Porters algorithm

Innhold

- Stemming generelt
- Beskrivelse av Porters algorithm
- Funn og resultater

Stemming

- Normalisering
- Termer til baseform
- Motivasjon
  - Bøyde ord skal tolkes likt
  - “Jeg spiser mat”, “Jens spiste salat”
  - Bedre recall (kan gi verre presisjon)

Hvordan?

- Finnes flere former
  - Lookup table, regelbasert
- Porters er regelbasert
  - Suffix stripping (prefix finnes og)
- Performance
  - vs Lemmatization
- Språk avhengig
  - spansk, tysk, finsk

Eksempel

- Mål:
  - car, cars, car's, cars' → car
  - box, boxes → box
- Regler kan være:
  - Fjerne “s ”
  - Fjerne “`s”
  - Fjerne “s`”
  - Fjerne “es”
- Kan gi feil:
  - cares → car
  - Lemmatization vil ikke gjøre samme feilen

Porters algoritme

- Stemming
- Engelsk-spesifik
- Deler opp termer i vokaler og konsonanter
- 5 steg

Regler for konsonanter

- Deler opp ord i vokaler(v) og konsonanter(c)
- Regner konsonanter som alle bokstaver utenom vokalene A, E, I, O, U, og utenom Y etter en konsonant.
- Vokaler:
  - A, E, I, O, U
  - F.eks. STORY - O, Y
- Konsonanter:
  - TOY - T, Y
- En bokstav som ikke regnes som en konsonant, regnes som en vokal

Regler for konsonanter

- Alle ord kan noteres på formen [C](VC)*[V]
- C(C)* -> C
- V(V)* - > V
- Teller antallet (VC) forekomster -> measure(m)

Regler for stemming

- Reglene for å bytte ut en suffix er på formen (betingelse) S1 -> S2
  - Hvis betingelsen oppnås, og termen har suffixen S1, så vil suffixen endres til S2.
- (betingelse) kan være:
  - (m >/=/< x) - antallet measures
  - *S - slutter på S (Fungerer likt for andre bokstaver)
  - *v* - inneholder en vokal
  - *d - slutter med dobbel-konsonant
  - *o - slutter med cvc, der den andre c-en ikke er W, X, eller Y
  - uttrykk med and, or og not

Stegene i algoritmen

- Ordet går gjennom 5 steg i algoritmen med ulike stemming-regler, der noen vil treffe, og andre ikke.
  - 1: Flertall og partisipp
  - 2 - 4: Fjerner ulike stemminger
  - 5: Opprydding

Step 1a

1. SSES → SS
2. IES → I
3. SS → SS
4. S →

Step 1b

1. (m>0) EED → EE
2. (*v*) ED →
3. (*v*) ING →

If the second or third of the rules in Step 1b is successful, the following is performed.

1. AT → ATE
2. BL → BLE
3. IZ → IZE
4. (*d and not (*L or *S or *Z)) → single letter
5. (m=1 and *o) → E

Step 1c

1. (*v*) Y → I

Step 2

1. (m>0) ATIONAL → ATE
2. (m>0) TIONAL → TION
3. (m>0) ENCI → ENCE
4. (m>0) ANCI → ANCE
5. (m>0) IZER → IZE
6. (m>0) ABLI → ABLE
7. (m>0) ALLI → AL
8. (m>0) ENTLI → ENT
9. (m>0) ELI → E
10. (m>0) OUSLI → OUS
11. (m>0) IZATION → IZE
12. (m>0) ATION → ATE
13. (m>0) ATOR → ATE
14. (m>0) ALISM → AL
15. (m>0) IVENESS → IVE
16. (m>0) FULNESS → FUL
17. (m>0) OUSNESS → OUS
18. (m>0) ALITI → AL
19. (m>0) IVITI → IVE
20. (m>0) BILITI → BLE

Step 3

1. (m>0) ICATE → IC
2. (m>0) ATIVE →
3. (m>0) ALIZE → AL
4. (m>0) ICITI → IC
5. (m>0) ICAL → IC
6. (m>0) FUL →
7. (m>0) NESS →

Step 4

1. (m>1) AL →
2. (m>1) ANCE →
3. (m>1) ENCE →
4. (m>1) ER →
5. (m>1) IC →
6. (m>1) ABLE →
7. (m>1) IBLE →
8. (m>1) ANT →
9. (m>1) EMENT →
10. (m>1) MENT →
11. (m>1) ENT →
12. (m>1 and (*S or *T)) ION →
13. (m>1) OU →
14. (m>1) ISM →
15. (m>1) ATE →
16. (m>1) ITI →
17. (m>1) OUS →
18. (m>1) IVE →
19. (m>1) IZE →

Step 5a

1. (m>1) E →
2. (m=1 and not *o) E →

Step 5b

1. (m > 1 and *d and *L) → single letter

Kvantitativ analyse

- Utvide Normalizer klassen med PortersNormalizer.
- Porters algoritme er hentet fra github-repoet til jedijulia.

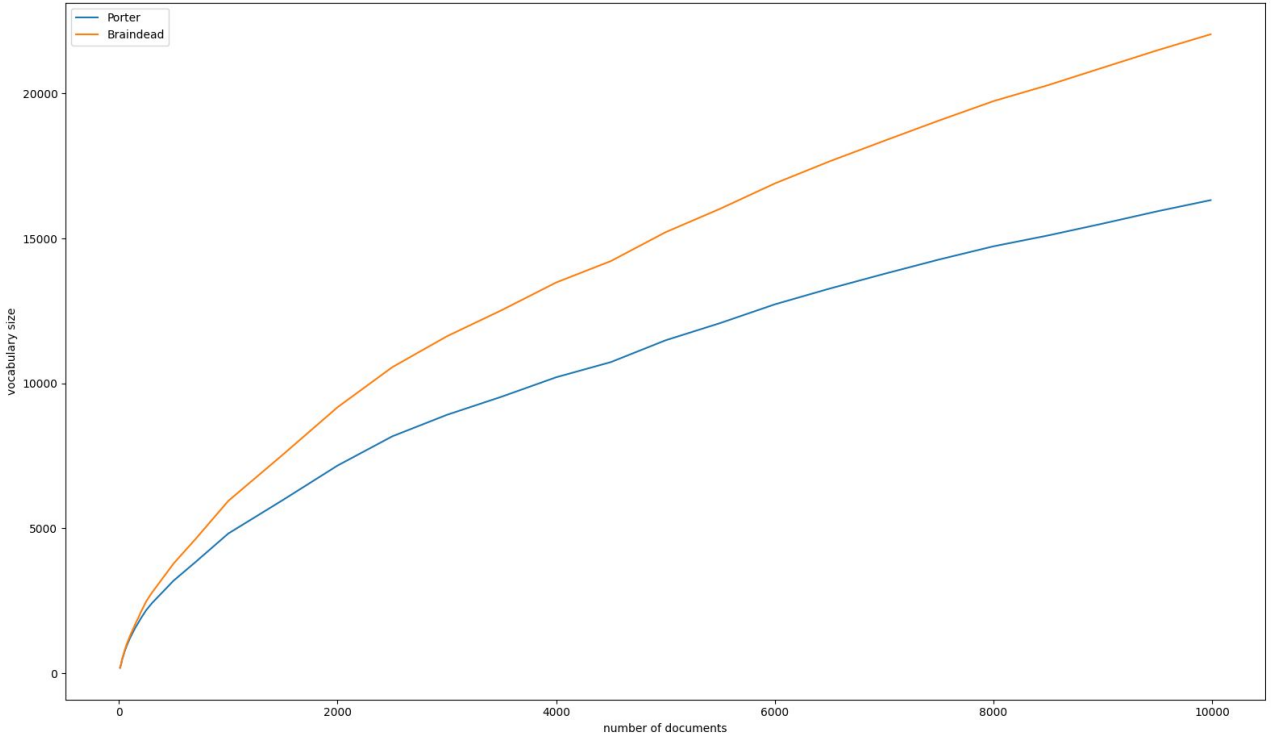
```
class PortersNormalizer(Normalizer):
    def __init__(self) -> None:
        self.porter = PorterStemmer()

    def canonicalize(self, buffer: str) -> str:
        return buffer

    def normalize(self, token: str) -> str:
        return self.porter.stem(token.lower())
```

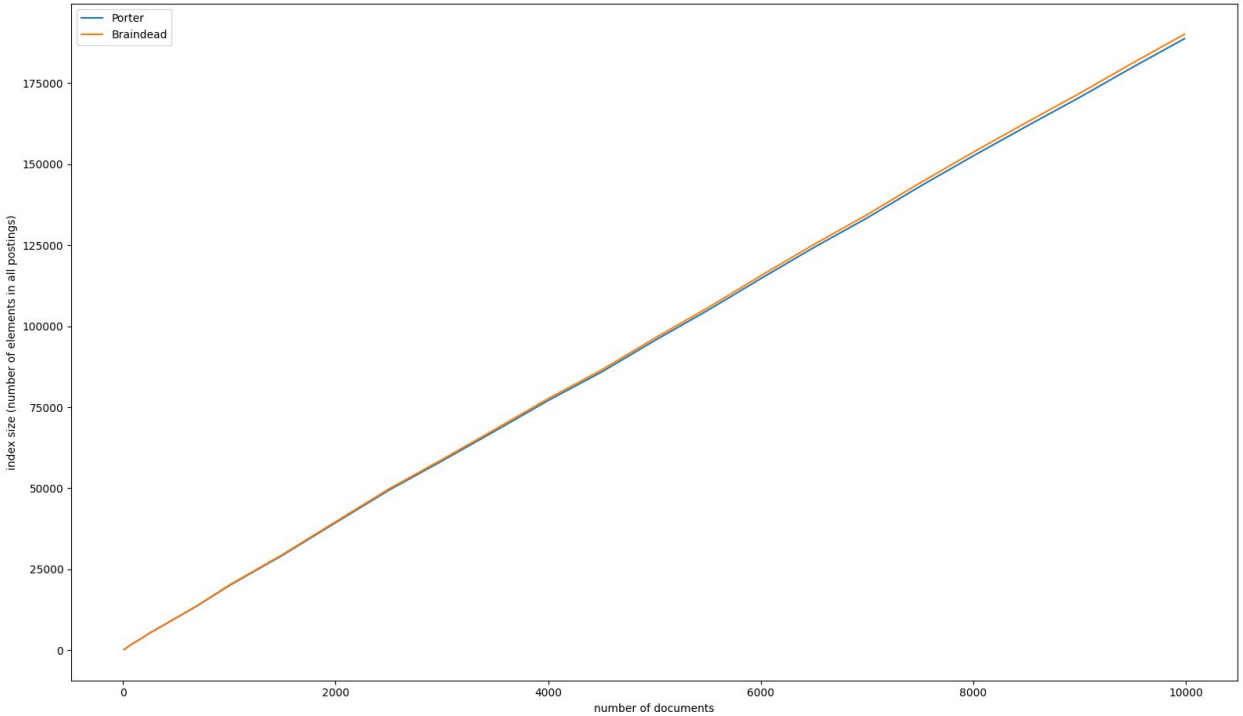
PortersNormalizer VS. BraindeadNormalizer
corpus: “en.txt”

Størrelse på vokabular



- **Størrelse = Antall unike termer i vokabularet**
- Forskjellen mellom Porters og Braindead øker med størrelsen på corpsus
- Vi tror avstanden mellom de 2 kurvene vil flate ut når n blir veldig stor.

Størrelse på invertert indeks



- **størrelse = sum(lengden av alle inverter indekser)**
- Forskjellen er avhengig av størrelsen på dokumenter

Kvalitativ analyse av recall

QUERY: “wage increase”

Braindead normalizer

- “How would you address critics who say Wal-Mart should go beyond the wage increase it announced?”

Porters Normalizer

- “Voters in November 2014 approved increasing the minimum wage from $7.75 an hour.”
- “Q. How would you address critics who say Wal-Mart should go beyond the wage increase it announced?”
- “Is increasing the minimum wage a good idea?'”
- “Corporate America is blaming its poor profits on minimum wage increases.”