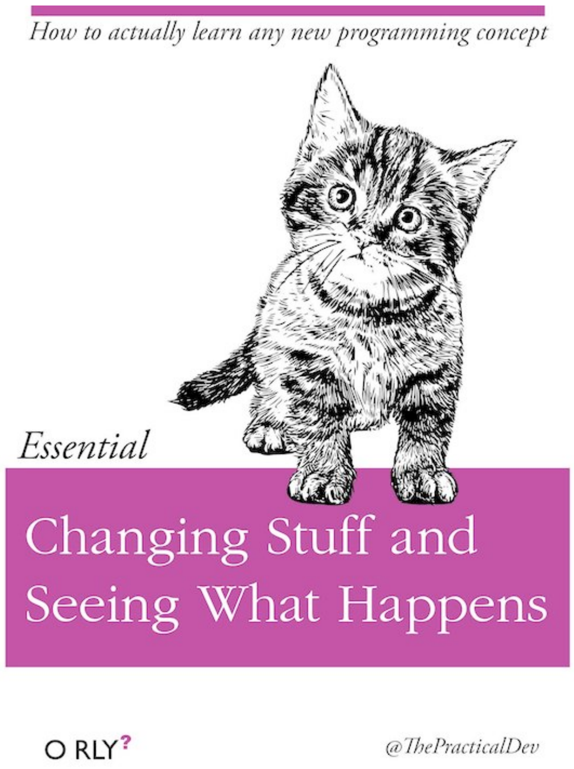
Hackathon

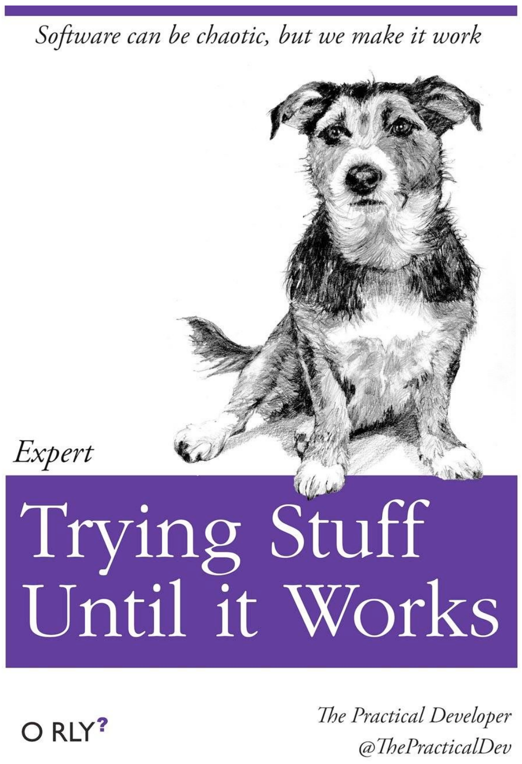
> 5 December. 9:00-14:00

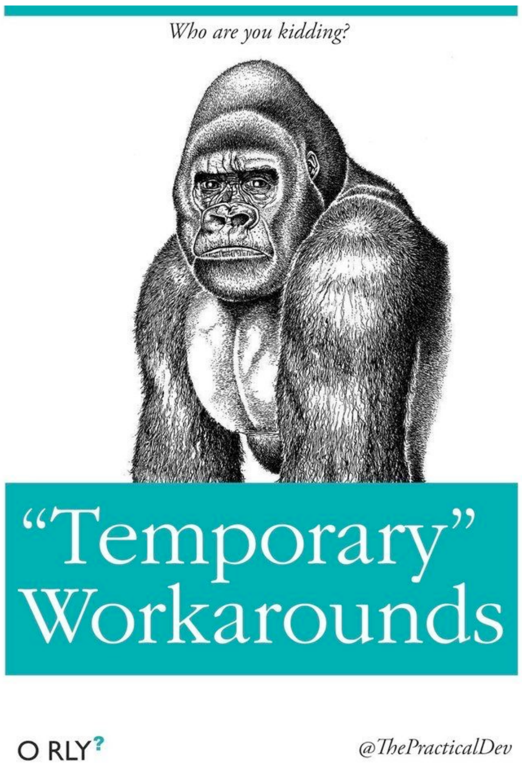
Our Data

What? Why? How?

Today's Plan


How to actually learn any new programming concept
Essential
Changing Stuff and Seeing What Happens
O RLY?
@ThePracticalDev


Software can be chaotic, but we make it work
Expert
Trying Stuff Until it Works
O RLY?
The Practical Developer
@ThePracticalDev


Who are you kidding?
"Temporary" Workarounds
O RLY?
@ThePracticalDev

Preprocessing

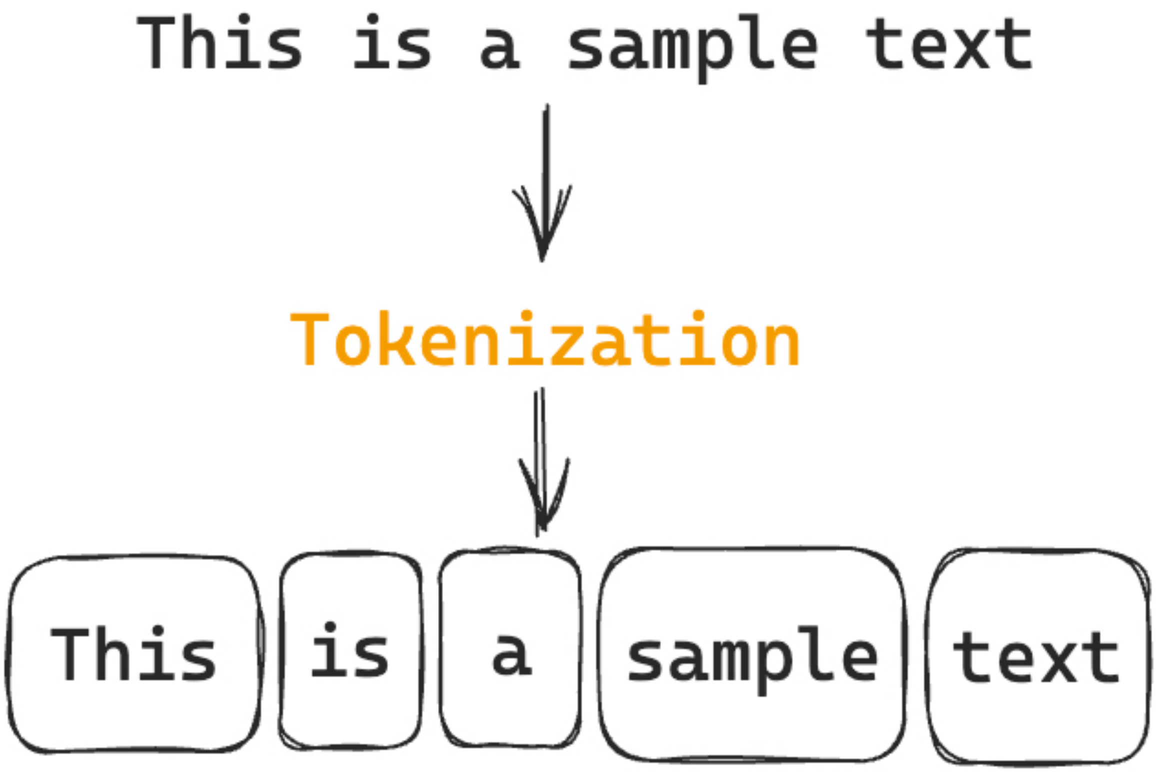
> transitive verb;
>
> *to do preliminary processing of (something, such as data)*

Why? Demo..

Transformation

- Lowercase (K > k)
- Remove accents (ŏ > o)
- Parse HTML (`<a href="...">Korea</a>` > Korea)
- Remove URLs (www. > `removed`)

Tokenization



This is a sample text

Tokenization

This

is

a

sample

text

Tokenization

- Orange has some issues..

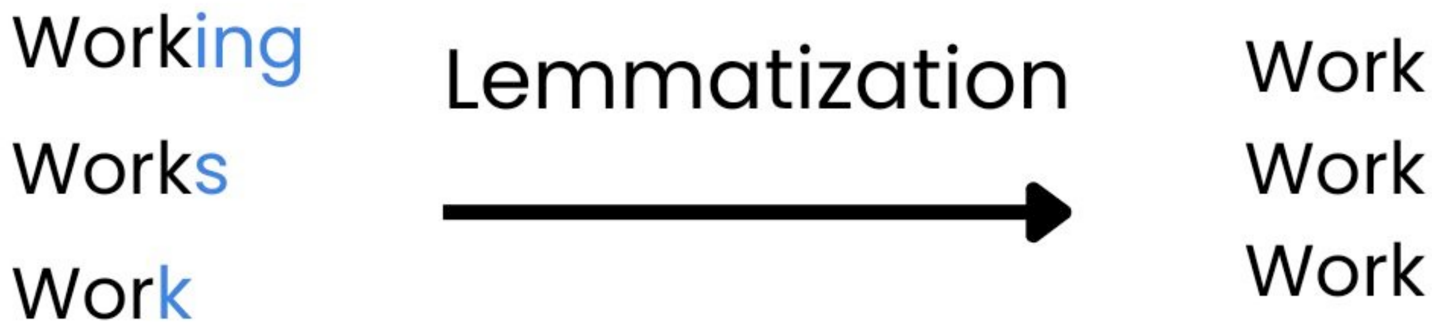
Best way to tokenize is with `regex` (regular expression)

`[가-힣]+`

`[]` = range of characters (e.g a-d, 가-힣)

`[]+` = any length

Lemmatization



Working

Works

Work

Lemmatization

Work

Work

Work

Korean Lemmatization

Word (표면형)	Lemma (기본형)	En
갔다	가다	go
먹었습니다	먹다	eat
봤어요	보다	see
예뻤다	예쁘다	be pretty
살았어요	살다	live
들었지만	듣다	hear
친구들과	친구	friend
학교에서	학교	school
민족의	민족	nation
사람들은	사람	person
빨랐다	빠르다	fast
됐습니다	되다	become

Korean Lemmatization

- Orange Korean Lemmatization not good.
- we adjust our regex.

1. Take Any word any length,
2. if ends in 을/에서/가 etc etc.. remove that
3. Keep starting part.

```
\b(?![가-힁]*(?:을|를|이|가|은|는|의|에서|으로|로|와|과|에게|께|한|하다|하고|하며|했던|했다|했다가|주의)\b)([가-힁]{2,})\b
```

For Perfection > `Python/R`

Filtering

Stopwords

Language	Example Stopwords
English	the, and, they, to, in, is
Dutch	de, het, en, van, in, ik
Korean (한국어)	그리고, 하지만, 매우, 다시, 안, 함께

Filtering

1 Length Syllables

본, 다, etc. Leftovers

Regex to remove:

```
^.$
```

Filtering

Document Frequency

Extremely Common, or rare are not informative.

Usually good baseline:

- If a token appears in fewer than 10% of documents
- If a token appears in more than 90% of documents

Filtering

POS Tags

Tag the Part-of-Speech

`Students prompt GPT desperately.`

```
Students NOUN   prompt VERB   GPT PROPN   desperately ADV   . PUNCT
```

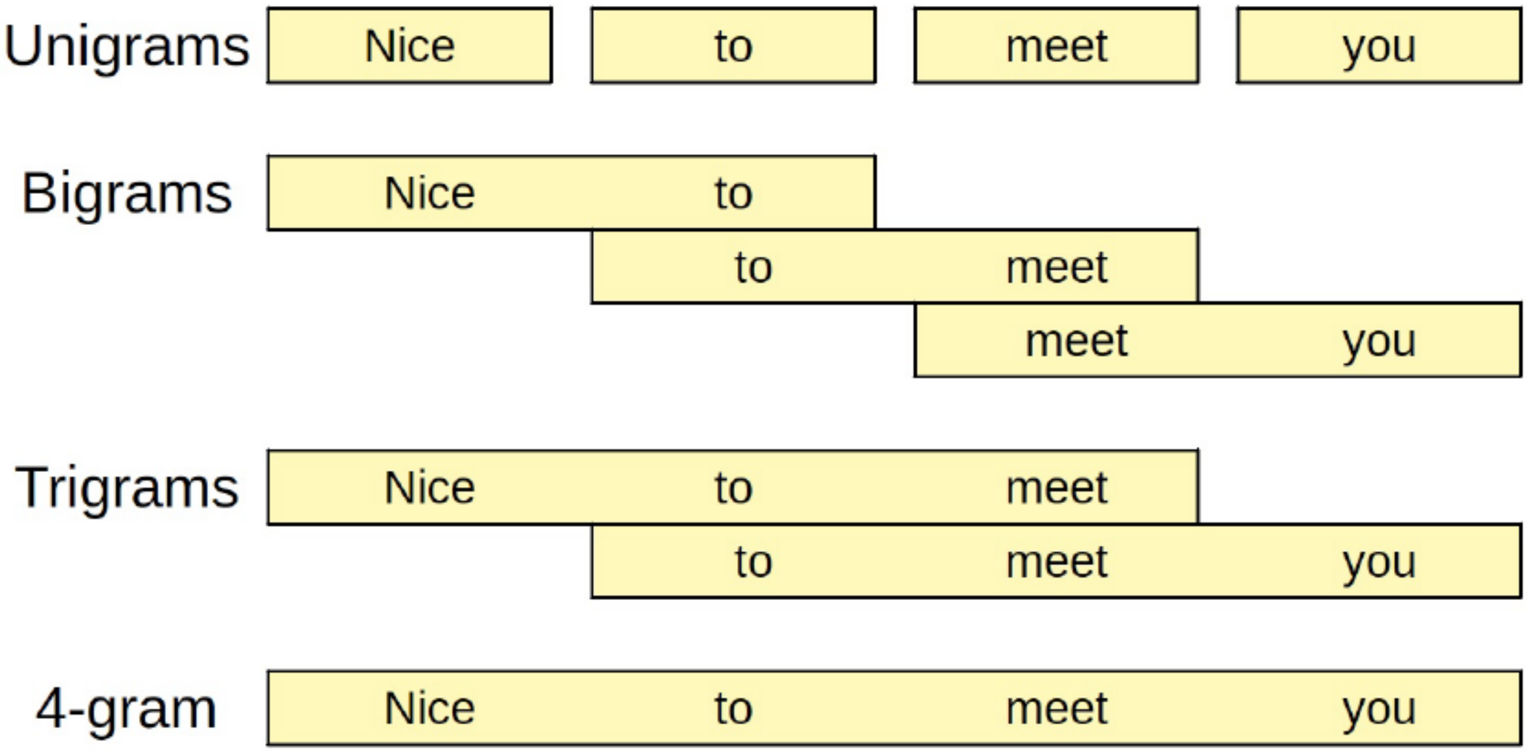
or

GPT가 교수님 이메일보다 먼저 답한다.

```
GPT가 f+jcs   교수님 ncn   이메일보다 ncn+jca   더 mag   빨리 mag   답한다 ncpa+xsv+ef   . sf
```

Usually we keep only nouns.

N-Grams



Unigrams: Nice | to | meet | you

Bigrams: Nice to | to meet | meet you

Trigrams: Nice to meet | to meet you

4-gram: Nice to meet you

N-Grams

- Helps find Bi-grams / Tri-grams. e.g

Our Data

Our Data

Counting Words..

Step 1 — The naïve count

Word	Count
내용	**4,112**
나라	2,740
생활	2,411
민족	2,390
역사	2,305

Step 2 — Look closer

Examples from the corpus:

> “이 단원의 **내용**을 살펴봅시다.”
> “다음 **내용**을 읽고 답하시오.”
> “학습 **내용**을 정리합시다.”

> High frequency, zero insight.

The problem

Counting words only tells us **what appears a lot** — not **what really matters**.

The idea

TF-IDF gives weight to words that are
👉**common in one text**,
but 👈**uncommon in others.**

How it works

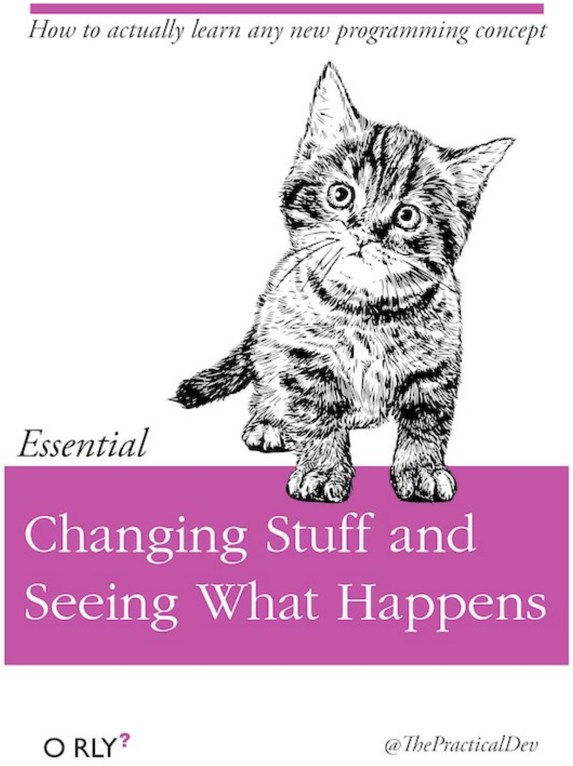
- **TF (Term Frequency):** how often a word appears in a document
- **IDF (Inverse Document Frequency):** how rare that word is across all documents
- Multiply them → **TF × IDF = importance score**

TF-IDF

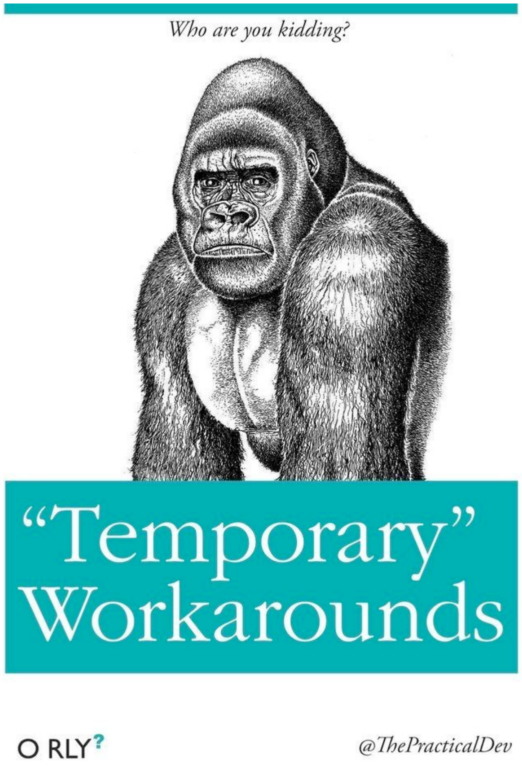
Word	**TF-IDF**
식민통치	0.88
자주독립	0.94
근대화	0.82
민족정신	0.80
내용	**0.02**

Assignment

Remember


How to actually learn any new programming concept
Essential
Changing Stuff and Seeing What Happens
O RLY?
@ThePracticalDev

Software can be chaotic, but we make it work
Expert
Trying Stuff Until it Works
O RLY?
The Practical Developer
@ThePracticalDev


Who are you kidding?
"Temporary" Workarounds
O RLY?
@ThePracticalDev