Pemrosesan Bahasa Alami

Perkuliahan #5
POSTagging

Tim Dosen Pemrosesan Bahasa Alami
Prodi S1 Informatika

Referensi

- Materi dari Andrew McCallum & Kathy McKeown
- Materi dari John Longley, School of Informatics, University of Edinburgh
- Speech and Languange Processing, Dan Jurafsky
- Indonesian Rule-based POSTagger, Rashel et.al, 2014
- Slide dari Natalie Parde, http://www.natalieparde.com/teaching/cs_421_fall2019/Part-of-Speech%20Tagging%20and%20Formal%20Grammars.pdf

Kerangka Bahasan

- Definisi *Part of Speech* (POS)
- Aplikasi *POS*
- Definisi *task POS Tagging*
- Pendekatan untuk menyelesaikan *POS Tagging*
- Evaluasi *POS Tagging*

Apa itu *POS Tag*?

- *Part of Speech* (POS), umum disebut juga sebagai kelas kata, atau kategori sintaktik
- Sebuah kategori yang diberikan ke sebuah kata sesuai dengan fungsi sintaktiknya
- Contoh:
  - *Noun*/kata benda: *people, animal, things*, kucing, pohon
  - *Verb*/kata kerja: *run, study*, makan, bekerja
  - *Adjective*/kata sifat: *soft, diligently*, pintar, berat
- *Part of Speech* memberikan informasi tambahan untuk sebuah kata dan tetangga-tetangganya
  - Contoh: sebuah kata benda biasanya diikuti dengan kata sifat atau kata keterangan (sesuai dengan aturan Bahasa Indonesia)

Apa yang dimaksud dengan kelas kata?

Kumpulan kata-kata yang mempunyai karakteristik mirip:

- Muncul di konteks yang serupa atau mirip
- Mempunyai fungsi sintaktik yang sama dalam kalimat
- Mengalami transformasi yang serupa

Kelas kata umum/tradisional:

noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction

Definisi PoS dahulu kala

- Dionysius Thrax dari Alexandria (100 B.C.) mendefinisikan 8 PoS :
  - Noun
  - Verb
  - Pronoun
  - Preposition
  - Adverb
  - Conjunction
  - Participle
  - Article

Perkembangan PoS saat ini

- 45 tag di Penn Treebank dataset
- 87 tag di Brown Corpus
- 146 tag di C7 tagset
- Tag spesifik untuk teks Bahasa Indonesia?

Contoh *POS Tag* dan deskripsinya

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb

Number	Tag	Description
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Contoh kata-kata yang termasuk dalam POS Tag tertentu

- Noun : book/books, nature, Germany, Sony
- Verb : eat, wrote
- Auxiliary : can, should, have
- Adjective : new, newer, newest
- Adverb : well, urgently
- Numbers : 872, two, first
- Article/Determiner : the, some
- Conjunction : and, or
- Pronoun : he, my
- Preposition : to, in
- Particle : off, up
- Interjection : Ow, Eh

Konvensi POS Tag Bahasa Indonesia

- Referensi: http://inacl.id/inacl/wp-content/uploads/2017/06/INACL-POS-Tagging-Convention-26-Mei.pdf
- Terdapat dua kategori kata:
  - Kata konten: kata yang mempunyai makna leksikal (kita bisa menemukan maknanya di kamus, seperti: KBBI, *Oxford English Dictionary*). Kelas utama: nomina, verba, adjektiva, dan adverbia
  - Kata fungsi: kata yang menunjukkan hubungan antar konsep di dalam sebuah kalimat. Contoh kelas: konjungsi, preposisi, interjeksi, determina

Universal POS Tag

- Kelas/POS utama:
  - *open class*: ADJ, ADV, INT, NOUN, PROPN, VERB
  - *closed class*: ADP, AUX, CCONJ, DET, NUM, PART, PRON, SCONJ
- POS lainnya: PUNCT, SYM, X

Sumber: https://universaldependencies.org/u/pos/

Closed vs. Open Class

- *Closed class*: kata-kata yang masuk ke dalam kelas tersebut relatif tetap
  - Prepositions: of, in, by, …
  - Auxiliaries: may, can, will, had, been, …
  - Pronouns: I, you, she, mine, his, them, …
  - Biasanya kata fungsi
- *Open class*: anggota kelas kata tersebut berubah secara dinamis (terutama bertambah)
  - Nouns, Verbs, Adjectives, Adverbs
    - Nouns (Proper Nouns) -> Telkom University, Soekarno
    - Verbs -> Please **email** me the invitation

Contoh pemanfaatan informasi *POS Tag*: pemeriksaan tata bahasa (*grammar checking*)

- Perhatikan dua contoh kalimat berikut:
  - Dia pergi ke toko buku kemarin
  - Saya pergi ke pasar hari ini

Pertanyaan: Apakah pemeriksaan tersebut juga dapat dilakukan berdasar *language model*? Apa bedanya dengan menggunakan informasi *POS Tag*?

Contoh pemanfaatan informasi *POS Tag*: ekstraksi informasi

- Identifikasi named entity orang/*Person* dan organisasi/*Organization* berdasarkan *POS proper noun*
- …… coba cari contoh lainnya!

Contoh pemanfaatan informasi *POS Tag*: mesin penerjemah

- Give me a **round** figure. [adjective]
- Shall we play another **round** of cards? [noun]
- He had a look **round** before he kept going. [adverb]
- They walked **round** the tree. [preposition]
- The floor function **rounds** down. [verb]

Contoh pemanfaatan informasi *POS Tag*: parafrase dan peringkasan

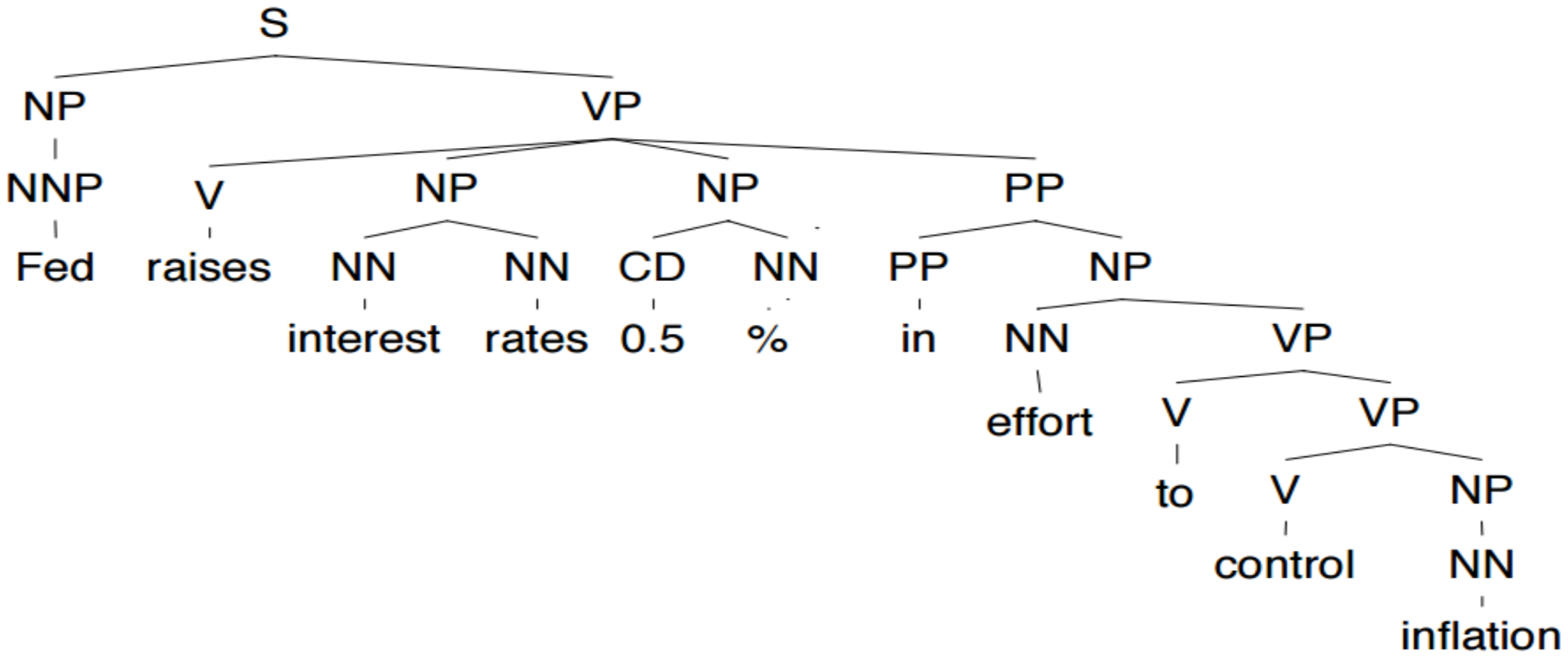
- The **well** was drilled fifty meters deep
- All is **well** with us

- The
  so [conjunction]
  spring [noun]
  grow [verb]
  okay [adjective]
  successfully [adverb]
  anyway [interjection]
  was drilled fifty meters deep
- All is
  so [conjunction]
  spring [noun]
  grow [verb]
  okay [adjective]
  successfully [adverb]
  anyway [interjection]
  with us

Contoh pemanfaatan informasi *POS Tag*: *speech synthesis*

- Contoh pertama:
  - They **live/verb** in Bandung
  - Opening PON is **live/adj** on TV
- Contoh kedua:
  - Eggs have a high protein **content/Noun**
  - She was **content/Verb** to step down after four years as CEO

Contoh pemanfaatan informasi *POS Tag*: *parsing*



POS Tagging

Apa itu *POS Tagging*?

Pemberian *POS Tag* untuk tiap kata dalam sebuah kalimat.

Tantangan: ambiguitas (sebuah kata dapat memiliki > 1 POS)

Contoh (*tagging* berdasar http://bahasa.cs.ui.ac.id/postag/tagger):

- Bisa/NN ular/NN sangat/RB berbahaya/VB
- Saya/NN tidak/NEG bisa/MD pergi/VB kemarin/NN

Task POS Tagging

Diberikan sebuah kalimat yang terdiri atas rangkaian kata-kata, akan dihasilkan label POS/*POS Tag* untuk tiap kata dalam kalimat tersebut.

Contoh masukan:

Pemerintah kota Delhi mengerahkan monyet untuk mengusir monyet-monyet lain yang berbadan lebih kecil dari arena Pesta Olahraga Persemakmuran.

Contoh keluaran:

Pemerintah/NNP kota/NNP Delhi/NNP mengerahkan/VB monyet/NN untuk/SC mengusir/VB monyet-monyet/NN lain/JJ yang/SC berbadan/VB lebih/RB kecil/JJ dari/IN arena/NN Pesta Olahraga/NNP Persemakmuran/NNP ./Z

Seberapa banyak kasus ambiguitas *POS Tag*?

- 45-tags Brown corpus (word types)
  - Unambiguous (1 tag): 38,857
  - Ambiguous: 8,844
    - 2 tags: 6,731
    - 3 tags: 1,621
    - 4 tags: 357
    - 5 tags: 90
    - 6 tags: 32
    - 7 tags: 6 (well, set, round, open, fit, down)
    - 8 tags: 4 ('s, half, back, a)
    - 9 tags: 3 (that, more, in)

Seberapa sulit *task POS Tagging*?

- Beberapa kata hanya mempunyai 1 POSTag (e.g. adalah, Budi, sangat, terbesar)
- Kata-kata lain mempunyai 1 POSTag yang sangat umum ditemui (missal: dog -> photographers seemed to dog her every step ???)
- Dengan metode *state-of-the-art*, seberapa akurat POSTagger saat ini?
  - Kurang lebih 97% (catatan: POSTagger Bahasa Inggris)
- Penggunaan metode *baseline* dapat menghasilkan akurasi 90%
- Pendekatan *baseline*:
  - Tag tiap kata dengan *most-frequent tag* kata tersebut (dari data latih)
  - Tag *unknown words* dengan tag yang paling sering muncul (biasanya *Noun*)

Pendekatan dalam *POS Tagging*

Pendekatan dalam *POS Tagging*

- Secara umum terdapat tiga pendekatan
  - Berbasis aturan / *rule-based*
  - Berbasis statistik
  - Kombinasi berbasis aturan dan statistik (*transformation-based tagger*)
- Derajat supervisi:
  - *Supervised*
  - *Unsupervised*
  - *Partly supervised,* misal*:* data latih tidak dilabeli manual, tetapi diberi label berdasar informasi kamus

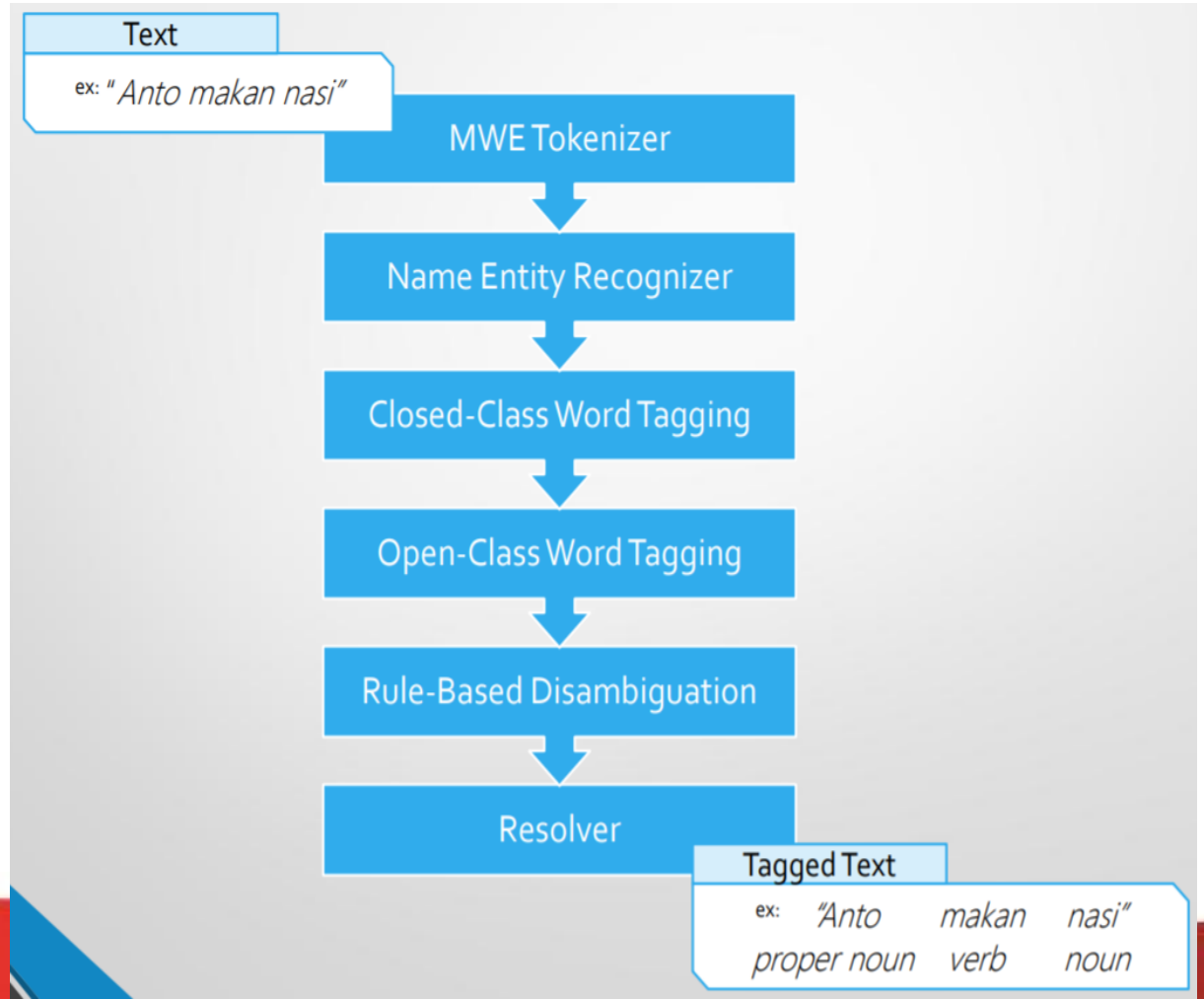
POS Tagging berbasis aturan

- Biasanya pemberian tag diawali dengan keberadaan kamus dan daftar tag yang telah terdefinisi
- Inisialisasi *tagging* berdasar informasi dari kamus
- Pendefinisian aturan secara manual untuk menyaring tag
- Proses pemeriksaan berdasarkan aturan berhenti saat untuk satu kata hanya ada 1 tag

Contoh *POSTagger* yang menerapkan pendekatan berbasis aturan:

http://bahasa.cs.ui.ac.id/postag/tagger

Gambaran umum proses pada *POSTagger* Bahasa Indonesia berbasis aturan



POS Tagging berbasis statistik

- Membutuhkan dataset berlabel
- Secara umum terdapat dua pendekatan:
  - Klasifikasi non-sekuensial
  - Klasifikasi sekuensial

Anotasi/Pelabelan Korpus untuk *POSTagging*

- *Corpus*/korpus: koleksi teks digital yang dituliskan dalam bahasa alami, digunakan sebagai sumber informasi bahasa
- Proses pelabelan korpus untuk data latih *POSTagging* dilakukan pada level token/kata
- Proses pelabelan 100% manual sangat mahal (dari sisi waktu dan biaya), sehingga umumnya dilakukan kombinasi: pelabelan awal secara heuristik secara otomatis, kemudian akan diperiksa oleh ahli bahasa sesuai dengan panduan pelabelan

Fitur yang digunakan dalam non-sekuensial *POS Tagging* berbasis statistika

- Berdasar info atribut dari sebuah kata tersebut sendiri, sudah dapat memberikan hasil cukup bagus:
  - *Word* (kata itu sendiri). Misal, kata 'the' dapat memberi informasi POSTag DET
  - Kata yang dinormalisasi menjadi dituliskan dengan huruf kecil semua (*lowercased word*)
  - Prefiks. Misal kata unfathomable mempunyai prefix un, dapat diambil informasi, POSTag-nya = JJ
  - Sufiks. Misal kata importantly, mempunyai sufiks ly, dapat diambil informasi, POSTag-nya = RB
  - *Capitalization*. Apakah sebuah kata dituliskan dengan huruf awal kapital. Jika iya, kandidat POSTag = NNP
  - Bentuk penulisan, misal apakah mengandung karakter '-', jika iya, kemungkinan besar POSTag = JJ

Fitur tambahan untuk meningkatkan akurasi *POSTagging*

- Menambahkan fitur kata berikutnya:

			RB				
PRP	VBD	IN	RB	IN	PRP	VBD	
They	left	as	soon	as	he	arrived	.

Token ‘as’ merupakan satu rangkaian frase ‘as soon as’. Dengan melihat keberadaan token selanjutnya, ‘soon’, prediksi dapat lebih akurat.

Diskusi

Kelebihan dan kekurangan:

- *rule-based POSTagging*
- Non-sekuensial *supervised POSTagging*

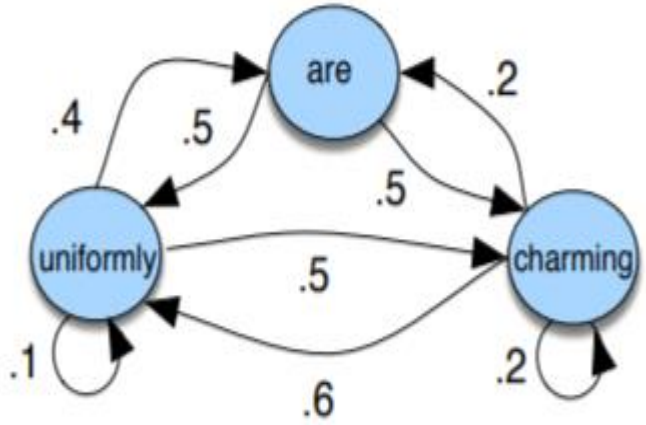
Hidden Markov Model (HMM) POS *Tagger*

Model *Hidden Markov* (HMM)

- Merupakan sebuah model sekuensial
  - Tujuan yang ingin dicapai dengan model sekuensial adalah pemberian label (POS Tag) untuk sebuah sekuens kata. Pemetaan dari masukan berupa sekuens kata ke sekuens label POS Tag.
- Model HMM berdasarkan Markov *chain*
  - Markov *chain*: memodelkan probabilitas sekuens variabel acak, yaitu *state*
  - Asumsi Markov *chain*: untuk dapat mempredikasi apa yang akan terjadi selanjutnya (*next state*), yang berpengaruh adalah hanya *current state*

$$P(q_i = a|q_1 \ldots q_{i-1}) = P(q_i = a|q_{i-1})$$

Contoh *Markov Chain* untuk Sekuens Kata



Mengingat kembali model bahasa bigram, lihat nilai *probability* di sisi diagram tersebut, menunjukkan $P(w_i|w_{i-1})$!

Komponen *Markov Chain*

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

HMM untuk *POS Tagging*

- *Observed events*: kata
- *Hidden events*: POS Tag

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state q_i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Asumsi pada *first-order* HMM:

- Seperti pada asumsi Markov *chain*, *probability* pada sebuah *state* hanya bergantung pada *state* sebelumnya (*previous state*).

transition probability

- *Probability output*
- *observation* hanya bergantung pada *state* yang memproduksi *observation*, tidak terkait dengan *state* atau observasi lain.

emission probability

Komponen *Tagger* HMM

- Matriks probability A / *transition probability*, $P(t_i \mid t_{i-1})$

$$P(t_i \mid t_{i-1}) = \frac{count(ti_{-1}, ti)}{count(ti_{-1})}$$

- Matriks probability B / *emission probability,* P(wi |ti)

$$P(w_i \mid t_i) = \frac{count(ti,wi)}{count(ti)}$$

Task decoding

- Definisi *decoding*:

Diberikan masukan sebuah model HMM (matriks A dan B) dan sebuah sekuens observasi/kata O = o_1, o_2, …, o_T , tentukan sekuens tag dengan *probability* paling tinggi Q = q_1,q_2, …, q_T

- *Goal*: $\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$

Algoritma Viterbi

- Penyelesaian decoding/POS Tagging dengan pemodelan HMM adalah dengan algoritma Viterbi (pendekatan *dynamic programming*)

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*, *path-prob*

create a path probability matrix *viterbi*$[N,T]$
for each state s **from** 1 **to** N **do** ; initialization step
  $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
  $backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
  for each state s **from** 1 **to** N **do**
    $viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$; termination step
$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$; termination step
bestpath ← the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time
return *bestpath*, *bestpathprob*

Contoh singkat *decoding* dengan Algoritma Viterbi (1)

- Diketahui matriks *transition probability* dan *emission probability* sebagai berikut:

	to N	to V
from start	.8	.2
from N	.4	.6
from V	.8	.2

Transitions

	deal	fail	talks
N	.2	.05	.2
V	.3	.3	.3

Emissions

- Lakukan POS Tagging untuk kalimat '*deal talks fail*'

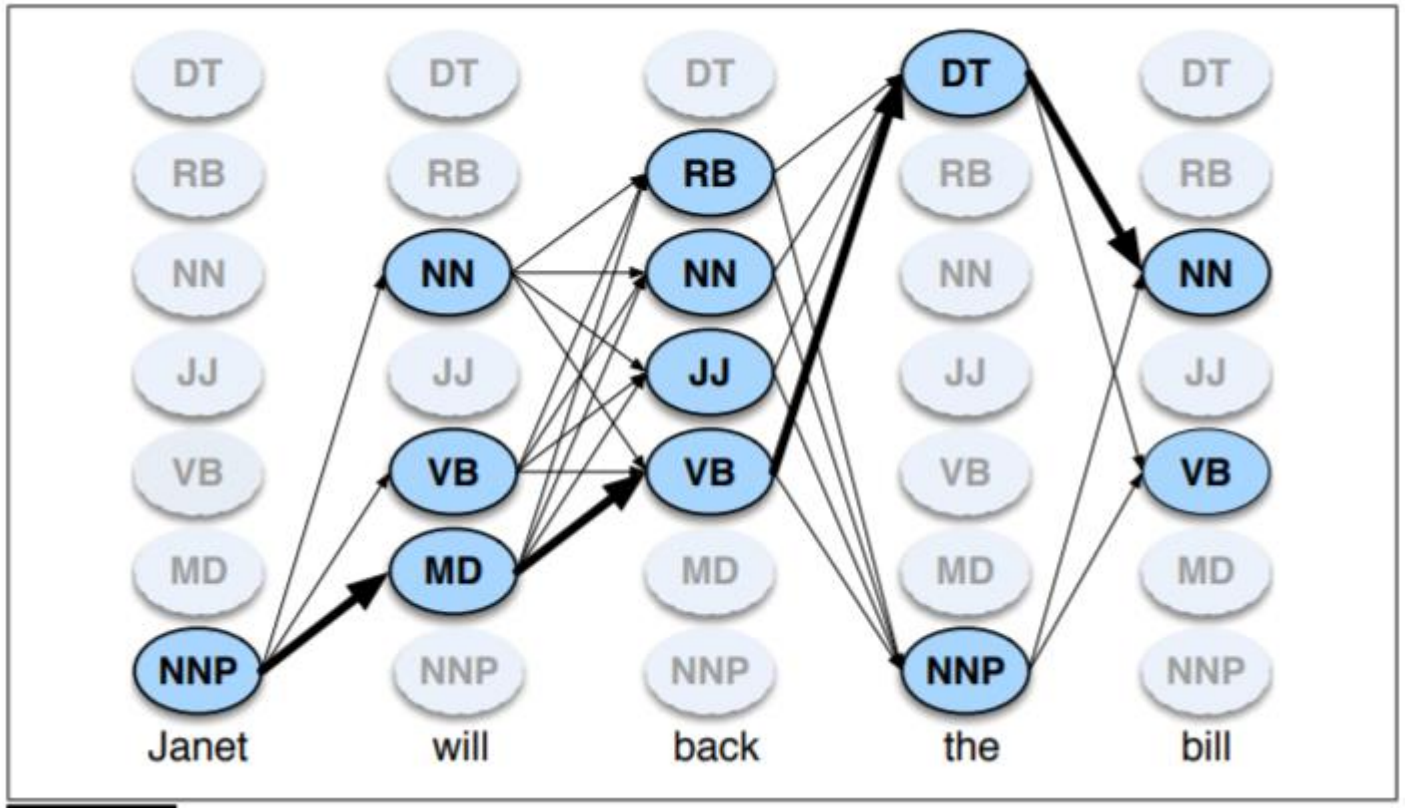
Contoh singkat *decoding* dengan Algoritma Viterbi (2)

- Matriks Viterbi hasil perhitungan:

	deal	talks	fail
N	.8x.2 = .16	← .16x.4x.2 = .0128 (since .16x.4 > .06x.8)	↙ .0288x.8x.05 = .001152 (since .0128x.4 < 0.0288x.8)
V	.2x.3 = .06	↖ .16x.6x.3 = .0288 (since .16x.6 > .06x.2)	↖ .0128x.6x.3 = .002304 (since .0128x.6 > 0.0288x.2)

- Pada kolom paling kanan, nilai *probability* yang paling tinggi adalah pada tag V, kemudian dilakukan *backtracking* sesuai arah *pointer*, maka hasil akhir POS Tagging adalah: N N V

Contoh lain yang lebih kompleks *decoding*/*POS Tagging* dengan Algoritma Viterbi



Lihat buku Speech and Language Processing 3rd edition, chapter 8 halaman 12
https://web.stanford.edu/~jurafsky/slp3/8.pdf

Transformation Based Tagging

Pendahuluan *transformation based tagging*

- Brill *tagging*
- Termasuk dalam *transformation-based learning*
- Cara kerja secara umum adalah dengan mengkombinasikan metode berbasis aturan dan statistika
- Masukan:
  - Korpus latih
  - Kamus dengan informasi *most frequent tags*, yang dibangun dari korpus latih

Cara kerja *transformation based tagger*

- Ide dasar:
  - Inisialisasi dengan memberikan *tag* berdasar *most probable tag* sebuah kata
  - Ubah tag sesuai dengan aturan dalam urutan tertentu
    - Contoh: "jika w_1 adalah *determiner*, dan w_2 adalah *verb*, maka ubah tag w_2 menjadi *noun*"
- Dilakukan proses pembelajaran *rule*/aturan berdasar korpus latih
  - Dari tag inisial, periksa semua kemungkinan transformasi
  - Pilih satu transformasi yang menghasilkan peningkatan akurasi *tagging* paling baik
  - Tag ulang data berdasarkan aturan/rule yang dipilih
  - Ulangi proses pemeriksaan transformasi paling baik dan tag ulang sampai kondisi berhenti
- Catatan: *rule* dapat menghasilkan *error*, yang akan diperbaiki oleh *rule* berikutnya lagi

Contoh *rule*

- Inisialisasi, tagger akan memberikan tag sesuai dengan info most *probable tag,* di mana diketahui P(NN|race) = 0,98 dan P(VB|race)=0,02

Secretariat	is	expected	to	race	tomorrow
NNP	VBZ	VBN	TO	NN	NR

- *Rule* baru dipelajari: jika tag token sebelum adalah TO, maka ubah *tag* kata *race* (yang muncul berikutnya) menjadi VB

- Tag ulang data sesuai *rule* baru

Secretariat	is	expected	to	race	tomorrow
NNP	VBZ	VBN	TO	VB	NR

Pertanyaan: kapan penambahan *rule* dan tagging ulang berhenti?

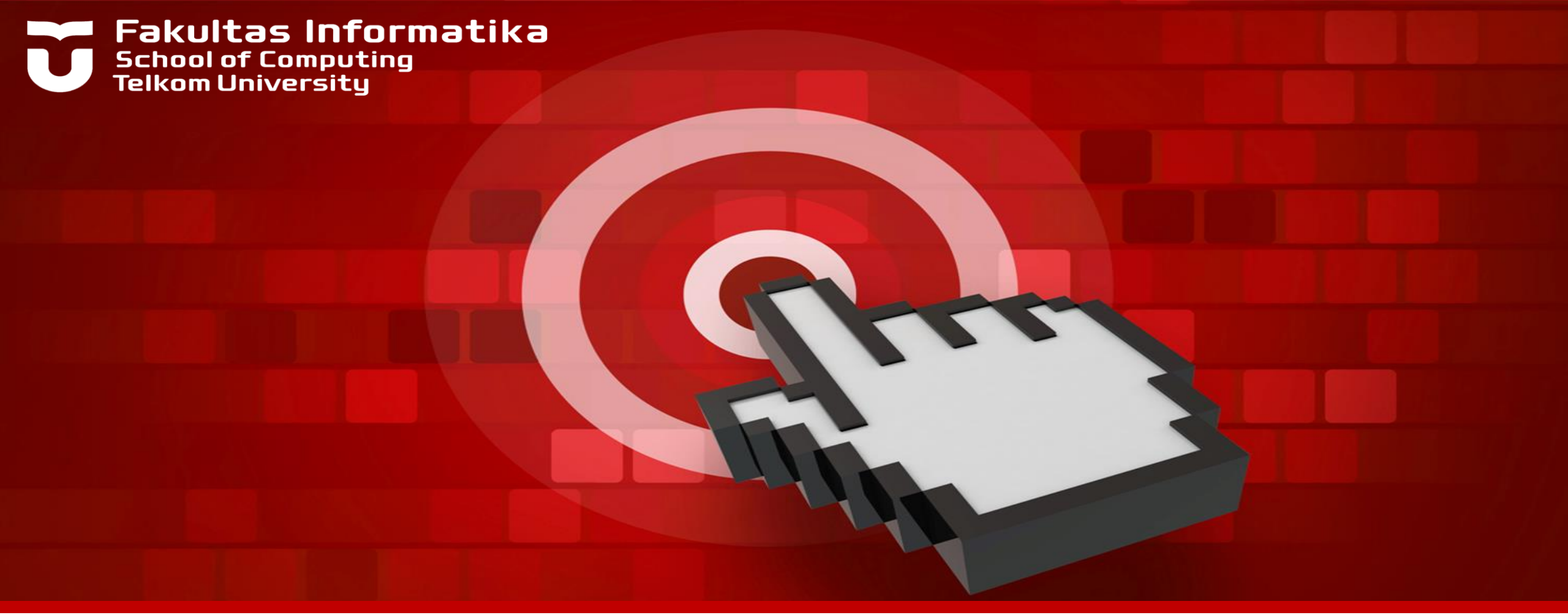
- Secara teori, tidak ada batasan pasti kondisi berhenti penambahan rule pada *Brill tagger*
- Namun, *Brill tagger* mendefinisikan *template* bagi tiap *rule* yang akan ditambahkan:
  - Ubah tag a ke tag b jika tag kata sebelum (atau sesudah) adalah z
  - Ubah tag a ke tag b jika tag dua kata sebelum (atau sesudah) adalah z
  - Ubah tag a ke tag b jika tag satu dari dua kata sebelum (atau sesudah) adalah z
  - Ubah tag a ke tag b jika tag satu dari tiga kata sebelum (atau sesudah) adalah z
  - Ubah tag a ke tag b jika tag kata sebelum adalah z dan tag kata sesudah adalah w
  - Ubah tag a ke tag b jika tag kata sebelum (atau sesudah) adalah z dan tag dua kata sebelum (atau sesudah) adalah w

Diskusi

- Kelebihan dan kekurangan *rule-based tagger* vs *statistical-based tagger*?
- Bagaimana *tagging* untuk *OOV*?

Evaluasi *POS Tagger*

- POS Tagger umumnya dibangun berdasar korpus, disebut sebagai korpus latih
- Pengujian pada data uji, di mana sebelumnya telah dilabeli secara manual oleh manusia -> *gold label dataset*
- Evaluasi: perbandingan *tag* hasil keluaran sistem (POSTagger yang dibangun) dengan *gold label*
- Metriks evaluasi: *precision, recall, F1,* akurasi



THANK YOU