Asemsel awurman n tutlayt

D tayult n tsenselkimt d tigzi n tmacint i d-ilehhun deg temyigawt gar umdan akked tmacint. Tayult-a tsemras s wudem awurman ilugan i tesbadu tesnilest. Telha-d ladya s :

```
    Aεqal n taγect

- Gezzu n tutlayt n umdan

    Asnulfu n tutlayt n umdan

Yal asemsel yelha-d deg yiwet n tayult n tesnilest. Amedya: acrad asnalyan aseddasan yelha-
d deg tesledt n talya akked tseddast, ama d awalen ney tifyar.
```

Assay gar tigzi n tmacint d tesnilest Asemsel awurman n tutlayt d tayult i d-yekkan seg usdukkel

n tesnilest d tigzi n tmacint. D asemres n yilugan n tesnilest s useqdec n uselkim, i d-yesnulfan tayult n usemsel awurman n tutlayt. Aya yegber merra tiqula n tesnilest: Seg yimesla arma d inumak n yedrisen

Assay gar tigzi n tmacint, almad awurman, almad leqqayen

Tigzi n tmacint d aḥric meggren deg tsenselkimt. Tigzi n tmacint d aseqdec n usihel deg yinurar n tussniwin ama i tesledt n yisefka, ney anadi n tifrat i ddeqs n wuguren i icudden

yer tudert n umdan. Aḥric-a tikwal yessemras titiknikin n usluymu n yisefka. D wagi iwumi ggaren almad awurman. Ticki tatiknikt tessemras izedwan n yinurunen, ahric-a ggaren-as almad leqqayen

1. Tagrumma n yisefka isnalyanen iseddasen: Yal awal ney aferdis ilelli deg tefyirt, teṭṭafar-it-id tecredt-is ney asmel-is anejrum. Aḥric-a ur d-ilehhu ara deg twuri n wawalen, maca deg usmil-is kan anejrum.

Acrad asnalyan aseddasan i tutlayt taqbaylit:

Amedya: Yefka/VP -as/PSV -t/PSV i/PRP yemma/NMC -s/PAN ./. Anida:

 PRP: Préposition ney tanzeyt NMC: Nom commun ney isem amagnu.

PSV: Pronom préfixe de verbe ney amqim udfir n umyag.

1. Awal s timmad-is.

• PAN: Pronom affixe de nom ney amgim awsil n yisem

2. Ma yezga-d wawal deg tazwara n tefyirt.

VP: Verbe prétérit ney amyag izri

Talyiwin n wawal d tseddast n tefyir

3. Ma yezga-d deg taggara n tefyirt. 4. Ma ibeddu wawal s usekkil meggren. 5. Ma yura i merra s usekkil meqqren.

6. Ma yura i merra s usekkil mezziyen.

Akken ad d-naf asmil anejrum n wawal deg tefyirt n tutlayt tagbaylit, tlag tesledt tasnalyant taseddasant. Gar talyiwin-a ad d-naf:

```
7. 1 usekkil seg tazwara n wawal.
  8. 2 isekkilen seg tazwara n wawal.
  9. 3 isekkilen seg tazwara n wawal.
 10. 4 isekkilen seg tazwara n wawal.
 11. 5 isekkilen seg tazwara n wawal.
 12. 6 isekkilen seg tazwara n wawal.
 13. 1 usekkil seg taggara n wawal.
 14. 2 isekkilen seg taggara n wawal.
 15. 3 isekkilen seg taggara n wawal.
 16. 4 isekkilen seg taggara n wawal.
 17. 5 isekkilen seg taggara n wawal.
 18. Awal uzwir amenzu (i d-yezgan uqbel).
 19. Awal uzwir wis sin (i d-yezgan uqbel n wawal uzwir menzu).
 20. Awal uḍfir amezu (i d-yezgan mbeεd).
 21. Awal uḍfir wis sin (i d-yezgan mbeεd n wawal ufir amenzu).
 22. Ma yegber kan izwilen.
 23. Ma yegber asekkil megqren daxel-is
 24. Ma yegber asekkil ur nelli ara deg ugemmay neş deg usenqed(@,$,%,£ ...atg).
Amedya:
```

Tayara agmud

Awal s timmad-is tettwabder

Ma yezga-d wawal deg tazwara n tefyirt. False

Ma yezga-d deg taggara n tefyirt. False

Ma ibeddu wawal s usekkil meqqren. False

Ma yura i merra s usekkil meqqren. False Ma yura i merra s usekkil mezziyen. True

Tiziri tettwabder-d deg tesdawsit ass-a.

Deg tefyirt ddaw-a, tasledt n wawal tettwabder, ad tefk:

```
1 usekkil seg tazwara n wawal. t
                                    2 isekkilen seg tazwara n wawal. te
                                    3 isekkilen seg tazwara n wawal. tet
                                    4 isekkilen seg tazwara n wawal. tett
                                    5 isekkilen seg tazwara n wawal. tettw
                                    6 isekkilen seg tazwara n wawal. tettwa
                                     1 usekkil seg taggara n wawal. r
                                    2 isekkilen seg taggara n wawal. er
                                    3 isekkilen seg taggara n wawal. der
                                    4 isekkilen seg taggara n wawal. bder
                                    5 isekkilen seg taggara n wawal. abder
                               Awal uzwir amenzu (i d-yezgan uqbel). Tiziri
               Awal uzwir wis sin (i d-yezgan uqbel n wawal uzwir menzu). False
                                Awal udfir amezu (i d-yezgan mbeɛd). -d
               Awal udfir wis sin (i d-yezgan mbeed n wawal ufir amenzu). deg
                                           Ma yegber kan izwilen. False
                                 Ma yegber asekkil meggren daxel-is False
     Ma yegber asekkil ur nelli ara deg ugemmay neş deg usenqed(@,$,%,£
Timsisyal
  1. Idlisen
    <img src="img1.jpg"/>
        <img src="img2.jpg"/>
```


In [6]: udfiren=[]

uzwiren=[]


```
A. Timsirin srid

    Almad awurman s Pytorch: https://classroom.udacity.com/courses/ud188

    Almad awurman s tensorFlow: https://developers.google.com/machine-learning/crash-course/ml-intro?hl=fr

               Almad awurman s Octave (Fortran): https://www.coursera.org/learn/machine-learning/
             A. Tiyerarin:

    NLTK

    Scikit-Learn

           Bettu asnalyan aseddasan
           Beṭṭu asnalɣan aseddasan, d asemsel awurman uzwir n tutlayt ara yeǧǧen aselkim ad yebḍu tifyar ɣer wawalen
           ilelliyen akken ad yizmir ad d-yaf asmil anejrum n yal awal ney ayen iwumi neqqar: acrad asnalyan aseddasan.
           Amedya: Ad as-t-id-yawi. -> Ad as- t- id- yawi . Yewwi-as-ten-id. -> Yewwi -as -ten -id .
           Algiritm n ddaw-a, ad as-nefk afaylu n tefyar s teqbaylit, netta ad ay-d-yerr afaylu anida ya tefyirt tebda yer
           wawalen.
           Amek i iteddu beţţu asnalyal aseddasan?
```

uzwiren.append(str(a).upper())

while awal.find('-')>=0:

while awal.find('-')>=0:

awal=awal[awal.find('-')+1:len(awal)] awal_yebdan=awal_yebdan+' '+amurfim

awal_yebdan=awal_yebdan+' '+awal

amurfim=awal[0:awal.find('-')] awal yebdan=awal yebdan+' '+amurfim awal=awal[awal.find('-')+1:len(awal)]

amurfim=awal[0:awal.find('-')+1] awal=awal[awal.find('-')+1:len(awal)] awal_yebdan=awal_yebdan+' '+amurfim

amurfim=awal[0:awal.find('-')]

Acrad asnalyan aseddansan

#Construction du texte global à aprtir du corpus étiqueté

for ligne in open("corpus-kab.txt",encoding='utf-8'):

A. Alguritm CRF - lbfgs - sklearn_crfsuite

from nltk.tag.util import untag

line=ligne.split() taille=len(line) for i in line:

> j=i.split('/') couple=(j[0],j[1]) sentence.append(couple)

taille2=taille+taille2

print("Amḍan n yiferdisen: ", taille2)

def features(sentence, index):

#Définition des caractériqtuques d'un mot

tagged_sentences.append(sentence)

print("Amdan n tefyar: ", len(tagged_sentences))

In [1]: | from sklearn_crfsuite import CRF

tagged_sentences=[]

taille=0

first=1

if (first!=0): sentence=[]

first=0

awal_yebḍan=''

else:

if (amurfim in uzwiren):

ufaylu affixescolles.txt for i in open("imqimen_iwsilen.txt",encoding='utf-8'): a=i.replace("\ufeff","").replace("\n","").strip() if (a[len(a)-1]=="-"): uzwiren.append(str(a).lower())

Alguritm ad yeddem afaylu brut_text.txt, ad yebdu awalen, sakin ad yeslal afaylu tokenazed_text.txt

d udifr) nev deg taggara (acku d uzwir): -ak -t .. d udfiren. ak- t- ... d uzwiren.

• Deg tazwara, merra imqimen iwşilen, ama d wid n yimyagen, n tzelşiwin, n yismawen neş wiyad, ad ten-nessers deg ufaylu n yimqimen iwşilen : imqimen_iwsilen.txt. Yal amqim ad asen-nseddu tizdit ama deg tazwara (acku

Asenqeḍ=['...',',',';','?','!',':','"','(',')','*','_',','[',']','{','}','\"','\"','\"']

#ad d-nekkes akk imqimen, tizelyiwin ... i yettentaden d yimyagen, awalen, tinzay.....atg seg

udfiren.append(str(a).lower()) udfiren.append(str(a).upper()) # tawuri-a tbeţţu awal def bdu_awal(awal,udfiren,uzwiren): amurfim=awal[0:awal.find('-')+1]

```
awal_yebḍan=awal_yebḍan+' '+'-'+amurfim
           awal=awal[awal.find('-')+1:len(awal)]
   if ('-'+awal in udfiren):
        awal_yebdan=awal_yebdan+' '+'-'+awal
        awal_yebḍan=awal_yebḍan
    return awal_yebdan
# tawuri-a tbeţţi tafyirt
def bdu_tafyirt(sentence,udfiren,uzwiren):
      a=sentence.split()
      tafyirt1=""
      for i in a: #mots
       if(i.find('-')<0):</pre>
           tafyirt1=tafyirt1+' '+i
           awals=bdu_awal(i,udfiren,uzwiren)
           tafyirt1=tafyirt1+' '+awals
      tafyirt1=tafyirt1.strip()
      return tafyirt1
#afaylu n tefyar
f= open("tokenized_text.txt","w+",encoding='utf-8')
# Afaylu n tefyar yebḍan
g=open("brut_text.txt",encoding='utf-8')
for adur in g:
       adur=adur.replace(i,' '+i+' ').replace("\ufeff","")
   izirig=bdu_tafyirt(adur,udfiren,uzwiren)
   izirig=izirig.replace(" "," ")
    print(izirig)
   f.write(izirig+'\n')
f.close()
g.close()
print (" ")
print (" ")
print ("-----")
print ("Yedda akken iwata!! Bettu asnalyan yemmed. ")
As -D ass -A.
Azul fell -awen .
Ad as- t- id- awiy ma yebya -tent .
Si tama -nniḍen , axeddim -a ad d- yeglu s kra n ubayur meqqren i leqdic -nniḍen ara iḍerrun
akka tura deg umaḍal umḍin .
Tirebbaε n usideg i yekkren i tmerna n teqbaylit ɣer tkarḍa tamaḍalant tillellit OpenStreetMa
p , ad izmiren ad sqedcen awalen -a yer tama n yismawen n tuddar , isaffen , iberdan , akken
azekka ad naf taqbaylit deg wallalen am GPS , ladya akka tura imi yezmer kra n wagguren d asa
wen , ad nernu dayen tayect taqbayit .
Ittusemma , amdan ad yizmir ad isexdem GPS s teqbaylit xersum di tazwara di tmurt n Leqbayel
Ass -a , tuget n yimdanen , seqdacen allalen atraren i yesseqdacen titiknulujiyin timaynutin
am uselkim , tiliyriyin .
Yef waya , acḥal d tikkelt ara d- neggar tiɣri i yimdanen akken ad suzuren s ṭaqa axeddim -ns
en deg Internet , ad d- bettun tiktiwin d tmussniwin -nsen i yilmezyen ara igeddcen deg unnar
n usideg n yifecka new deg unnar n usemsel awurman n tutlayt .
Sin -a n yiḥricen ineggura , ad ǧǧen i yimdanen ad sqedcen imeẓla ama deg uselkim neɣ deg Int
ernet s tutlayt -nsen tayemmat war ma terra -ten tmara ad smersen tutlayin tiberraniyin .
Deg taggara , ad snemmrey Aymis n Yimaziyen i yettilin yal tikkelt deg unnar akken ad d- yawi
isallen i icudden yer leqdic yezdin i tutlayt taqbaylit .
Ad snemmrey dayen Mass Rabia Farid i yellan yef uqerru n tseqqamutt n usmuzzey n twennadt tad
eblant deg ugezdu n Tizi Uzzu , imi i ay- d- yefka tagnit akken ad d- nmudd ayen nessen , ney
ad d- nerfed tamussni seg yimussnawen yellan deg unnar -a .
Nessaram ad d- derrunt temliliyin am tigi s waţas .
Yedda akken iwata!! Beţţu asnalɣan yemmed.
```

```
'word': sentence[index],
                                   # Awal s timmad-is
        'is_first': index == 0,
                                  # Ma yezga-d deg tazwar n tefyirt
        'is_last': index == len(sentence) - 1, # Ma yezgma-d deg taggar n tefyirt
        'is_capitalized': sentence[index][0].upper() == sentence[index][0], # MA ibeddu s usekkil
meggren
        'is_all_caps': sentence[index].upper() == sentence[index], # Ma yura meṛṛa s usekkil
        'is_all_lower': sentence[index].lower() == sentence[index], # ma yura meṛṛa s usekkil
        'prefix-1': sentence[index][0], #1 usekkil uzwir
        'prefix-2': sentence[index][:2], #2 isekkilen uzwiren
        'prefix-3': sentence[index][:3], #3 isekkilen uzwiren
        'prefix-4': sentence[index][:4], #4 isekkilen uzwiren
        'prefix-5': sentence[index][:5], #5 isekkilen uzwiren tettecmumuḥenḍ (aoriste intensif)
        'suffix-1': sentence[index][-1], #1 usekkil udfir
        'suffix-2': sentence[index][-2:], #2 isekkilen udfiren
        'suffix-3': sentence[index][-3:], #3 isekkilen uḍfiren
        'suffix-4': sentence[index][-4:], #4 isekkilen uḍfiren
        'suffix-5': sentence[index][-5:], #5 isekkilen udfiren
        'prev_word': '' if index == 0 else sentence[index - 1], #awal uzwir
        'prev1_word': '' if index == 1 else sentence[index - 2], #awal uzwir
        'next_word': '' if index == len(sentence) - 1 else sentence[index + 1], #awal uḍfir
        'is_numeric': sentence[index].isdigit(), #ma yegber kan izwilen
        'capitals_inside': sentence[index][1:].lower() != sentence[index][1:] #ma yegber asekkil
meggren daxel-is
#transformation du corpus x: contient les tokens et y les tags
def transform_to_dataset(tagged_sentences):
   X, y = [], []
    for tagged in tagged_sentences:
        X.append([features(untag(tagged), index) for index in range(len(tagged))])
        y.append([tag for _, tag in tagged])
    return X, y
total=int(len(tagged_sentences)*0.80)
X,y=transform_to_dataset(tagged_sentences)
X_train, y_train = transform_to_dataset(tagged_sentences[:total])
X_test, y_test = transform_to_dataset(tagged_sentences[total:])
#déclaration du modèle suivant l'algotihme lbfgs
model = CRF(
    algorithm='lbfgs', # Limited-memory Broyden-Fletcher-Goldfarb-Shanno Algorithm.
                       #Used to pos tag words or other information provided by the model within the
situation.
   c1=0.01,
    c2=0.1,
    max_iterations=100,
    all_possible_transitions=True
#entrainement
model.fit(X, y)
#y_pred = model.predict(X_test)
#labels = list(model.classes_)
#print(labels)
#metrics.flat_f1_score(y_test, y_pred,
                       average='weighted', labels=labels)
# Sauvegarde du modèle
from joblib import dump, load
dump(model, 'model-lbfgs.joblib')
print('Aslal n tnewruft tasnalwant tasedasant yemmed')
Amḍan n tefyar: 1633
Amḍan n yiferdisen: 18383
Aslal n tnewruft tasnalwant tasedasant yemmed
```

In []: 2. Alguritm

