

Navn: Majdi Alali

Uio-brukernavn: majdioa

### Noen notater:

-Først må jeg si at jeg har programmert alt inni de to python-filene som er gitt.

-Legg merke til at du kan ukommentere print(«») som jeg har lagt til på alle mine metoder (i tillegg til main-område) slik at du kan teste mine resultater/output en hver for seg eller alle hvis du vil.

-Legg merke også til at jeg har lagt til noen kommentarer på koden slik at det blir lettere for deg å lese koden. Men noen ganger er koden selvforklarende, derfor skriver jeg litt om den her i rapporten.

### Logistisk regressjon:

Jeg har opprettet to instans-variabler. Den ene heter **AlleUniqueSymboler\_liste**, og den tar liste over alle unike trekk i datasettet. Den andre heter **mapping**, og den tar en ordbok hvor språk er som nøkler og heltall er som verdier.

**b**

**1.**

- Jeg har laget to hjelpe metoder som har noe å gjøre med **extract\_unique\_symbols()**.
- Den ene er **IPA\_liste()** tar alle transkripsjoner og returnerer en liste over alle trekkene (hvor et trekk telles hvis det forekommer flere enn en gang)
- Den andre er **IPA\_tupel()** tar en **IPA\_liste()** og returnerer en liste over alle trekk hvor trekkene ikke kan forekomme flere enn en gang.
- **extract\_unique\_symbols()** kaller de to siste metodene. I tillegg begrenser den antall trekk som skal være med videre hvor det tar bare trekkene som forekommer minst 10 ganger i trainingssettet.

- Her er resultatene fra de tre metodene:  
lengde av alle sympoler (med forkomester) i trainingssettet er: 3236862  
lengde av alle unike sympoler (uten forkomester) i trainingssettet er: 203  
lengde av alle unike sympoler i trainingssettet, som forekommer minst 10 ganger, er: **158**

**2.**

- For denne delen, har jeg fulgt anbefalingene i textoppgaven når det gjelder å opprette en np.array med zeros. Etterpå har jeg fylt noen celler med 1 hvis trekk er med på transkripsjon.
- Den ser sånn ut:

```
[[0 0 0 ... 0 0 0]]
```

```
[1 0 0 ... 0 0 0]]
```

```
[0 0 0 ... 0 0 0]
...
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]]
```

3.

I **train()** har jeg kalt på listen over unike trekk og har lagret dem i en instans-variabel kalt **mapping**. Denne tar språk som nøkler og heltall som verdier hvor hvert språk har et eget unik ID/heltall. Etterpå har jeg tatt vare på heltallene i en variabel kalt språk\_som\_heltall slik det blir lett å få tak i språk sine ID-ene og bruke dem i **fit()**. Så har jeg sjekket om lengde av transkripsjonene er like språks. Til slutt har jeg trent min metode vha. den innbygde-metode **fit()** ved hjelp av **extract\_feats()** og ID-ene, og returnert nemlig den.

c)

1.

Metoden **predict()** gjør som den skal. Men det er viktig å nevne at når vi predikerer de gitte transkripsjonene ved hjelp av **SGDClassifier** får 'rumensk' isteden av 'norsk'. Slik:

Mest sannsynlige språk for ordene: ['spansk', 'rumensk', 'islandsk', 'tysk']

2.

evaluering

På **evaluate()**, har jeg brukt **predict()** for å predikere transkripsjonene, som er et parameter til metoden, og lagret den i en variabel kalt 'predicted-spraak'. Den har vi bruk til som et parameter for innebygde metoder til **scikit-learn**. Resten av koden er bare implementasjon til innebygd-metodene til **sk-learn**.

**Her er resultatene etter kjøring av metoden evaluate():**

Legge merke til at tallene (som står på toppen av kolonnene) er ID-ene til språknavnene. Dvs. at i dette tilfelle representerer heltall 0 'arabisk' og 1 representerer 'islandsk' osv. slik at vi vet hvilket scorene som tilhører til hvilke språk. Derfor har jeg opprettet variabelen **orderedSpråk** som sorterer språkene basert på ID-ene.

```
accuracy score: 0.9166909323859996
orderedSpråk {'arabisk': 0, 'islandsk': 1, 'norsk': 2, 'khmer': 3, 'spansk': 4, 'tysk': 5, 'japansk': 6, 'koreansk': 7, 'patwa': 8, 'fransk': 9, 'engelsk': 10, 'rumensk': 11, 'kantonisk': 12, 'malayisk': 13, 'vietnamesisk': 14, 'svensk': 15, 'swahilisk': 16, 'mandarin': 17, 'finsk': 18, 'farsi': 19}
0      1      2      3      4      5      6      7      8      9      10     11     12     13     14     15     16     17     18     19
precision_score 0.952835 0.946841 0.915152 0.980760 0.933126 0.938486 0.977273 ... 0.982143 0.761520 0.930512 0.943020 0.777008 0.948636 0.968416
recall_score    0.939169 0.955949 0.948492 0.996187 0.918367 0.912110 0.929969 ... 0.266990 0.796324 0.901316 0.947067 0.923897 0.914389 0.968909
f1_score        0.945953 0.951373 0.931524 0.988413 0.925688 0.925110 0.953034 ... 0.419847 0.778533 0.915681 0.945039 0.844110 0.931197 0.968662

[3 rows x 20 columns]
f1_score_micro is : 0.9166909323859996
f1_score_macro is : 0.8848039281501446
```

**d) Analyse av modellen**

litt om koden:

- Jeg har opprettet metode ***analyse\_av\_modellen()***. Den finner ut symbolet som bidrar mest til å regne ord som bidrar mest til å klassifisere ord som norsk og symbolet som gjør det motsatte. Metoden henter ID-en som representerer norsk i mapping. Så slår den opp på norsk via modellen sin variabel `coef_`. Etterpå har jeg opprettet to temp-variabler for å hjelpe meg i sammenligning mot vektene(=verdiene som er vektet vha. `model.coef_`). Jeg valgte at ***temp\_høyst*** skulle starte med verdi -1000 for å garantere at ingen av vektene er mindre enn -1000 fordi vektene ser ut som det er mellom -10 og 10. Det motsatte gjelder ***temp\_minst***. Legg merke til at jeg har tatt være på indeksene til minst og høyst vektene slik at jeg kan ekstrahere de høyste og minste trekkene på norsk fra *AlleUniqueSymboler\_liste*.
- Merk også at jeg har skrevet i ***main*** koden som finner antall ganger trekket, som bidrar mest å klassifere ordet som norsk, forkommer på norsk vs. alle andre språk.

Resultatene:

del1:

- a) Det fonetiske symbolet som bidrar mest til å øke sansynnligheten for at et ord er klassifisert som norsk er: 'u'.
- b) Det total antall ganger u kommer med norske ord med 1615 ganger vs. ikke norske ord med 0 ganger. Det vil si at dette symbolet finnes bare i norsk.

del2:

- Det fonetiske symbolet som bidrar mest til å redusere sansynnligheten for at et ord er klassifisert som norsk er: 'a'

## Sekvensmodeller

**Obs.** hvis du vil teste alle mine resultater, vennligst ukommenter `print(skrivUt())` i *main-område*, nederst i py-filen. Jeg har brukt `nor_train.py` for å teste mine metoder. Men har testet dem mot dataene i `nor_test.py` også.

f)

litt om implementasjonen til `get_BIO_sequence()`:

Jeg har opprettet en liste kalt ***bio\_markeringListe*** som har samme lengde som *sentence\_length* og så har fylt den med 'O' er for å starte med fordi ordene som ikke har merkelapp/label er flere enn de har. Så looper metoden over alle elementer(som er tupler) i spans. Og hvis det er B-tag, legges til listen den i riktig plass. Så har jeg opprettet to tellere (kalt *i* og *j*) hvor *i* starter fra I-tag til og ikke med *j*. Dermed hvis det er I-tag, legges de til listen. Til slutt returneres listen.

g)

- Ordene er tilgjengelige i ***vocab*** ( det totale antallet ord i datasettet er 28262).
- labels i datasettet er: {'B-ORG', 'B-PER', 'B-GPE', 'B-EVT', 'B-LOC', 'B-MISC', 'B-DRV', 'B-PROD'}
- label\_counts (lengde er 18) her er de:

{'O': 1, 'start': 14068, 'B-PROD': 599, 'I-PROD': 534, 'B-LOC': 567, 'B-PER': 3632, 'I-PER': 2066, 'B-GPE': 2253, 'B-ORG': 2512, 'B-DRV': 463, 'I-ORG': 815, 'I-DRV': 105, 'B-MISC': 8, 'I-GPE': 139, 'I-LOC': 209, 'B-EVT': 118, 'I-EVT': 60, 'I-MISC': 4}

- Resten av resultatene i denne delen er litt vanskelig å inkludere i rapporten fordi de er store. Derfor vil jeg bare nevne lengden til variablene og ta skjermbilder:
  - transition\_probs og transition\_counts har lengde 60.
  - emisjon\_counts og emisjon\_probs har lengde 5494.

```
transition_counts
('start', 'B-PROD'): 372, ('B-PROD', 'B-PROD'): 54, ('start', 'B-LOC'): 309, ('start', 'B-PER'): 2508, ('B-PER', 'B-PER'): 536, ('start', 'B-GPE'): 1171, ('B-GPE', 'B-PER'): 168, ('B-PER', 'B-GPE'): 288, ('start', 'B-ORG'): 1443, ('B-ORG', 'B-ORG'): 450, ('B-ORG', 'B-PER'): 246, ('B-PER', 'B-ORG'): 428, ('B-PER', 'B-DRV'): 74, ('B-DRV', 'B-PER'): 95, ('start', 'B-DRV'): 299, ('B-ORG', 'B-GPE'): 191, ('B-GPE', 'B-GPE'): 441, ('B-GPE', 'B-DRV'): 23, ('B-DRV', 'B-GPE'): 53, ('B-ORG', 'B-DRV'): 35, ('B-ORG', 'B-MISC'): 1, ('B-MISC', 'B-MISC'): 1, ('B-DRV', 'B-ORG'): 26, ('B-PER', 'B-PROD'): 95, ('B-PROD', 'B-GPE'): 16, ('B-GPE', 'B-ORG'): 110, ('B-ORG', 'B-PROD'): 42, ('B-DRV', 'B-MISC'): 8, ('B-PROD', 'B-DRV'): 11, ('B-PER', 'B-LOC'): 71, ('B-PROD', 'B-PER'): 48, ('B-LOC', 'B-PER'): 27, ('B-ORG', 'B-EVT'): 15, ('B-PER', 'B-EVT'): 18, ('B-EVT', 'B-GPE'): 11, ('B-GPE', 'B-PROD'): 21, ('B-PROD', 'B-EVT'): 2, ('B-LOC', 'B-GPE'): 82, ('start', 'B-EVT'): 64, ('B-GPE', 'B-LOC'): 51, ('B-ORG', 'B-LOC'): 29, ('B-PROD', 'B-ORG'): 34, ('B-EVT', 'B-LOC'): 4, ('B-LOC', 'B-EVT'): 4, ('B-EVT', 'B-EVT'): 1, ('B-GPE', 'B-EVT'): 12, ('B-DRV', 'B-PROD'): 12, ('B-DRV', 'B-DRV'): 11, ('B-LOC', 'B-LOC'): 88, ('B-EVT', 'B-ORG'): 5, ('B-DRV', 'B-EVT'): 2, ('B-PROD', 'B-LOC'): 7, ('B-GPE', 'B-MISC'): 1, ('B-LOC', 'B-ORG'): 16, ('B-EVT', 'B-PER'): 4, ('start', 'B-MISC'): 5, ('B-LOC', 'B-DRV'): 8, ('B-LOC', 'B-PROD'): 3, ('B-EVT', 'B-DRV'): 1, ('B-MISC', 'B-DRV'): 1}

emisjon_counts
('O', 'UNK'): 1, ('B-PROD', 'Parfait'): 2, ('I-PROD', 'à'): 3, ('I-PROD', 'la'): 3, ('I-PROD', 'Mette-Marit'): 2, ('B-PROD', 'Coquilles'): 1, ('I-PROD', 'St.'): 1, ('I-PROD', 'Jacques'): 1, ('I-PROD', 'Prince'): 1, ('I-PROD', 'de'): 5, ('I-PROD', 'Norvege'): 1, ('B-PROD', 'Turbot'): 1, ('I-PROD', 'au'): 1, ('I-PROD', 'Four'): 1, ('I-PROD', 'à'): 6, ('I-PROD', 'purée'): 1, ('I-PROD', 'chou-fleur'): 1, ('B-PROD', 'Filet'): 1, ('I-PROD', 'd'agneau'): 1, ('I-PROD', 'girolles'): 1, ('I-PROD', 'a'): 2, ('I-PROD', 'crème'): 1, ('I-PROD', 'coulis'): 1, ('I-PROD', 'muron'): 1, ('I-PROD', 'arctique'): 1, ('B-PROD', 'Snøfrisk'): 1, ('B-LOC', 'Slottet'): 5, ('B-PER', 'Sonja'): 10, ('B-PER', 'Harald'): 25, ('B-PER', 'Heakon'): 13, ('B-PER', 'Mette-Marit'): 12, ('B-PER', 'Marit'): 15, ('I-PER', 'Tjesssen'): 1, ('B-PER', 'Sven'): 2, ('I-PER', 'O.'): 3, ('I-PER', 'Halby'): 4, ('B-PER', 'Carl'): 18, ('I-PER', 'Gustaf'): 1, ('B-PER', 'Margrethe'): 2, ('I-PER', 'au'): 5, ('I-PER', 'Danmark'): 1, ('B-GPE', 'Islands'): 1, ('B-PER', 'O.'): 1, ('I-PER', 'Rognan'): 1, ('I-PER', 'Grønsmoen'): 1, ('B-PER', 'Silvia'): 1, ('I-PER', 'Sverige'): 1, ('B-PER', 'Henri'): 1, ('I-PER', 'Luxembourg'): 1, ('B-GPE', 'Islands'): 3, ('B-PER', 'Tarja'): 1, ('I-PER', 'Håkon'): 1, ('B-PER', 'Albert'): 1, ('I-PER', 'Monaco'): 2, ('B-PER', 'Sofia'): 2, ('B-PER', 'Willem-Alexander'): 1, ('B-PER', 'Maxima'): 1, ('I-PER', 'Zorreguieta'): 1, ('I-PER', 'Bernadotte'): 1, ('B-PER', 'Märtha'): 13, ('I-PER', 'Louise'): 8, ('I-PER', 'Louises'): 1, ('B-PER', 'Ragnhild'): 2, ('B-PER', 'Astrid'): 2, ('B-PER', 'Espan'): 18, ('B-PER', 'Kristin'): 18, ('I-PER', 'Björnsy'): 1, ('B-PER', 'Jagland'): 2, ('B-ORG', 'Ap.'): 3, ('B-ORG', 'Stortinget'): 65, ('B-ORG', 'Høyne'): 23, ('B-PER', 'Thorbjørn'): 10, ('I-PER', 'Jagland'): 9, ('B-ORG', 'Fr.p.'): 20, ('B-PER', 'Siv'): 10, ('I-PER', 'Jensen'): 10, ('B-PER', 'John'): 25, ('I-PER', 'Alvheim'): 2, ('B-PER', 'Alvheim'): 1, ('B-ORG', 'Arbeiderpartiet'): 20, ('B-ORG', 'Ap.s'): 1, ('I-PER', 'Blankenborg'): 1, ('B-ORG', 'Kr.F.s'): 2, ('B-PER', 'Einar'): 8, ('I-PER', 'Steenstnas'): 1, ('B-ORG', 'NTB'): 31, ('B-ORG', 'Kr.F.'): 4, ('B-ORG', 'Sp.'): 1, ('B-ORG', 'SV'): 32, ('B-PER', 'Rolf'): 2, ('I-PER', 'Reikvam'): 1, ('B-PER', 'Agot'): 1, ('I-PER', 'Valle'): 1, ('B-PER', 'Kjell'): 27, ('I-PER', 'Magne'): 23, ('I-PER', 'Bondevik'): 23, ('B-DRV', 'Fr.p.-forma'): 1, ('I-PER', 'I.'): 12, ('I-PER', 'Hagens'): 3, ('B-ORG', 'Fr.p.s'): 7, ('B-DRV', 'Fr.p.-gruppen'): 2, ('B-PER', 'Bondevik'): 8, ('B-PER', 'Hagen'): 21, ('B-DRV', 'Håe-leader'): 2, ('B-PER', 'Jan'): 25, ('I-PER', 'Peterson'): 5, ('B-DRV', 'Venstre-leader'): 1, ('B-PER', 'Lars'): 10, ('I-PER', 'Sponheim'): 1, ('B-DRV', 'Fr.p.-støtte'): 1, ('B-PER', 'Hagens'): 6, ('B-ORG', 'Aftenposten'): 13, ('B-ORG', 'Bondevik'): 1, ('I-ORG', 'II'): 2, ('B-ORG', 'Fremskrittspartiet'): 6, ('B-ORG', 'Kristelig'): 3, ('I-ORG', 'Folkeparti'): 9, ('B-ORG', 'Venstre'): 16, ('B-DRV', 'Ap.-regjering'): 1, ('I-PER', 'Hagen'): 17, ('B-ORG', 'Stortingets'): 14, ('B-PER', 'Bondeviks'): 1, ('B-PER', 'Om'): 4, ('B-ORG', 'Talibans'): 2, ('B-PER', 'Mohammad'): 1, ('I-PER', 'Omar'): 1, ('B-DRV', 'Taliban-soldatene'): 1, ('B-ORG', 'AIP'): 1, ('B-GPE', 'Kandahar'): 1, ('B-GPE', 'Afghanistan'): 24, ('B-DRV', 'Taliban-ledelsen'): 1, ('B-ORG', 'FN'): 44, ('B-PER', 'Kofi'): 17, ('I-PER', 'Annan'): 17, ('B-PER', 'Annan'): 9, ('B-ORG', 'FNs'): 27, ('B-DRV', 'HIV'): 2, ('I-DRV', '/'): 3, ('I-DRV', 'AIDS-arbeidet'): 1, ('B-ORG', 'Sikkerhetsrådets'): 1, ('B-DRV', 'HIV-smittet'): 1, ('B-DRV', 'HIV-smittede'): 1, ('B-DRV', 'IDS-døden'): 1, ('B-GPE', 'Afrika'): 13, ('B-DRV', 'FN-familien'): 1, ('B-ORG', 'UNAIDS'): 1, ('B-GPE', 'Genève'): 11, ('B-ORG', 'Verdens'): 8, ('I-ORG', 'helseorganisasj'): 2, ('B-PER', 'Gno'): 10, ('I-PER', 'Harlem'): 7, ('I-PER', 'Brundtland'): 6, ('B-ORG', 'UNICEF'): 6, ('I-DRV', 'AIDS-katastrofen'): 1, ('B-MISC', 'HIV'): 1, ('B-MISC', 'AIDS'): 1, ('B-ORG', 'UNHCR'): 3, ('B-GPE', 'Sudan'): 2, ('B-GPE', 'Angola'): 4, ('B-GPE', 'Kongo'): 22, ('B-GPE', 'Eritrea'): 1, ('B-DRV', 'FN-organet'): 1, ('B-ORG', 'IRWA'): 1, ('I-ORG', 'matvareprogram'): 1, ('B-ORG', 'WFP'): 1, ('B-ORG', 'Den'): 96, ('I-ORG', 'Internasjonale'): 5, ('I-ORG', 'domstolen'): 1, ('B-GPE', 'Haag'): 1, ('B-
```

## litt om koden:

- ErFunnet() er en hjelpe metode. Dentar imot en ordbok og en nøkkel som vi er ute etter. Hvis nøkkel er på ordboken, returnerer den true ellers returnerer false. Merk at nøkkel kan være et ord, tag eller en liste. Den hjelper mye å unngå å skrive mange kode linjer for denne metoden.
- Det å finne og legge ordene til **vocab** er bare å loope over hele *'sentence'* som er parameter til metoden.
- Etter løkken, har jeg opprettet en temp-liste kalt **merkelapper\_i\_rekkefølge**. Den hjelper meg med å finne alle transisjoner i setningen. Jeg antar at hver setning, som har minst en label, må ha <start>. Det er derfor jeg har lagt <start> til listen fra begynnelsen. Men i tilfelle en stening ikke har noen label ser vi bort fra <start>. Så kommer hovedløkke. Den lopper innen rekkevidde av *'sentence'* og gjør mange if-else sjekk. Jeg tror det er best at jeg legger detaljerte kommentarer til koden fremfor jeg forklarer koden her slik at du vet hvilken sjekk det er snakk om. Generelt sett, er det tre hovedsjekk inni løkken og en utenfor. Den ene ligger <start> til **merkelapper\_i\_rekkefølge**. Den andre sjekker om det finnes B-tag i setningen, mens den tredje sjekker I-tag. Den som er utenfor løkken, lager transisjoner (hvis det er minst en label i setningen, i tillegg til <start>).
- **Obs.** Med tanke på ord som ikke har merkelapp, har jeg lagt ('o' : 1) til **label\_count** for å unngå å få feilmelding opp på skjermen.

## h) Sannsynligheterfordelinger

- For transition\_probs, Har jeg delt verdiene til transisjoner på/ total summen til labelen vi er interessert å få informasjon om (label\_counts[key[0]]). Men jeg har en enannen tilnærnering hvor jeg bare deler verdien til transisjoner og delt en hver av dem på det totale antallet for alle transisjoner.
- Når det gjelder emission\_probs, har jeg slavisk fulgt oppskriften som står i oppgaveteksten.
- resultatene jeg har fått:

```
transition_probs
{('start', 'B-PROD'): 0.026442991185669604, ('B-PROD', 'B-PROD'): 0.09015025041736227, ('start', 'B-LOC'): 0.021964742678419108, ('start', 'B-PER'): 0.17827694057435314, ('B-PER', 'B-PER'): 0.14757709251101322, ('start', 'B-GPE'): 0.08323855558714814, ('B-GPE', 'B-PER'): 0.07456724367509987, ('B-PER', 'B-GPE'): 0.07929515418502203, ('start', 'B-ORG'): 0.10257321580892806, ('B-ORG', 'B-ORG'): 0.17914012738853502, ('B-ORG', 'B-PER'): 0.09792993630573249, ('B-PER', 'B-ORG'): 0.11784140969162996, ('B-PER', 'B-DRV'): 0.020374449393207047, ('B-DRV', 'B-PER'): 0.20518358531317496, ('start', 'B-DRV'): 0.02125390958203014, ('B-ORG', 'B-GPE'): 0.07603503184713375, ('B-GPE', 'B-GPE'): 0.0.9573901464713714, ('B-GPE', 'B-DRV'): 0.010208610741233911, ('B-DRV', 'B-GPE'): 0.11447084233261338, ('B-ORG', 'B-DRV'): 0.01393312101910828, ('B-ORG', 'B-MISC'): 0.000398391719745223, ('B-MISC', 'B-LOC'): 0.125, ('B-DRV', 'B-ORG'): 0.056155507559395246, ('B-PER', 'B-PROD'): 0.026156387665198236, ('B-PROD', 'B-GPE'): 0.02671118530884808, ('B-GPE', 'B-PROD'): 0.048823790591553485, ('B-ORG', 'B-PROD'): 0.016719745222929936, ('B-DRV', 'B-LOC'): 0.017278617710583154, ('B-PROD', 'B-DRV'): 0.01836393899833055, ('B-PER', 'B-LOC'): 0.019548458149779735, ('B-PROD', 'B-PER'): 0.0801335592654424, ('B-LOC', 'B-PER'): 0.047619047619047616, ('B-ORG', 'B-EVT'): 0.005971337579617835, ('B-PER', 'B-EVT'): 0.004959947136563877, ('B-EVT', 'B-GPE'): 0.093220338983050885, ('B-GPE', 'B-PROD'): 0.009320905459387484, ('B-PROD', 'B-EVT'): 0.00333889816360601, ('B-LOC', 'B-GPE'): 0.14462081128747795, ('start', 'B-EVT'): 0.004549331816889394, ('B-GPE', 'B-LOC'): 0.022636484687083888, ('B-ORG', 'B-LOC'): 0.011544585987261146, ('B-PROD', 'B-ORG'): 0.05676126878130217, ('B-EVT', 'B-LOC'): 0.03389838508474576, ('B-LOC', 'B-EVT'): 0.007054673721340368, ('B-EVT', 'B-EVT'): 0.00847457627118644, ('B-GPE', 'B-EVT'): 0.953231691070562, ('B-DRV', 'B-PROD'): 0.02591792055587473, ('B-DRV', 'B-DRV'): 0.023758099352051837, ('B-LOC', 'B-LOC'): 0.15520282186948853, ('B-EVT', 'B-ORG'): 0.04237881359322, ('B-DRV', 'B-EVT'): 0.004319654427645789, ('B-PROD', 'B-LOC'): 0.011686143572621035, ('B-GPE', 'B-MISC'): 0.0004438526409232135, ('B-LOC', 'B-ORG'): 0.028218698836155, ('B-EVT', 'B-PER'): 0.03389830508474576, ('start', 'B-MISC'): 0.00035541654819448395, ('B-LOC', 'B-DRV'): 0.014109347442680775, ('B-LOC', 'B-PROD'): 0.00529180521005291, ('B-EVT', 'B-DRV'): 0.00847457627118644, ('B-MISC', 'B-DRV'): 0.125}
```

```
emission_probs
{('O', '<UNK>'): 1.028263, ('B-PROD', 'Parfait'): 0.03160089983305509, ('I-PROD', 'à'): 0.03387997940074906, ('I-PROD', 'la'): 0.03387997940074906, ('I-PROD', 'Mette-Marit'): 0.0320073202247191, ('B-PROD', 'Coquilles'): 0.029931450751252087, ('I-PROD', 'St.'): 0.030134661048689136, ('I-PROD', 'Jacques'): 0.030134661048689136, ('I-PROD', 'Prince'): 0.030134661048689136, ('I-PROD', 'de'): 0.037625297752808984, ('I-PROD', 'Norvege'): 0.030134661048689136, ('B-PROD', 'Turbot'): 0.029931450751252087, ('I-PROD', 'a'): 0.030134661048689136, ('I-PROD', 'Four'): 0.030134661048689136, ('I-PROD', '-'): 0.03949795692883895, ('I-PROD', 'purée'): 0.030134661048689136, ('I-PROD', 'chou-fleur'): 0.030134661048689136, ('B-PROD', 'Filet'): 0.029931450751252087, ('I-PROD', 'd'agneau'): 0.030134661048689136, ('I-PROD', 'girolles'): 0.030134661048689136, ('I-PROD', 'a'): 0.0320073202247191, ('I-PROD', 'creme'): 0.030134661048689136, ('I-PROD', 'coulis'): 0.030134661048689136, ('I-PROD', 'muron'): 0.030134661048689136, ('I-PROD', 'arctique'): 0.030134661048689136, ('B-PROD', 'Snøfrisk'): 0.029931450751252087, ('B-LOC', 'Slottet'): 0.037080343915343913, ('B-PER', 'Sonja'): 0.031015304240088103, ('B-PER', 'Harald'): 0.03514526018722467, ('B-PER', 'Haakon'): 0.031841295429515415, ('B-PER', 'Mette-Marit'): 0.03156596503303965, ('B-PER', 'Marit'): 0.03239195622246696, ('I-PER', 'Tjessem'): 0.028746027589545014, ('B-PER', 'Sven'): 0.02881266106828194, ('I-PER', 'O.'): 0.02971408180058083, ('I-PER', 'Håiby'): 0.03019810890609874, ('B-PER', 'Carl'): 0.03321794741189427, ('I-PER', 'Gustaf'): 0.028746027589545014, ('B-PER', 'Margrethe'): 0.02881266106828194, ('I-PER', 'av'): 0.03068213601161665, ('I-PER', 'Danmark'): 0.028746027589545014, ('B-GPE', 'Islands'): 0.028705853084775853, ('B-PER', 'Olafur'): 0.028537330671806166, ('I-PER', 'Ragnar'): 0.028746027589545014, ('I-PER', 'Grimsson'): 0.028746027589545014, ('B-PER', 'Silvia'): 0.028537330671806166, ('I-PER', 'Sverige'): 0.028746027589545014, ('B-PER', 'Henri'): 0.028537330671806166, ('I-PER', 'Luxembourg'): 0.028746027589545014, ('B-GPE', 'Finlands'): 0.02959355836662228, ('B-PER', 'Tarja'): 0.028537330671806166, ('I-PER', 'Hälönen'): 0.028746027589545014, ('B-PER', 'Albert'): 0.028537330671806166, ('I-PER', 'Monaco'): 0.029230054695062922, ('B-PER', 'Sofia'): 0.02881266106828194, ('B-PER', 'William-Alexander'): 0.028537330671806166, ('B-PER', 'Maxima'): 0.028537330671806166, ('I-PER', 'Zorrreguieta'): 0.028746027589545014, ('I-PER', 'Bernadotte'): 0.028746027589545014, ('B-PER', 'Martha'): 0.031041295429515415, ('I-PER', 'Louise'): 0.03213421732817037, ('I-PER', 'Louises'): 0.028746027589545014, ('B-PER', 'Ragnhild'): 0.02881266106828194, ('B-PER', 'Espen'): 0.03321794741189427, ('B-PER', 'Kristin'): 0.03321794741189427, ('I-PER', 'Bjørnøy'): 0.028746027589545014, ('B-PER', 'Jagland'): 0.02881266106828194, ('B-ORG', 'Ap.'): 0.02945626791401274, ('B-ORG', 'Stortinget'): 0.05413779657643312, ('B-ORG', 'Høyne'): 0.037418051353503185, ('B-PER', 'Thorbjørn'): 0.031015304240088103, ('I-PER', 'Jagland'): 0.032618244433688284, ('B-ORG', 'Fr.p.'): 0.0362237838375962, ('B-PER', 'Siv'): 0.031015304240088103, ('I-PER', 'Jensen'): 0.033102271539206196, ('B-PER', 'John'): 0.03514526018722467, ('I-PER', 'Alvheim'): 0.029230054695062922, ('B-PER', 'Alvheim'): 0.028537330671806166, ('I-PER', 'Jensen'): 0.02881266106828194, ('B-ORG', 'Arbeiderpartiet'): 0.0362237838375962, ('B-ORG', 'Ap.s.'): 0.028660089570063694, ('I-PER', 'Blankenborg'): 0.028746027589545014, ('B-ORG', 'Kr.F.s.'): 0.029058178742088216, ('B-PER', 'Einar'): 0.030464643447136563, ('I-PER', 'Steensnas'): 0.028746027589545014, ('B-ORG', 'NTB'): 0.040602764729299363, ('B-ORG', 'Høyne'): 0.02985435708598706, ('B-ORG', 'Sp.'): 0.028660089570063694, ('B-ORG', 'SV'): 0.041000853901273886, ('B-PER', 'Rolf'): 0.02881266106828194, ('I-PER', 'Reikvam'): 0.028746027589545014, ('B-P-DRV', 'Ågot'): 0.028537330671806166, ('I-PER', 'Vallé'): 0.028746027589545014, ('B-PER', 'Kjell'): 0.03569592098017621, ('I-PER', 'Magne'): 0.03939462391093901, ('I-PER', 'Bondevik'): 0.038910596805421103, ('B-DRV', 'Fr.p.-Formann'): 0.03042182937365011, ('I-PER', 'I.'): 0.03407032575024201, ('I-PER', 'Hagens'): 0.02971408180058083, ('B-ORG', 'Fr.p.s.'): 0.031041295429515415, ('B-ORG', 'Lars'): 0.031015304240088103, ('B-PER', 'Sponheim'): 0.028746027589545014, ('B-DRV', 'Fr.p.-støtte'): 0.03042182937365011, ('B-PER', 'Hagens'): 0.02991398265418502, ('B-ORG', 'Aftenposten'): 0.03343715963375796, ('B-ORG', 'Bondevik'): 0.028660089570063694, ('I-ORG', 'II'): 0.030715988957055212, ('B-ORG', 'Fremskrittspartiet'): 0.039650535429936305, ('B-ORG', 'Kristelig'): 0.02945626791401274, ('I-ORG', 'Folkeparti'): 0.039304946012269935, ('B-ORG', 'Venstre'): 0.03463142714968153, ('B-DRV', 'Ap.-regjering'): 0.03042182937365011, ('I-PER', 'Hagen'): 0.03649046127783156, ('B-ORG', 'Stortingets'): 0.03383524880573248, ('B-PER', 'Bondeviks'): 0.028537330671806166, ('B-PER', 'Bondevik')
```