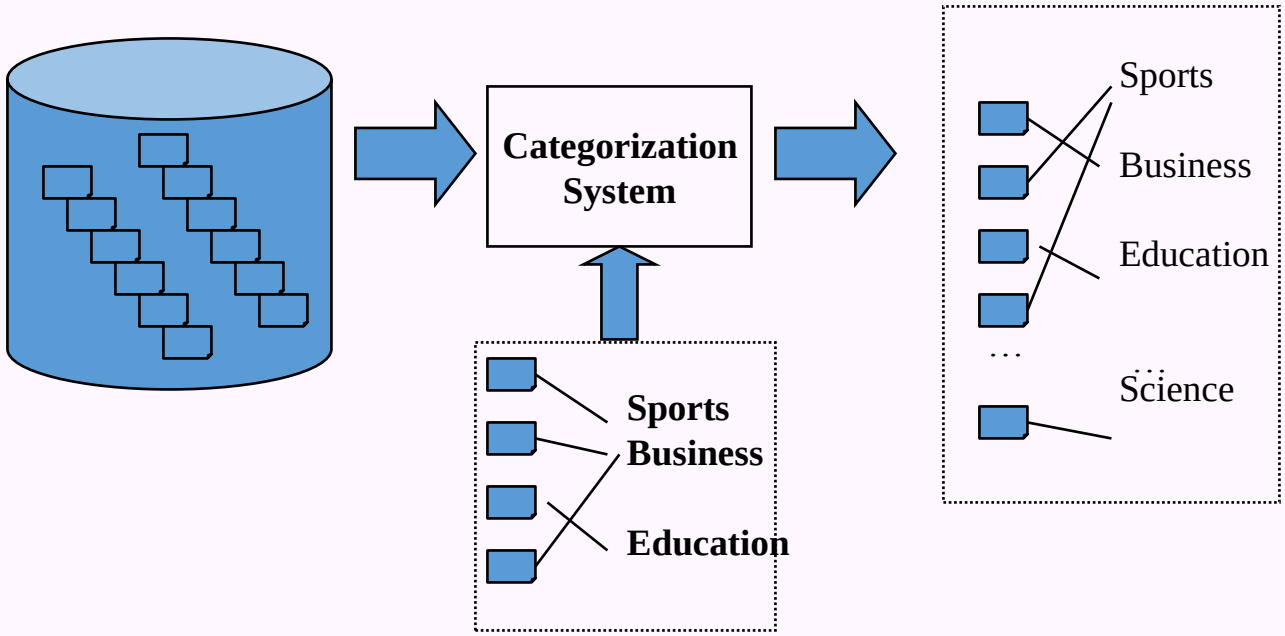


Probability of this tree=0.000011



Kategorisasi Teks

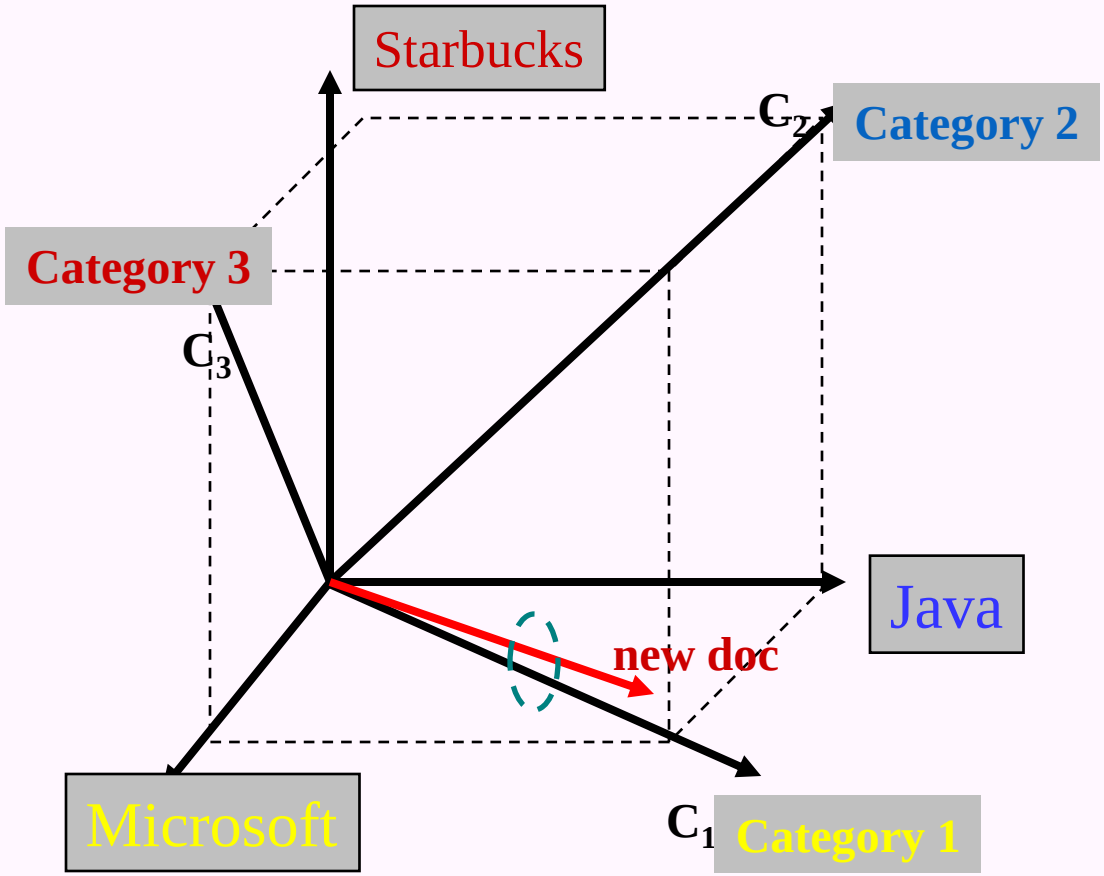
- Pre-given categories dan contoh dokument berlabel (Categories may form hierarchy)
- Mengklasifikasikan dokumen baru
- Masalah klasifikasi standar (supervised learning)



Model Ruang Vector (Vector Space Model)

- MerRepresentasikan suatu dokumen dengan vektor term
  - Term: basic concept, e.g., word or phrase
  - Setiap term mendefinisikan satu dimensi
  - N terms mendefinisikan ruang berdimesi N (N-dimensional space)
  - Element dari vector dipetakan ke bobot term (term weight)
  - E.g., d = $(x_1,\ldots,x_N)$, x_i is “importance” of term i
- New document is assigned to the most likely category based on vector similarity.

Ilustrasi Model Ruang Vektor



Yang tidak dispesifikasikan oleh Model Ruang Vektor

- How to select terms to capture “basic concepts”
  - Word stopping
    - e.g. “a”, “the”, “always”, “along”
  - Word stemming
    - e.g. “computer”, “computing”, “computerize” => “compute”
  - Latent semantic indexing
- How to assign weights
  - Not all words are equally important: Some are more indicative than others
    - e.g. “algebra” vs. “science”
- How to measure the similarity

Cara Mengukur Similaritas

- Given two document

$$D_i = (w_{i1}, w_{i2}, \cdots, w_{iN}) \qquad D_j = (w_{j1}, w_{j2}, \cdots, w_{jN})$$

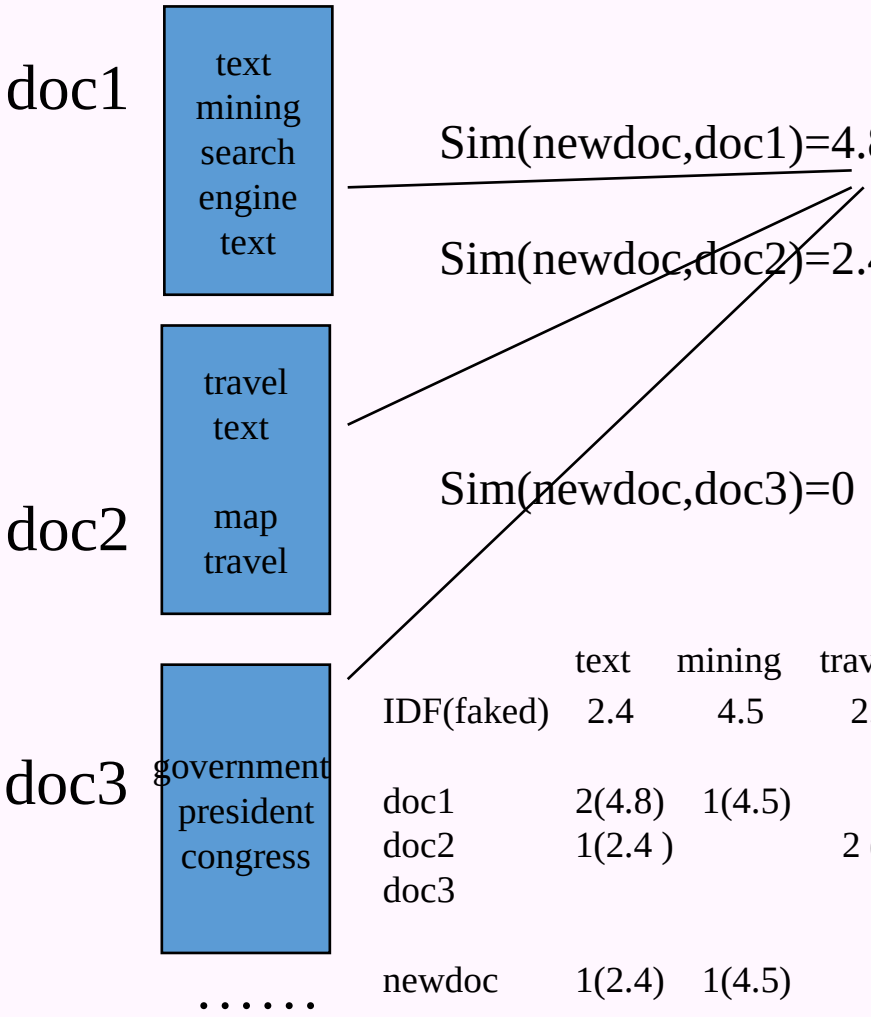
- Similarity definition
  - dot product

$$Sim(D_i, D_j) = \sum_{t=i}^{N} w_{it} * w_{jt}$$

  - normalized dot product (or cosine)

$$Sim(D_i, D_j) = \frac{\sum_{t=i}^{N} w_{it} * w_{jt}}{\sqrt{\sum_{t=1}^{N} (w_{it})^2 * \sum_{t=1}^{N} (w_{jt})^2}}$$

Contoh Ilustrasi



	text	mining	travel	map	search	engine	govern	president	congress
IDF(faked)	2.4	4.5	2.8	3.3	2.1	5.4	2.2	3.2	4.3
doc1	2(4.8)	1(4.5)			1(2.1)	1(5.4)			
doc2	1(2.4)		2 (5.6)	1(3.3)					
doc3							1 (2.2)	1(3.2)	1(4.3)
newdoc	1(2.4)	1(4.5)							

Analisis sintak, topic modelling & sosmed mining

Topic Model

- Mencari pola topik tersembunyi dalam koleksi dokumen menggunakan metode statistika
- Menganotasi dokumen dgn topik-topik yang ditemukan tsb
- Memanfaatk anotasi topik utk mengelola, menyimpulkan , mencari teks ...

Contoh Topic

Topic 247

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

Topic 5

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

Topic 43

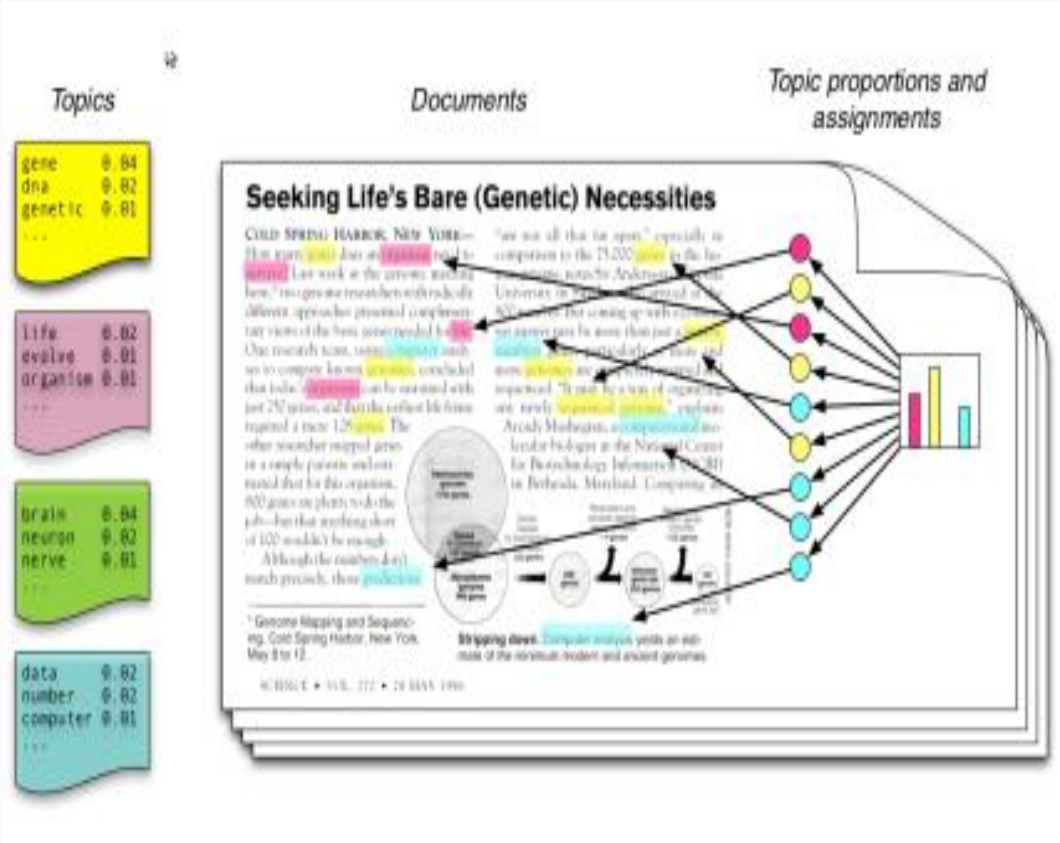
word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THINK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

Topic 56

word	prob.
DOCTOR	.074
DR.	.063
PATIENT	.061
HOSPITAL	.049
CARE	.046
MEDICAL	.042
NURSE	.031
PATIENTS	.029
DOCTORS	.028
HEALTH	.025
MEDICINE	.017
NURSING	.017
DENTAL	.015
NURSES	.013
PHYSICIAN	.012
HOSPITALS	.011

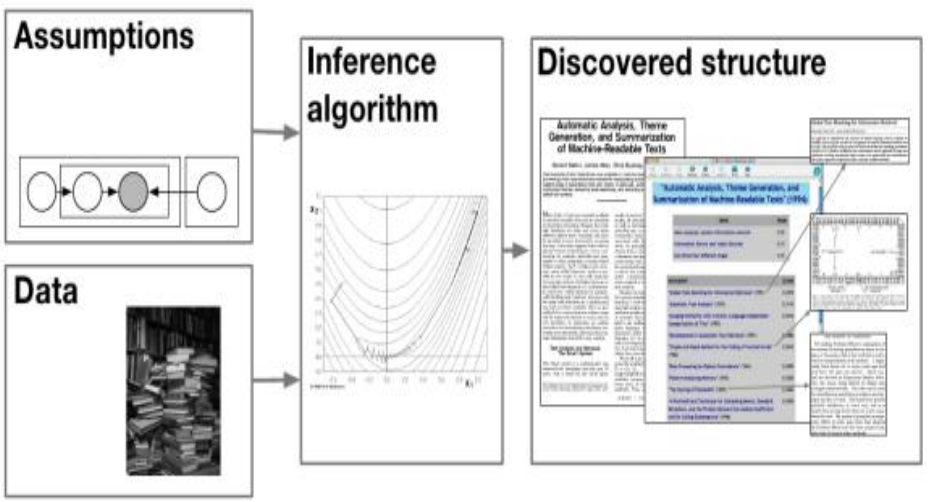
Figure 1. An illustration of four (out of 300) topics extracted from the TASA corpus.

Topic Model atas Dokumen



Topic Model: Struktur

- Observasi: koleksi teks
- Asumsi: teks telah digenerate menurut ke suatu/beberapa model
- Output: model yang telah menGenerate teks



Conto hTopic

SKY WATER TREE
MOUNTAIN PEOPLE

SCOTLAND WATER
FLOWER HILLS TREE

SKY WATER BUILDING
PEOPLE WATER

FISH WATER OCEAN
TREE CORAL

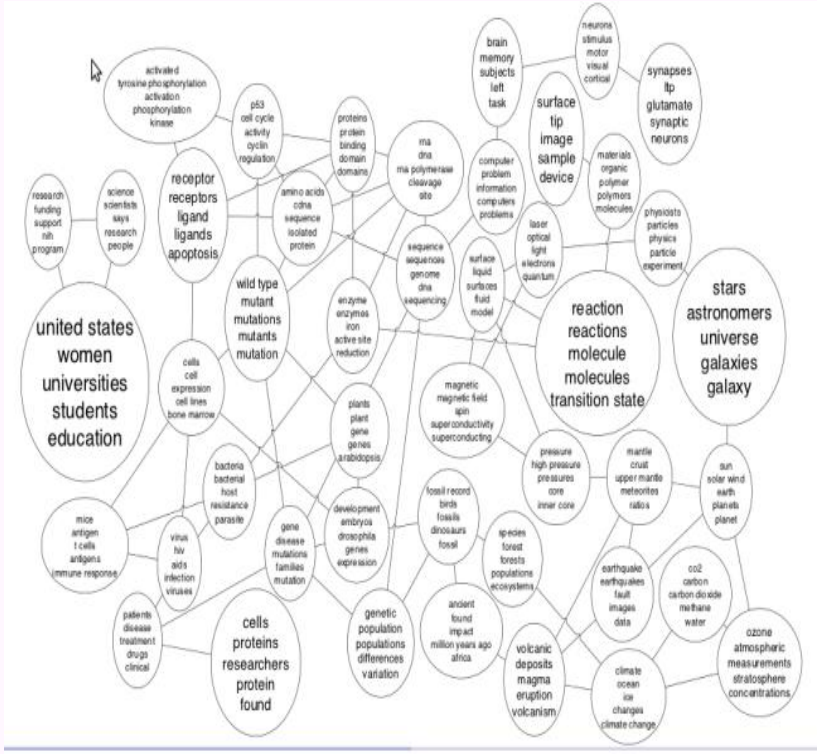
PEOPLE MARKET PATTERN
TEXTILE DISPLAY

BIRDS NEST TREE
BRANCH LEAVES

Object ≡ bag of words with labels

Contoh Topic

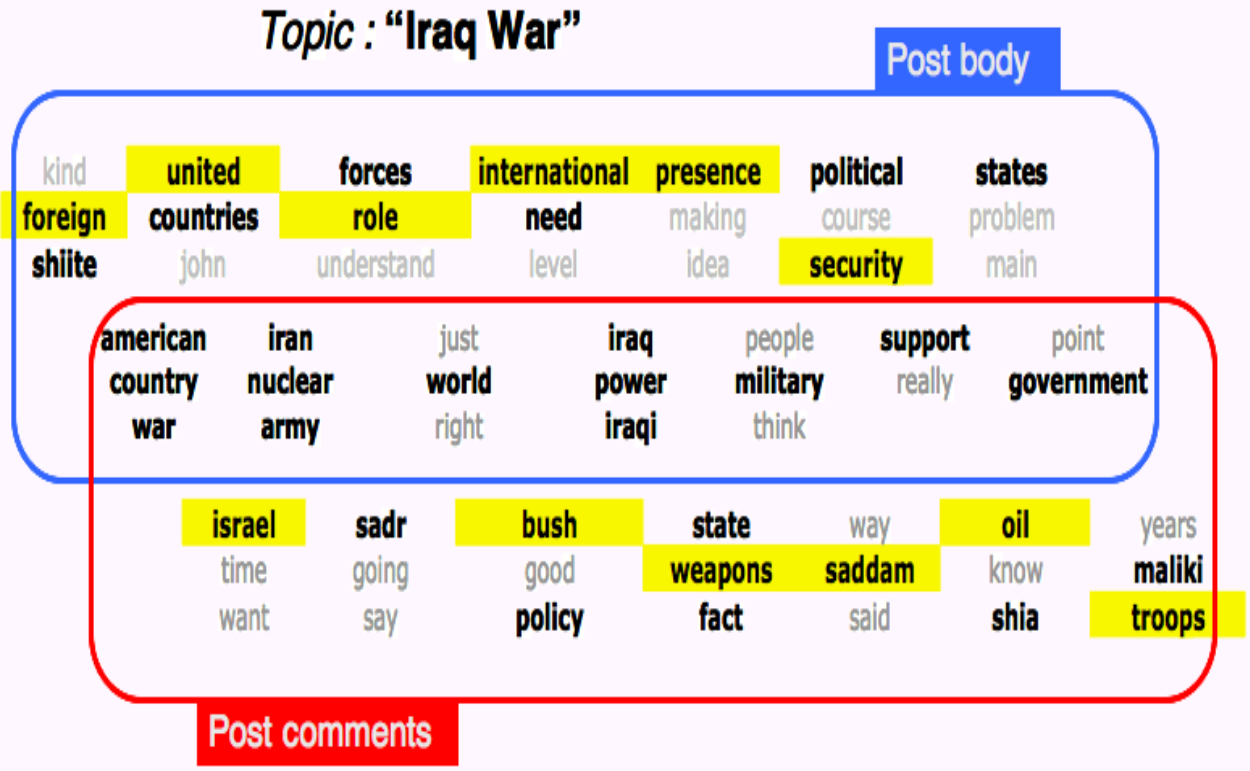
- Komponen dasar:
  - himpunan entitas (mis. dokumen, image, individual, gen)
  - himpunan relasi (mis. sitasi/kutipan/referensi, co-tag, pertemanan)



Topic Discovery

Finding related, but not obvious words unique to this corpus

Topic Discovery



The blogger discusses strategy, comments focus on "tangibles"

Topic Model dgn LDA

Pengantar LDA

- Ilustrasi LDA:

Misal ada himpunan kalimat:

(1) I like to eat broccoli and bananas.

(2) I ate a banana and spinach smoothie for breakfast.

(3) Chinchillas and kittens are cute.

(4) My sister adopted a kitten yesterday.

(5) Look at this cute hamster munching on a piece of broccoli.

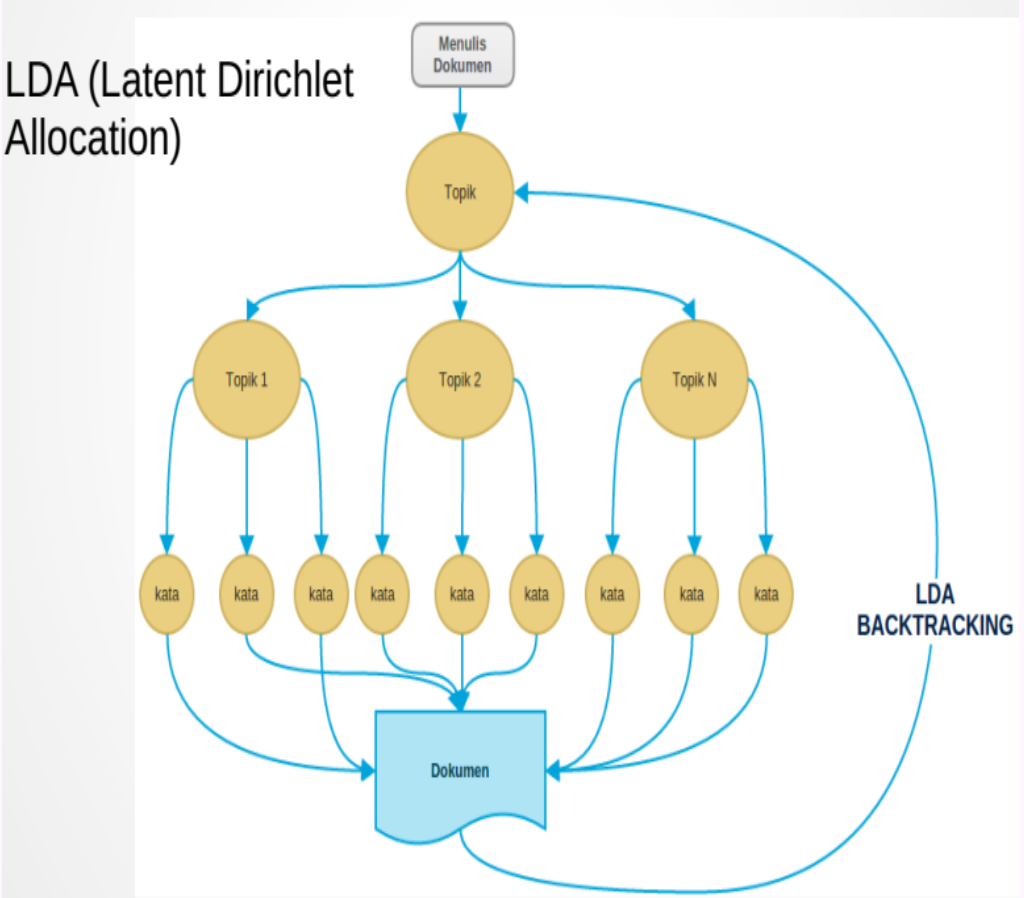
Apa yang dimaksud Latent Dirichlet allocation? Yakni cara untuk menemukan topik secara otomatis yang dimiliki kalimat. Sebagai contoh, diberikan kalimat tsbt dan ditanya tentang 2 topik, LDA akan menghasilkan luaran sbb:

- Kalimat 1 and 2: 100% Topic A
- Kalimat 3 and 4: 100% Topic B
- Kalimat 5: 60% Topic A, 40% Topic B
- Topic A: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, ... (dapat diintrepretasikan bahwa topic a tentang **makanan**)
- Topic B: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, ... (dapat diintrepretasikan bahwa topic B to tentang **hewan imut-imut**)

sumber: http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/

Pengantar LDA

- LDA merepresentasi dokumen sebagai gabungan topik-topik dari kumpulan kata-kata dengan probabilitas tertentu. Jika kita menulis dokumen:



Perhitungan dan Notasi LDA

Notasi:

- Kata: unit dasar suatu data diskrit, didefinisikan sebagai item dari kosakata berindeks {*1, ..., V*}
- Dokumen: sekuen kata *N*, *w* = (w_1, w_2, ..., w_N), kata ke-*n* dalam sekuen
- Korpus: koleksi dokumen *M*, *D* = (w_1, w_2, ..., w_M)

LDA berasumsi proses generatif utk setiap dokumen *w* dalam korpus *D*, sbb:

- Pilih N ~ Poisson(ξ)
- Pilih ~ Dir(α)
- Utk setiap N dari kata w_n:
  - Pilih topik z_n~Multinomial (θ)
  - Pilih sebuah kata wn dari p(w_n |z_n,β), probabilitas multinomial bersyarat dalam topic z_n.

Komparasi Topic

- Topik Model yang digenerasi selanjutnya perlu dibandingkan dengan penilaian pakar
- Hasil penilaian pakar menjadi training set baru sebagai data baru utk proses machine learning

Contoh output perhitungan topic model dgn LDA

NO. TEMA	VERBATIM CCD	TEMA/LABEL/ANOTASI	TOPIK MODEL
1	"Tapi sebelum kita mulai wawancaranya mbak, bisa tolong di perkenalkan dulu nama mbak?" "perkenalkan nama saya Sheba, umur saya 27 tahun, saya ibu rumah tangga." (mengatur posisi duduk)	Latar belakang significant other	topik 1: 1.000*perkenalkan

NO. TEMA	VERBATIM CCD	TEMA/LABEL/ANOTASI	TOPIK MODEL
2	"subjek kalau dirumah seperti apa mbak?" "dia kalau dirumah biasa aja, dia berangkat kerja pagi pulang malam, dia juga di rumah blom lama, karena sebelumnya dia ngekost, kalau kerumah paling cuma satu minggu sekali atau ngak kalau lagi ada tanggal merah aja, kalau malem pulang kerja saya jarang ketemu dia, karena saya pas jam dia pulang juga udah masuk kamar." "jadi mbak jarang ketemu subjek kalau udah malam?" "Iya jarang, itu juga kalau sekali-sekali aja kalau saya kebetulan tidur agak malem, kalau ketemu lagi paling pagi bareng suami saya kerja"	Pendapat significant other mengenai subjek	topik 1: 0.156*ketemu + 0.151*pulang + 0.145*kerja + 0.142*malem + 0.139*pagi + 0.138*malam + 0.130*dirumah topik 2: 0.174*kerja + 0.172*pulang + 0.170*ketemu + 0.123*dirumah + 0.121*malam + 0.120*pagi + 0.119*malem

Contoh output perhitungan topic model dgn LDA

NO. TEMA	VERBATIM CCD	TEMA/LABEL/ANOTASI	TOPIK MODEL
3	"subjek pernah cerita ngak soal kerjaan dia di kantor sama mbak?" "Pernah sekali-sekali dia cerita tentang kerjaannya, biasaya tentang personal karyawan disitu, misalnya ada sesuatu yang lucu tentang temen kantornya yang kalau ngomong pake bahasa inggris kurang ngak jelas tapi kalau masalah kerjaan, paling masalah kerjaannya yang banyak aja."	Subjek bercerita mengenai pekerjaannya significant other	topik 1: 0.250*masalah + 0.250*kerjaannya + 0.250*kerjaan + 0.250*cerita

NO. TEMA	VERBATIM CCD	TEMA/LABEL/ANOTASI	TOPIK MODEL
4	"kalau di rumah subjek pernah ngak menunjukkan gejala stres kerja? "Stres kerja ya, paling kayaknya dia males berdesak-desakan aja kalau berangkat dan pulang kerja di kereta, jadi stres kerja yang keliatan dari dia dia keliatan kecapeaan."	Stres kerja yang subjek tunjukkan di depan singnificant other	topik 1: 0.333*stres +0.333*kerja + 0.333*keliatan

Analisis Sentimen

Analisis Sentimen

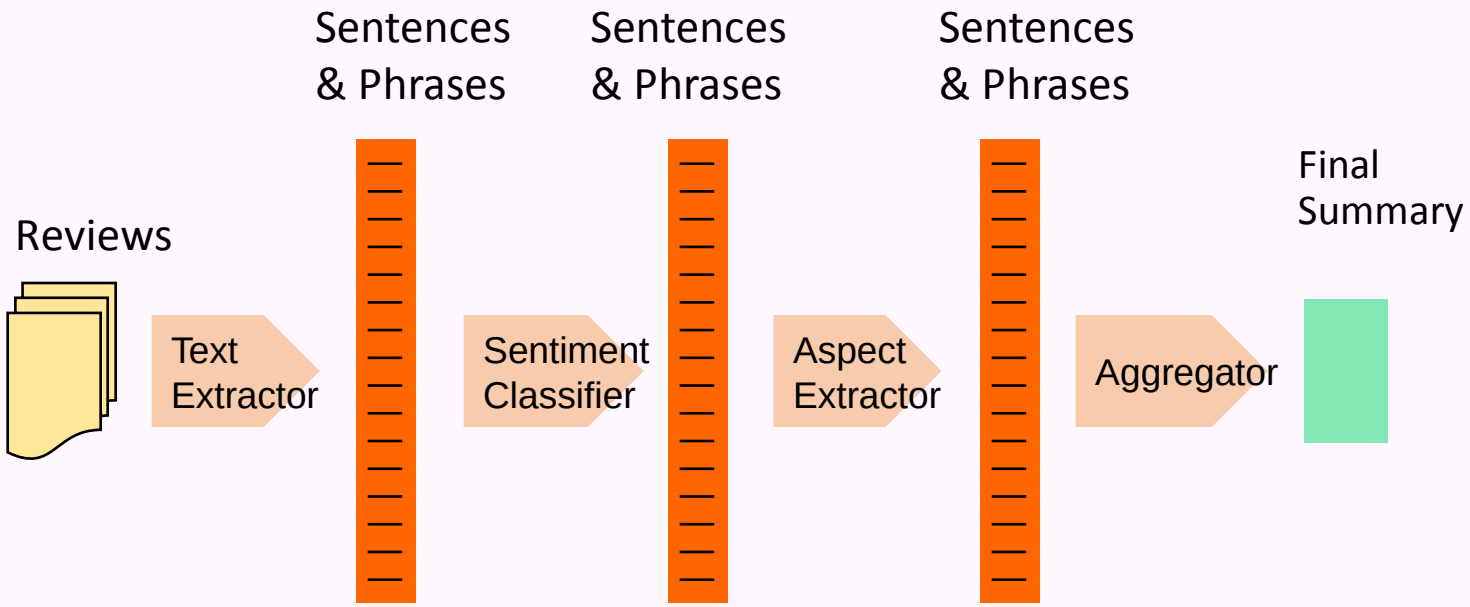
- Is the attitude of this text positive or negative?

- More complex:
  - Rank the attitude of this text from 1 to 5
- Advanced:
  - Detect the target, source, or complex attitude types

Tipologi Keadaan/Kondis Afektif *Scherer*

- **Emotion**: brief organically synchronized … evaluation of a major event
  - *angry, sad, joyful, fearful, ashamed, proud, elated*
- **Mood**: diffuse non-caused low-intensity long-duration change in subjective feeling
  - *cheerful, gloomy, irritable, listless, depressed, buoyant*
- **Interpersonal stances**: affective stance toward another person in a specific interaction
  - *friendly, flirtatious, distant, cold, warm, supportive, contemptuous*
- **Attitudes: enduring, affectively colored beliefs, dispositions towards objects or persons**
  - *liking, loving, hating, valuing, desiring*
- **Personality traits**: stable personality dispositions and typical behavior tendencies
  - *nervous, anxious, reckless, morose, hostile, jealous*

Tahapan Analisis Sentimen



Computational work on other affective states

- **Emotion**:
  - Detecting annoyed callers to dialogue system
  - Detecting confused/frustrated versus confident students
- **Mood**:
  - Finding traumatized or depressed writers
- **Interpersonal stances**:
  - Detection of flirtation or friendliness in conversations
- **Personality traits**:
  - Detection of extroverts

Detection of Friendliness

- Friendly speakers use collaborative conversational style
  - Laughter
  - Less use of negative emotional words
  - More sympathy
    -
  - More agreement
    -
  - Less hedges
    -

TERIMA KASIH