

# Kryssord Løser

Gruppe 26

Andreas Sandvik Solli

# Hva er et kryssord?

“Kryssord er en bokstavgåte hvor man skal fylle åpne ruter i et diagram med korrekte ord. Ordene bestemmes vanligvis ut fra to lister med nummererte stikkord, én for vannrette og én for loddrette ord, hvor tallene finnes igjen i de diagramrutene hvor ordene skal begynne. Stikkordene kan også noteres i selve diagrammet.” - SNL 2024.

## Oppsummert:

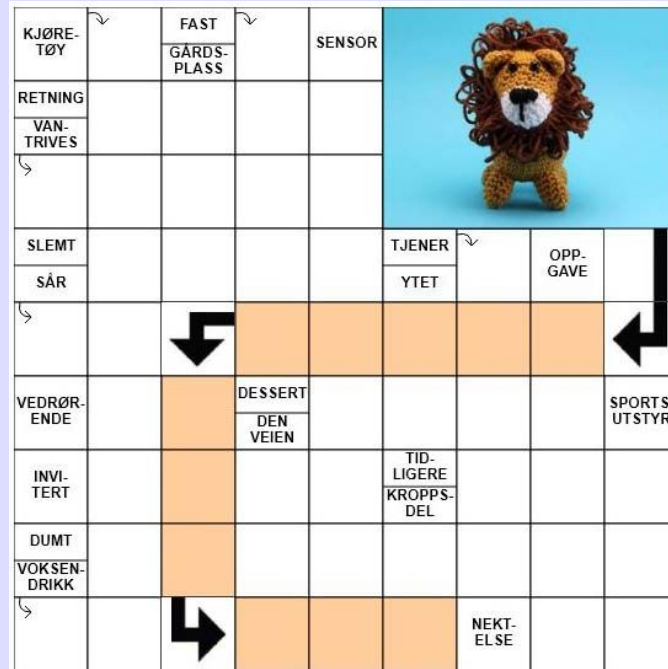
Kryssord er et brett med åpne ruter der du skal fylle inn ord basert på hint som følger med brettet. Det er et visst antall åpne felt til hvert ord, som indikerer antall bokstaver. Ord kan krysses ettersom de er loddrett og vannrett og derfor gi flere hint til hva riktig ord kan være.

## Takeaways:

Løsning via hint, antall bokstaver og bokstaver som krysser.

## Mitt kryssord:

Ikke bokstaver som krysser over :(



© Kryssord: Joakim Kalmar  
Foto: flickr.com/korova08 - CC BY-SA 2.0

# Mitt kryssord

Med begrenset tid så har mitt kryssord noen "limits".  
Altså ikke ord som krysser over, ikke abbreviations.  
Det er også tilfeldig hvor vanskelig kryssordet er.  
Mange kryssord er vanskelig pga lite brukte/gamle ord.  
Mitt er tilfeldig valgt, og derfor vil vanskelighets grad for en vanlig person være veldig varierende.

Algoritme vil få det like vanskelig samme hva slags ord så lenge kryssordet er basert på liknende/lik dictionary/data som løseren.

Kan oppstå problem dersom to ord har like mange bokstaver og samme synonym ettersom jeg ikke har kryssende ord.

Norsk eksempel:

Synonym for god kan være skarp **og** snill.

løsning      crosswords.json

Ord	Lengde	Hint
cram	4	Study for finals
untied	6	Set free
salt	4	NaCl
rhetoric	8	Bombast
laine	5	'Mule train' singer

# DATA

Ord + betydning

## 01 DATA

American.csv  
(55851 ord + hint)

~~abbreviations.json~~

## 02 KLASSER FRA OBLIG

Corpus  
Dictionary  
**Document**  
DocumentPipeline  
**InvertedIndex**  
**Normalizer**  
PorterStemmer  
Posting  
PostingList  
Soundex  
**Tokenizer**  
VariableByteCodec

## 03 KRYSSORD

Basert på American.csv  
Generert to stk med clues

Clues:  
Bokstaver + definisjon

## 04 HOVEDKLASSE

Oppretter  
InMemoryDocument &  
InMemoryInvertedIndex  
med American.csv

Finner kandidater basert  
på hint og antall bokstaver

Kalkulerer score mellom  
de potensielle  
kandidatene.

# Hvordan relevant til pensum?

Ettersom vi søker på ord og relevans bruker jeg klassene fra tidligere obliger til å forbedre søkingen. Er også en del klasser som er med fordi de brukes i disse klassene.

InvertedIndex:

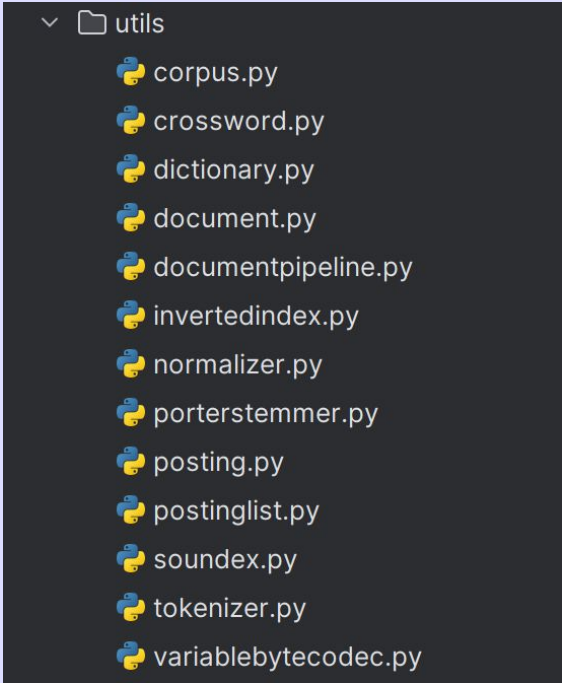
Muliggjør rask oppslag av dokumenter som matcher hint eller lengde på ord. Ved indeksering sikrer vi effektivt søk.

Normalizer og Tokenizer:

Normalisering konverterer teksten til kanonisk form, og tokenisering deler det opp i tokens for videre behandling. Det gjør at variasjon i tekst som størrelse osv. ikke ødelegger.

Ikke klasse fra oblig, men bruker også regex for effektiv ord mot pattern matching, ville vært mer relevant om jeg fikk inn at ordene er på tvers også.

Til slutt bruker jeg også rangering mellom de ulike kandidatene for å finne beste mulige. Som er basert på det har vi vært gjennom.



```
utils
├── corpus.py
├── crossword.py
├── dictionary.py
├── document.py
├── documentpipeline.py
├── invertedindex.py
├── normalizer.py
├── porterstemmer.py
├── posting.py
├── postinglist.py
├── soundex.py
├── tokenizer.py
└── variablebytecodec.py
```

# Kode (for den som måtte ønske), kun metoder

## Finne kandidater:

```
pattern_regex = self.convert_pattern_to_regex(pattern)
pattern_candidates = [
    doc for doc in self.corpus
    if re.fullmatch(pattern_regex, doc.get_field("length", ""))
]
ranked_candidates = sorted(
    pattern_candidates,
    key=lambda doc: self.clue_similarity_score(clue, doc.get_field("clue", "")),
    reverse=True
)
return [doc.get_field("length", "") for doc in ranked_candidates]
```

## Rangering:

```
normalized_clue = self.normalizer.normalize(clue)
normalized_candidate_clue = self.normalizer.normalize(candidate_clue)
return SequenceMatcher(None, normalized_clue, normalized_candidate_clue).ratio()
```

## Regex konvertering:

```
return pattern.replace('-', '.').lower()
```

```

Clue: Study for finals, Length: 4
Clue: "... or ___!", Length: 4
Clue: Honeybun, Length: 3
Clue: "Mule Train" singer, 1949, Length: 5
Clue: Los Angeles attraction, with "the", Length: 6
Clue: Pet protection org., Length: 5
Clue: Drained, Length: 3
Clue: Flip (out), Length: 5
Clue: Fluster, Length: 5
Clue: Magnate, Length: 5

```

	Word	Clue
0	cram	Study for finals
1	else	"... or ___!"
2	pet	Honeybun
3	laine	"Mule Train" singer, 1949
4	archer	Los Angeles attraction, with "the"
5	ASPCA	Pet protection org.
6	GNP	Drained
7	freak	Flip (out)
8	addle	Fluster
9	titan	Magnate

# Oppsummering, resultat

Venstre er hva som blir gitt til kryssord solveren.  
Høyre er hva som returneres/printes.



Kort demo av koden (24 sek)

# Takk for meg

In case forrige side ikke funket

[https://drive.google.com/file/d/12Xy5HePVF0BToqPzjsqyANWrU\\_d0-Fz2/view?usp=sharing](https://drive.google.com/file/d/12Xy5HePVF0BToqPzjsqyANWrU_d0-Fz2/view?usp=sharing)