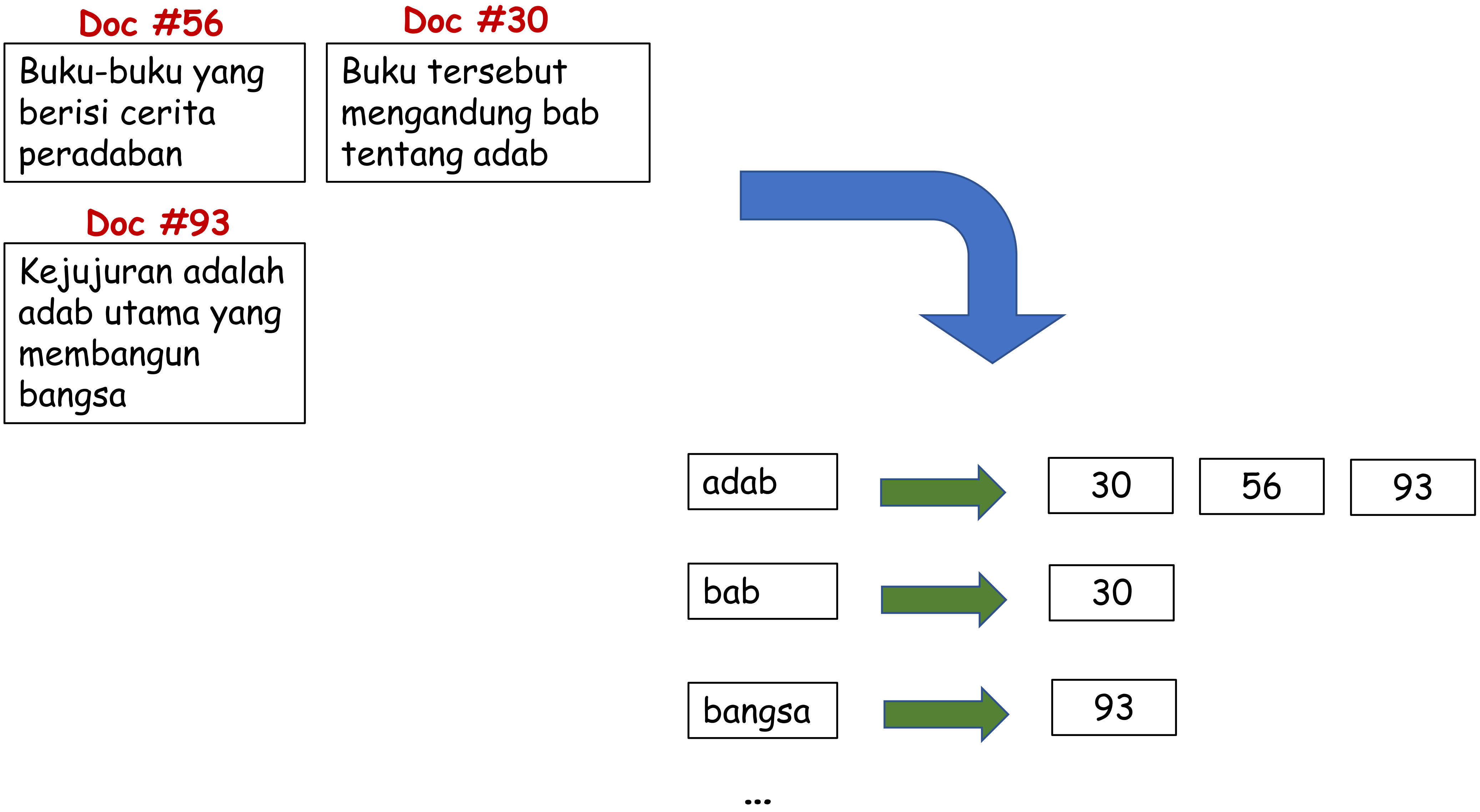
Index Construction

Alfan F. Wicaksono

Fakultas Ilmu Komputer, Universitas Indonesia

A High Level View of Index Construction



Doc #56

Buku-buku yang berisi cerita peradaban

Doc #30

Buku tersebut mengandung bab tentang adab

Doc #93

Kejujuran adalah adab utama yang membangun bangsa

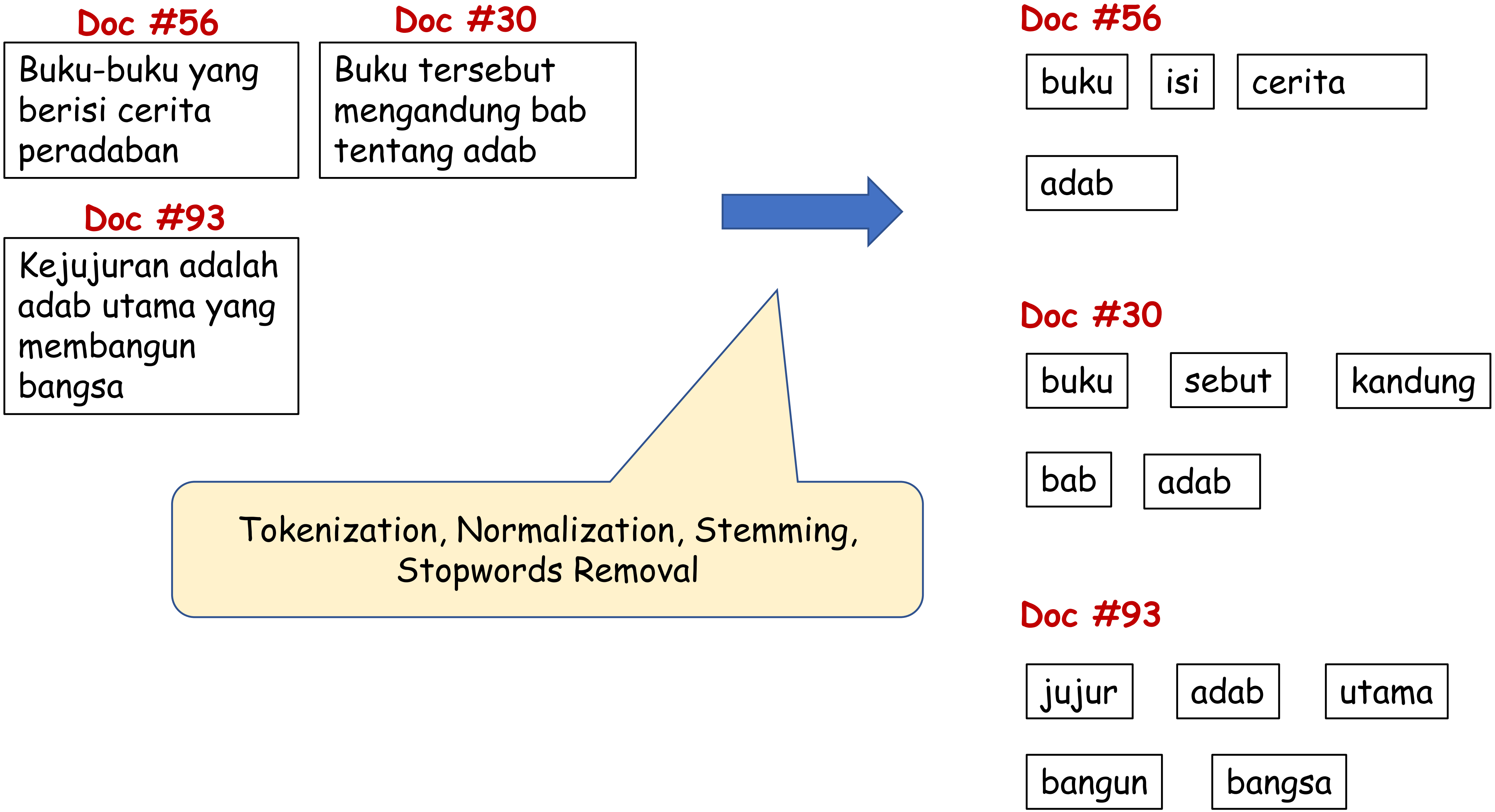
adab → 30 56 93

bab → 30

bangsa → 93

...

A Detail View – Step #1 Tokenization & Linguistic Preprocessing



Doc #56

Buku-buku yang berisi cerita peradaban

Doc #30

Buku tersebut mengandung bab tentang adab

Doc #93

Kejujuran adalah adab utama yang membangun bangsa

Tokenization, Normalization, Stemming, Stopwords Removal

Doc #56

buku | isi | cerita | adab

Doc #30

buku | sebut | kandung | bab | adab

Doc #93

jujur | adab | utama | bangun | bangsa

A Detail View – Step #1 Tokenization & Linguistic Preprocessing

-> Sequence of <Term, docID>

Doc #56

buku isi cerita

adab

Doc #30

buku sebut kandung

bab adab

Doc #93

jujur adab utama

bangun bangsa

Term	Doc ID
buku	30
sebut	30
kandung	30
bab	30
adab	30
buku	56
isi	56
cerita	56
adab	56
jujur	93
adab	93
utama	93
bangun	93
bangsa	93

A Detail View – Step #2 Sorting the Sequence of Terms

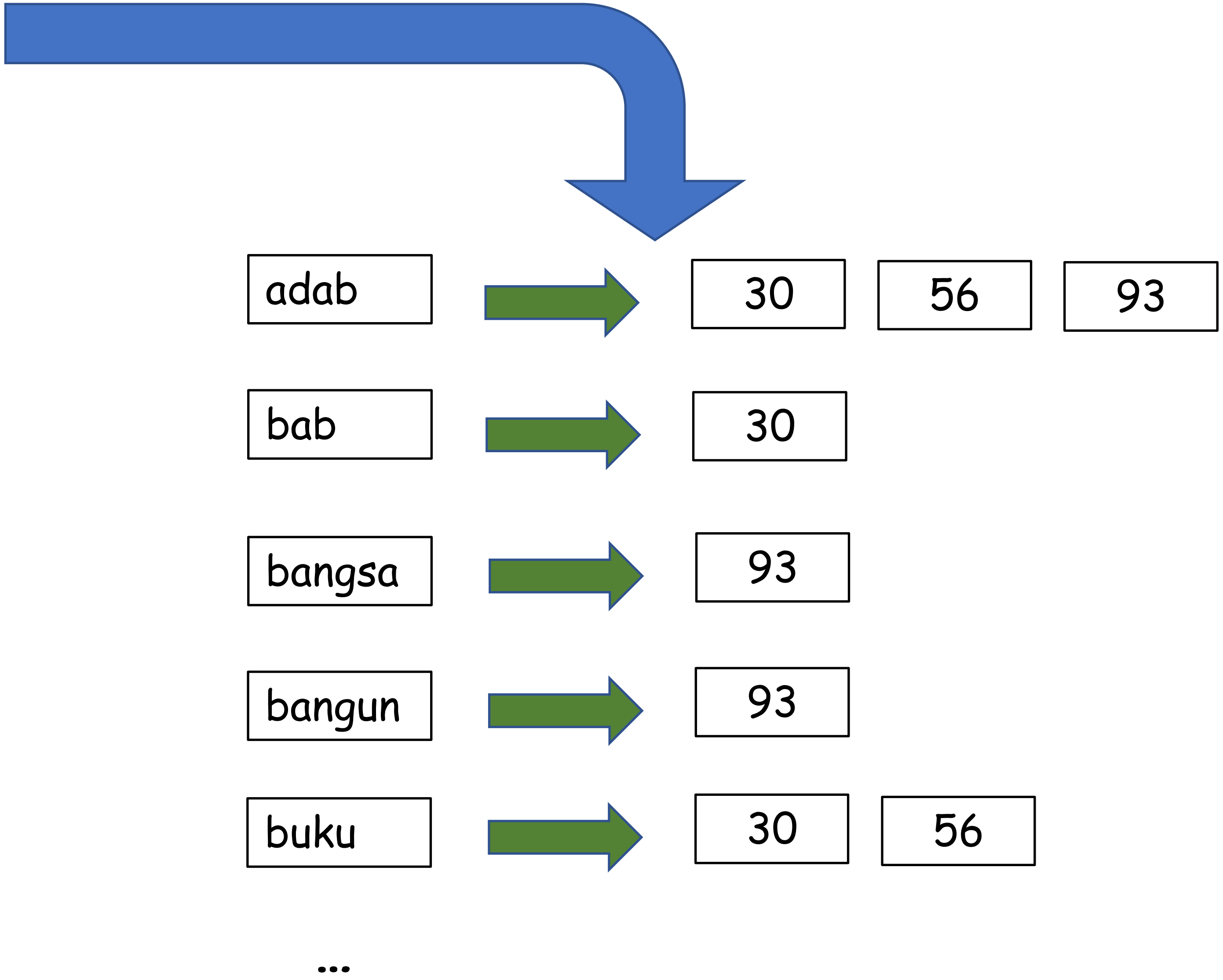
Term	Doc ID
buku	30
sebut	30
kandung	30
bab	30
adab	30
buku	56
isi	56
cerita	56
adab	56
jujur	93
adab	93
utama	93
bangun	93
bangsa	93

external sort, karena gak bisa di handle di memory

Term	Doc ID
adab	30
adab	56
adab	93
bab	30
bangsa	93
bangun	93
buku	30
buku	56
cerita	56
isi	56
jujur	93
kandung	30
sebut	30
utama	93

A Detail View – Step #3 Grouping

Term	Doc ID
adab	30
adab	56
adab	93
bab	30
bangsa	93
bangun	93
buku	30
buku	56
cerita	56
isi	56
jujur	93
kandung	30
sebut	30
utama	93



adab 30 56 93

bab 30

bangsa 93

bangun 93

buku 30 56

...

Preprocessing: Tokenization, Normalization, Stemming, Stopwords Removal

Let’s look at the construction process more detail …

Token, Type, and Term

- **Token** is a sequence of characters that has meaning.
- **Type** is a class of all tokens containing the character sequence.
- **Term** is a type that is included in the IR systems' dictionary (index).

makan untuk hidup, bukan hidup untuk makan.

There are 7 tokens [`makan, untuk, hidup, bukan, hidup, untuk, makan`]

There are 4 word types {`makan, untuk, hidup, bukan`}

There are 2 terms {`makan, hidup`}. The word "untuk" and "bukan" are stopwords that are usually omitted in the index.

*contoh di slide ini merupakan versi singkat dari puisi karya Taufik Ismail (Takut 66, Takut 98)

Sentence Segmentation

Presiden takut mahasiswa. Mahasiswa takut dosen. Dosen takut rektor. Rektor takut presiden.

→

```
[“presiden takut mahasiswa”, “mahasiswa takut dosen”,
 “dosen takut rektor”, “rektor takut presiden”]
```

For English & Bahasa Indonesia, we can somehow use simple regular expression (regex) to break on **punctuations** (`[.!?]`).

```
sentences = re.split(r"[.!?]", your_paragraph)
```

Are you sure?

gak bisa yakin berhasil

Apakah akan berhasil untuk kedua paragraf berikut?

Dr. Budi mengunjungi Jakarta untuk yang keempat kalinya. Ia akan berlibur di kota tersebut.
Mr. Moffat says that U.S. Dollar has been rising for geopolitical reasons.

State-of-the-art approach makes use of machine learning.

Tokenization: English

- Naïve approach: split on whitespace and remove punctuations

```
>>> re.findall(r"\w+", "Let’s run together!")
['Let', 's', 'run', 'together']
```

- O’Neill -> [neill]? [oneill]? [o’neill]? [o’, neill]? [o, neill]?
- Clitics: aren’t -> [aren’t]? [arent]? [are, n’t]? [aren, t]?
- Abbreviations: How to tokenize U.S.A ?
- Email address: [alfan, cs, ui, ac, id]? [alfan@cs.ui.ac.id]?

Tokenization: English

- Hypens: well-respected, merry-go-round
    - Ex: Budi is a “you-tell-me-i-can-code” person

- Numbers: 1,500,000 KM
- Dates: 29/07/2022
- IP Address: 123.32.45.233
- Multiword Units: New Zealand, Los Angeles

Tokenization: Bahasa Indonesia

- Naïve approach: sama dengan Bahasa Inggris; Sebagian besar tokenisasi dapat dilakukan dengan menggunakan "whitespace" sebagai pemisah.
- Klitika: "**kelakuanmupun** tidak bisa dibenarkan"
  - [kelakuanmupun]? [kelakuanmu, pun]? [kelakuan, mu, pun]?
- Angka (5.000), Gelar (dr. Budi, Sp.U.), Tanggal (23-12-2022)
- Kata majemuk & Entitas: Rumah Sakit, Universitas Indonesia
  - [Rumah, Sakit]? [Rumah Sakit]? collocation (biasa term yang sering muncul satu sama lain)
  - [Univesitas, Indonesia]? [Universitas Indonesia]?

So, tokenizer is not as simple as you think ☺ there are many issues that you need to consider.

Tokenization is Language Specific: Chinese

Beberapa bahasa di dunia ditulis tanpa spasi!

印度尼西亚大的学生与众不同

Tokenization is Language Specific: Chinese

Beberapa bahasa di dunia ditulis tanpa spasi!

印度尼西亚大的学生与众不同

↓

印度尼西亚大	的	学生	与众不同
UI	‘s	student(s) (are)	special

Tokenization is Language Specific: Arabic

Beberapa bahasa di dunia ditulis tanpa pemisah jelas, dan dari kanan ke kiri!

صباح الخير يا أصدقاء

[صباح, الخير, يا, أص, د, قاء]

Tokenization is Language Specific: Chinese

Simple Approach: **MaxMatch** Algorithm

“Tokenisasi kata terpanjang yang ditemukan di Vocabulary”

Vocab = {印, 度, 尼, 西, 亚, 大, 的, 学, 生, 与, 众, 不, 同,
学生, 不同, 与众不同, 印度尼西亚}

Langsung ke Slide Popular Tokenization (ada Byte Pair Encoding dan lalu ada juga yang dari google WordPiece.

印度尼西亚大的学生与众不同

Match **印度尼西亚** --> **印度尼西亚** is a token
Match **大** -->**大** is a token
Match **的** --> **的** is a token
Match **学生** --> **学生** is a token
Match **与众不同** --> **与众不同** is a token

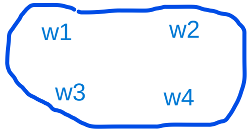
```
def tok(sent, vocab):
  if len(sent) == 0:
    return []
  else:
    for i in range(len(sent), -1, -1):
      head, tail = sent[:i], sent[i:]
      if head in vocab:
        return [head] + tok(tail, vocab)
```

Normalization

The process of canonicalizing tokens

- A query that contains USA should also match documents containing U.S.A.

Sinonim/Equivalent Class
Kalau ada kata w1, ganti jadi w2, w3 juga
ke w2, w4 ke w2 juga, intinya di transform



- Solution #1: maintain equivalence classes
  - anti-discriminatory & antidiscriminatory --> antidiscriminatory

query expansion: tidak saat indexing
Expansion saat indexing.

- Solution #2: maintain relations between unnormalized tokens using query expansion
  - Initial query: [new, car]
  - Expanded query: [new, car, automobile]

user gak tau automobile ini, cuman sistem yang tau.

Tambah term query dengan relasi sinonim.

Kita ingin agar query yang mengandung term **car** match dengan dokumen yang mengandung **car** & **automobile**

Normalization

The process of canonicalizing tokens

- Solution #3: Performing expansion during index construction
  - When a document contains “car”, it is also indexed under “automobile” as well

- Apakah kelebihan dan kekurangan ketiga solusi tersebut?

- Other issues:
  - Lower-casing (case-folding): Indonesia -> indonesia
    - Failed: C.A.T. -> cat (X)
  - True-casing: lower-casing yang lebih pintar dengan machine learning. Tahu kapan perlu lower-casing, dan kapan tidak.

Latihan - Memperkuat Pemahaman

Ada 3 dokumen di koleksi:
D1: car, vacation
D2: vacation, automobile, picnic
D3: picnic, motorcar, river

Misal, **{car, motorcar, automobile}** adalah konsep yang sama. Artinya, jika user mencari dokumen yang mengandung "motorcar", dokumen lain yang mengandung "car" dan "automobile" juga harus di-retrieve.

Bagaimana kondisi inverted index & seperti apa query processing yang dilakukan jika normalisasi terkait {car, motorcar, automobile} dilakukan dengan:

1) Solusi #1 -> dengan equivalence class {car, motorcar, automobile} -> car
2) Solusi #2 -> query expansion
3) Solusi #3 -> expansion saat indexing

1) automobile -> D2
Car -> D1
motorcar -> D3
Picnic -> D2, D3
River -> D3
Vacation -> D1, D2

harusnya 1) karena motorcar sama automobile diwakilin sama car

Jadi Car -> D1, D2, D3

2) sama kayak awal dari no 1, yang autombile, motorcar belom disamain

automobile -> D2
Car -> D1
motorcar -> D3
Picnic -> D2, D3
River -> D3
Vacation -> D1, D2

3) automobile -> D1 D2 D3
car -> D1 D2 D3
motorcar -> D1 D2 D3
picnic
vacation

Other Types of Normalization

- Spelling Corrections
  - Apel memilikki kanungan antioksian
  - Illegally parked cars will be fine --> can you spot a typo here? meaningnya salah nanti
- Spelling Variations
  - Normalisation --> Normalization
- Kata-kata “nggak” baku
  - U r so cooooooool!!! --> you are so cool
  - Nyari tempat makan yang santuy dan mantul --> Anda lebih paham ☺
- Expanding Abbreviations
  - IMHO --> in my humble opinion

Inflectional Morphology

infleksional tidak mengubah kata kata

Misalnya book menjadi books (cuman ganti dari singular ke plural), do menjadi does (meaningnya tetep sama).

- Inflection does not change part-of-speech (kelas kata).

- In English, nouns, verbs, and adjectives can be inflected:

- Nouns: plural or singular (-s / -es)
  - Book -> Books ; book and books are Nouns
- Verbs: number of subject (-s), the aspect (-ing), the tense (-ed)
  - Do -> Does ; do and does are Verbs
- Adjectives: comparatives (-er), superlatives (-est / -iest)
  - Happy -> Happiest ; happy and happiest are Adjectives

Inflectional Morphology

- Bahasa Indonesia also has Inflection suffixes

- Particles:
  - -lah: duduk -> duduklah ; duduk dan duduklah merupakan kata kerja
  - -kah: apa -> apakah

- Possesive Pronouns
  - -ku: buku -> bukuku
  - -mu: sepatu -> sepatumu
  - -nya: mobil -> mobilnya

Derivational Morphology

derivational ganti meaning katanya.

- Derivation changes part-of-speech (kelas kata) and sometimes meaning

- Some English derivational suffixes
  - -ly: honest -> honestly ; honest is an adjective and honestly is an adverb
  - -er: read -> reader ; read is a verb and reader is a noun
  - -ize: final -> finalize ; final is a noun and finalize is a verb
  - -ness: happy -> happiness ; happy is an adjective and happiness is a noun

- Some English derivational prefixes
  - un-: healthy -> unhealthy
  - re-: write -> rewrite

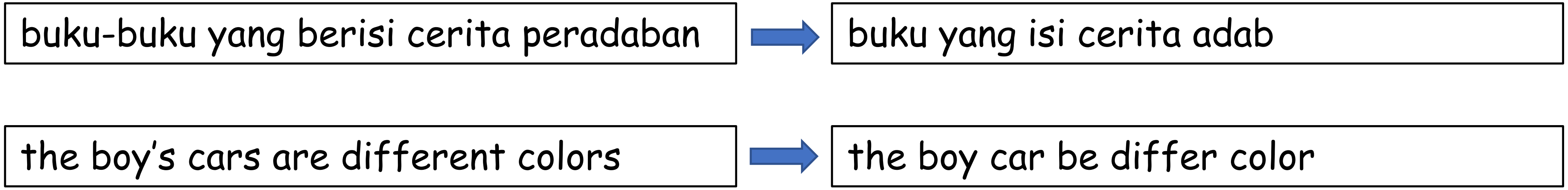
Derivational Morphology

- Some Indonesian derivational suffixes
    - -an: makan -> makanan ; mengubah kata kerja menjadi kata benda
    - -kan: mulia -> muliakan
- Some Indonesian derivational prefixes
    - pe-: muda -> pemuda
- Some Indonesian derivational confixes
    - ke-an: baik -> kebaikan
    - me-kan: aman -> mengamankan

Stemming & Lemmatization

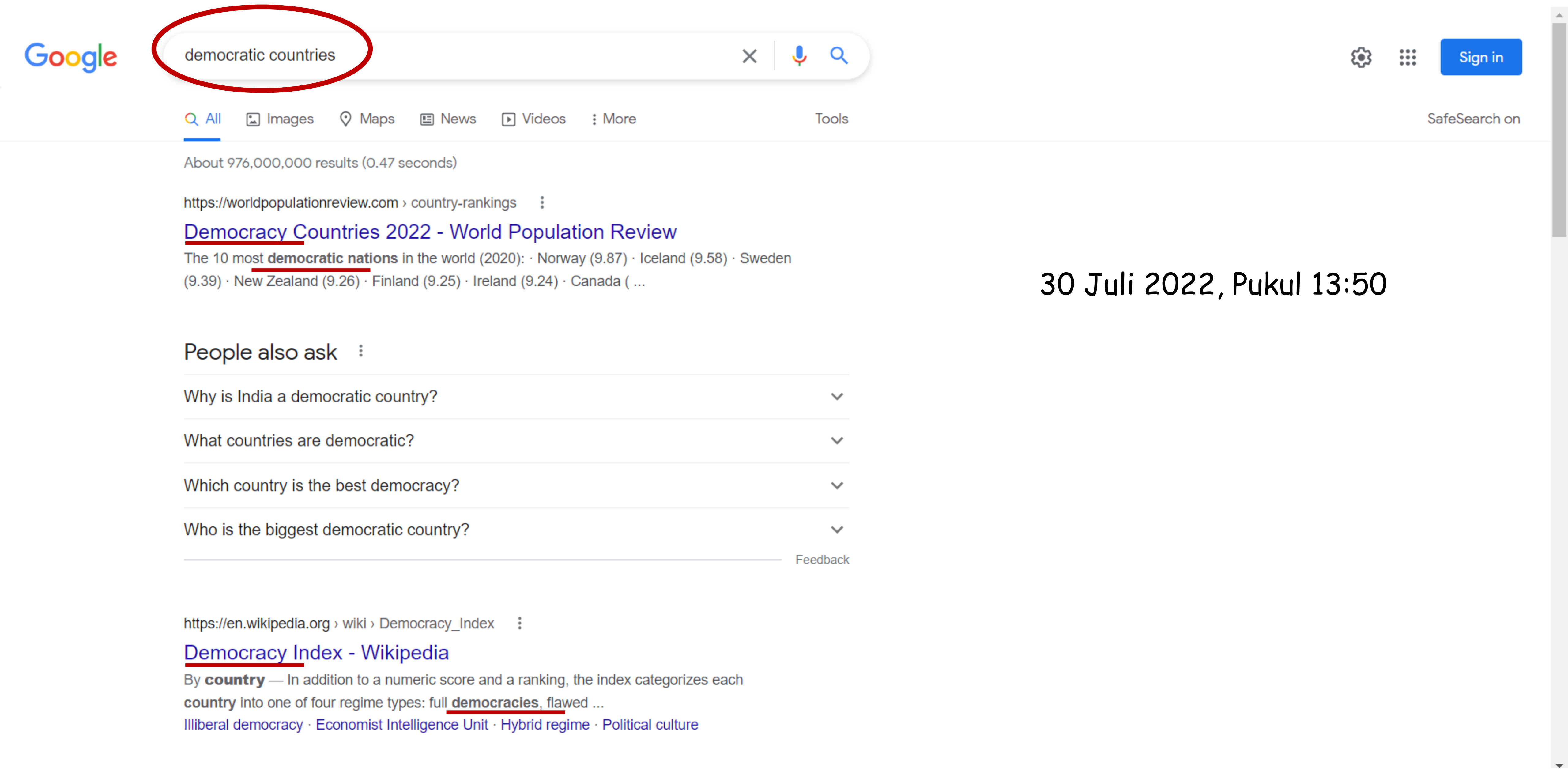
- A word can have many forms; but they are still in the same topic.
  - Query: "demokrasi" ; IR system should retrieve documents that contains "demokrasi", "demokrat", or "demokratisasi"
  - "organize", "organizes", and "organizing" are somehow related

- The goal of stemming & lemmatization is to reduce inflectional and derivational forms of a word to a common base/root form.



buku-buku yang berisi cerita peradaban → buku yang isi cerita adab

the boy's cars are different colors → the boy car be differ color



About 976,000,000 results (0.47 seconds)

https://worldpopulationreview.com › country-rankings

Democracy Countries 2022 - World Population Review

The 10 most **democratic nations** in the world (2020): · Norway (9.87) · Iceland (9.58) · Sweden (9.39) · New Zealand (9.26) · Finland (9.25) · Ireland (9.24) · Canada (...

People also ask

Why is India a democratic country?

What countries are democratic?

Which country is the best democracy?

Who is the biggest democratic country?

https://en.wikipedia.org › wiki › Democracy_Index

Democracy Index - Wikipedia

By **country** — In addition to a numeric score and a ranking, the index categorizes each **country** into one of four regime types: full **democracies**, flawed ...

Illiberal democracy · Economist Intelligence Unit · Hybrid regime · Political culture

30 Juli 2022, Pukul 13:50

Stemming vs Lemmatization

stemming: crude (heuristic process) -> hasil pengurangan kata depan dan belakang (tidak peduli kalau gak ada di kamus, jadi walaupun gak ada di kamus yaudah gitu, tetep aja termasuk.)

biasanya stemming dibarengkan dengan lemmatization

- Stemming

  komputasi lebih cepat, belum tentu bagus

  - A crude heuristic process that removes the beginnings or the ends of words
  - The stemmed word is not necessarily found in the dictionary
  - Stemming increases recall while harming precision

operate, operating, operates, operation, operative, operatives, operational → oper

Apa yang terjadi kalau kita kirim query:
1. operating AND system
2. operational AND research

- Lemmatization

  komputasi lebih berat

  - A proper process that removes inflection endings and return the dictionary form of a word (*lemma*)
  - This process involves dictionary and morphological analysis

operating → operate

Algoritma stemming yang populer

English: Porter's Stemmer (1980)

- This stemmer has been shown to be effective for IR
- There are five phases of word reductions
- First phase:

rule		
-SSES	→	-SS
-IES	→	I
-SS	→	-SS
-S	→	

example		
glasses	→	glass
studies	→	studi
caress	→	caress
books	→	book

- Later phases:
    - A concept of *measure* of a word
    - For example, count the number of syllables and use this number to decide whether a matching rule is a suffix or a part of the word stem.

Jika m > 1, EMENT -> “” replacement -> replac (O) cement -> c (X)

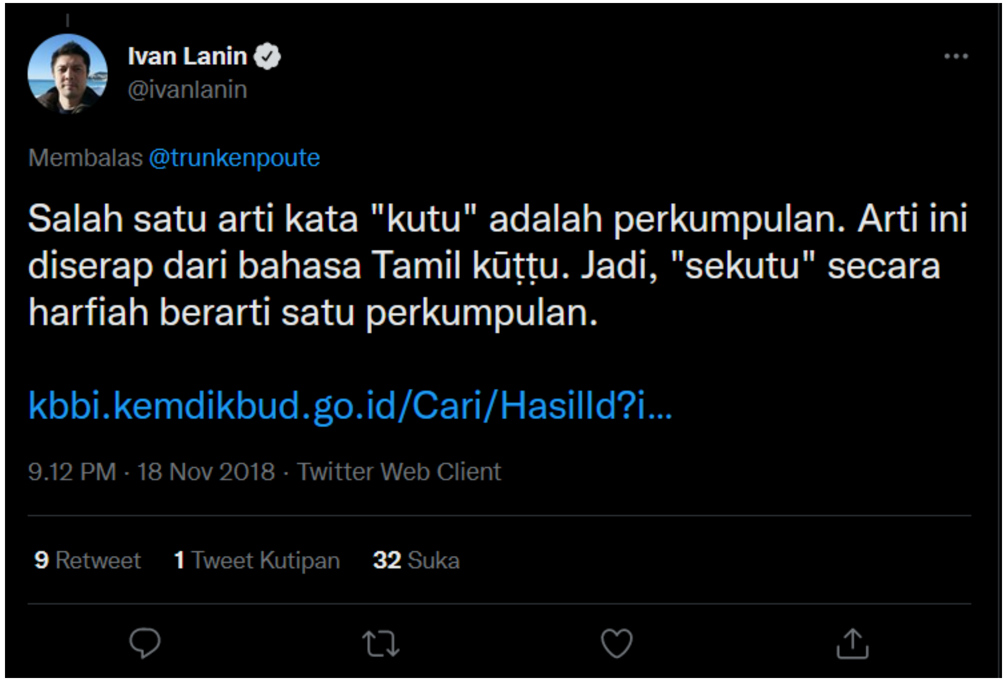
Indonesian: Nazief & Adriani's Algorithm (1996)

- Indonesian affix order of use:

[[[DP+]DP+]DP+] root-word [[+DS][+Possessive Pronouns][+Particles]]

DP = Derivational Particle

Contoh: mempersekutukannyalah
me+ per+ se+ kutu +kan +nya +lah



Ivan Lanin
@ivanlanin

Membalas @trunkenpoute

Salah satu arti kata "kutu" adalah perkumpulan. Arti ini diserap dari bahasa Tamil kūṭṭu. Jadi, "sekutu" secara harfiah berarti satu perkumpulan.

kbbi.kemdikbud.go.id/Cari/HasilId?i...

9.12 PM · 18 Nov 2018 · Twitter Web Client

9 Retweet 1 Tweet Kutipan 32 Suka

Adriani et al., Stemming Indonesian: A Confix-Stripping Approach, ACM TALIP 2007

Indonesian: Nazief & Adriani's Algorithm (1996)

- Indonesian affix order of use:

harus ada di kamus KBBI

[[[DP+]DP+]DP+] root-word [[+DS][+Possessive Pronouns][+Particles]]

- Words of three or fewer characters cannot contain affixes, so no stemming is performed on such short words.
- Affixes are never repeated, so a stemmer should remove only one of a set of seemingly repeating affixes.
- We can use confix restriction during stemming to rule out invalid affix combinations. (for example, be- word –i is not allowed).
- A dictionary is needed to check if the stemming has arrived at a root word.

Butuh banget dictionary (kalau bisa butuh dictionary yang besar)

Setiap kali motong di cek kamus (iteratively)

Adriani et al., Stemming Indonesian: A Confix-Stripping Approach, ACM TALIP 2007

Removing Stop Words ?

kata-kata selain content words (Nouns, Adjective, Verbs, etc)

- Stop words are extremely common words that has little value.
- To determine a stop list, we can sort terms by collection frequency and take the most frequent terms.
- Removing stop words can reduce the number of postings.

Some English Stop Words
own, same, so, than, too, very, no, not, such, in, out, on, off, be, to, above, at, by, of, …

Some Indonesian Stop Words
ada, adalah, agar, akan, amat, antara, apa, apabila, atau, bagai, bahwa, bahkan, yang, yaitu, …

Tala, F. Z. (2003). A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. M.Sc. Thesis. Master of Logic Project. Institute for Logic, Language and Computation. Universiteit van Amsterdam, The Netherlands.

Removing Stop Words ?

Jadi

1. Tokenization
2. Normalization
3. Stemming and Lemmatization
4. Stopwords Removal (optional now)

- Do we really have to remove stop words?
- What about the following phrase queries?
  - "to be or not to be"
  - "let it be"
  - "As we may think"

IR orang gak perlu lagi buat inverted indexnya

Karena nanti informasinya bisa ilang

- The general trend in IR systems over time has been from standard use of quite large stop lists (200–300 terms) to very small stop lists (7–12 terms) to no stop list whatsoever.
- Web search engines generally do not use stop lists.
- Although keeping stop words may increase the size of postings, our latest technologies say "no problem".
  - Good index compression techniques
  - The size of secondary storage has been increasing