

Ex1 Nooj

```
NooJ V7
# Inflectional/Derivational description
#
# Language is: fr
#
# Generic Commands:
# <B>: keyboard Backspace
# <C>: change Case
# <D>: Duplicate current char
# <E>: Empty string
# <L>: keyboard Left arrow
# <N>: go to end of Next word form
# <P>: go to end of Previous word form
# <R>: keyboard Right arrow
# <S>: delete/Suppress current char
# Arguments for commands <B>, <L>, <N>, <P>, <R>, <S>:
# xx number: repeat xx times
# W: whole word
# Examples
# <R3>: go right 3 times
# <LW>: go to beg. of word
#
# Language-Specific Commands:
# (None)
#
# Special Characters: '=' '<' '>' '\' '"' ':' '|' '+' '-' '/' '$' '_' ';' '#'
#
Avertir=<E>/INF | <B>s/PR+s+1 | <B>s/PR+s+2 | <B>t/PR+s+3 | <B>ssons/PR+p+1 | <B>ssez/PR+p+2 | <B>ssent/PR+p+3;
Craindre=<E>/INF | <B3>s/PR+s+1 | <B3>s/PR+s+2 | <B3>t/PR+s+3 | <B4>gnons/PR+p+1 | <B4>gnez/PR+p+2 | <B4>gnent/PR+p+3;
```

[illegible]

Compte rendu

The screenshot shows a Windows desktop environment. A Notepad++ window is open, displaying a dictionary file named 'dickhouloud.dic'. The file contains a list of French verbs and their conjugations, along with metadata like 'Special Command', 'Special Features', and 'Special Characters'. A Windows Firewall notification is visible in the bottom right corner, indicating that Windows Firewall is active and blocking some connections. The taskbar at the bottom shows various application icons, including the Start button, File Explorer, Google Chrome, and several other programs. The system clock in the bottom right corner shows the time as 18:50 on 25/12/2020.

NooJ - [C:\Users\users\Documents\NooJ\fr\Lexical Analysis\dickhouloud.dic]

File Edit Lab Project Windows Info DICTIONARY

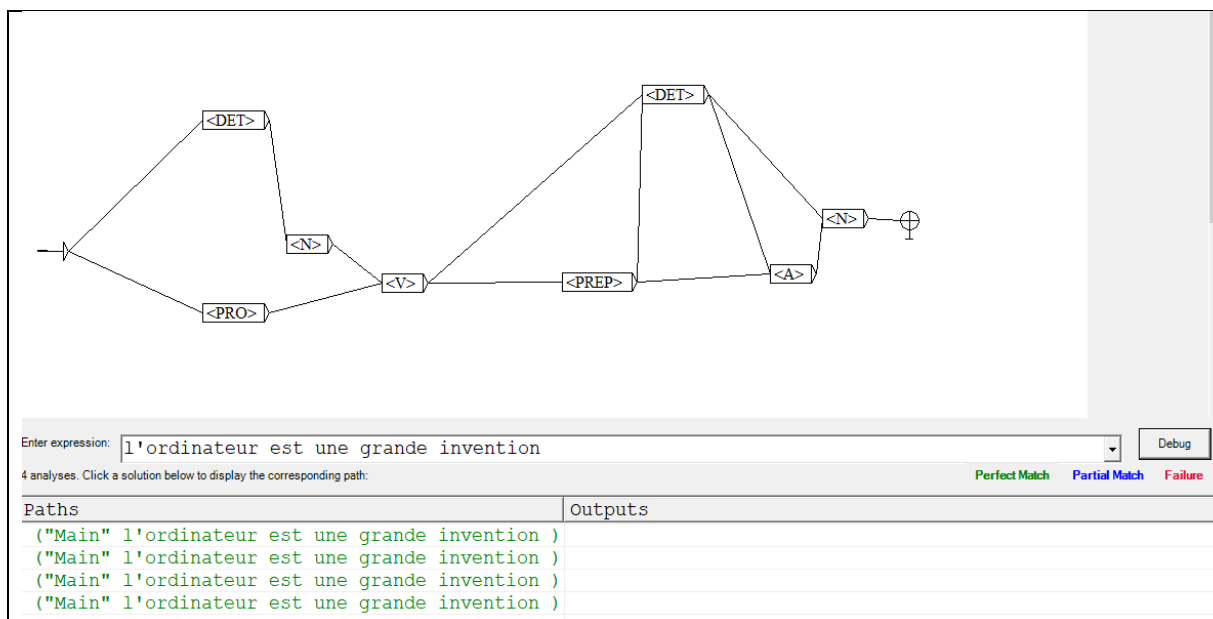
```
#
# Use inflectional & derivational paradigms' description files (.nof), e.g.:
# Special Command: #use paradigms.nof
#
# Special Features: +NW (non-word) +FXC (frozen expression component) +UNAMB (unambiguous lexical entry)
#                  +FLX= (inflectional paradigm) +DRV= (derivational paradigm)
#
# Special Characters: '\' '"' ' ' ',' '+' '-' '#'
#
# This dictionary was automatically built from C:\Users\users\Documents\NooJ\fr\Lexical Analysis\dickhouloud.dic
#
#use noojkouloud.nof
#use _Exemple.nof
#
réussir,réussir,V+FLX=Avertir+INF
réussis,réussir,V+FLX=Avertir+PR+s+1
réussis,réussir,V+FLX=Avertir+PR+s+2
réussit,réussir,V+FLX=Avertir+PR+s+3
réussissons,réussir,V+FLX=Avertir+PR+p+1
réussissez,réussir,V+FLX=Avertir+PR+p+2
réussissent,réussir,V+FLX=Avertir+PR+p+3
éteindre,éteindre,V+FLX=Craindre+INF
éteignons,éteindre,V+FLX=Craindre+PR+p+1
éteignez,éteindre,V+FLX=Craindre+PR+p+2
éteignent,éteindre,V+FLX=Craindre+PR+p+3
éteins,éteindre,V+FLX=Craindre+PR+s+1
éteins,éteindre,V+FLX=Craindre+PR+s+2
éteint,éteindre,V+FLX=Craindre+PR+s+3
```

3.6 sec Cancel

Activer Windows
Accédez aux paramètres de l'ordinateur pour activer Windows.

Windows taskbar icons: Start, File Explorer, Google Chrome, Microsoft Edge, X, N, O, W, 18:50 25/12/2020

Ex2 Nooj



Compte rendu

The diagram shows a finite state automaton (FSA) with the following states and transitions:

- Start state (indicated by an arrow) transitions to a state labeled **<DET>**.
- From **<DET>**, there is a transition to a state labeled **<PRO>** and another to a state labeled **<N>**.
- From **<PRO>**, there is a transition to a state labeled **<V>**.
- From **<N>**, there is a transition to a state labeled **<V>**.
- From **<V>**, there is a transition to a state labeled **<PREP>**.
- From **<PREP>**, there is a transition to a state labeled **<DET>**.
- From **<DET>**, there is a transition to a state labeled **<A>**.
- From **<A>**, there is a transition to a state labeled **<N>**.
- From **<N>**, there is a transition to the final state (indicated by a double circle).

Enter expression:

analyses. Click a solution below to display the corresponding path: Perfect Match Partial Match Failure

Paths	Outputs
("Main" la faculté accueille les étudiants)	
("Main" la faculté accueille les étudiants)	
("Main" la faculté accueille les étudiants)	
("Main" la faculté accueille les étudiants)	
("Main" la faculté accueille les étudiants)	

The diagram shows a finite state automaton (FSA) with the following states and transitions:

- Start state (indicated by an arrow) transitions to a state labeled **<DET>**.
- From **<DET>**, there is a transition to a state labeled **<PRO>** and another to a state labeled **<N>**.
- From **<PRO>**, there is a transition to a state labeled **<V>**.
- From **<N>**, there is a transition to a state labeled **<V>**.
- From **<V>**, there is a transition to a state labeled **<PREP>**.
- From **<PREP>**, there is a transition to a state labeled **<DET>**.
- From **<DET>**, there is a transition to a state labeled **<A>**.
- From **<A>**, there is a transition to a state labeled **<N>**.
- From **<N>**, there is a transition to the final state (indicated by a double circle).

Enter expression:

10 analyses. Click a solution below to display the corresponding path: Perfect Match Partial Match Failure

Paths	Outputs
("Main" il téléphone à son ami)	
("Main" il téléphone à son ami)	
("Main" il téléphone à son ami)	
("Main" il téléphone à son ami)	
("Main" il téléphone à son ami)	

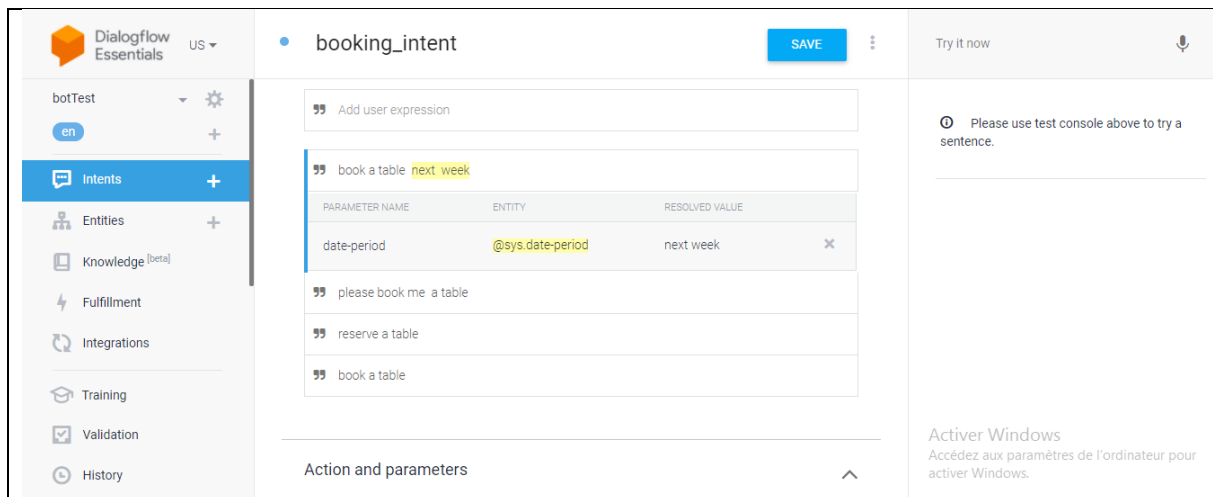
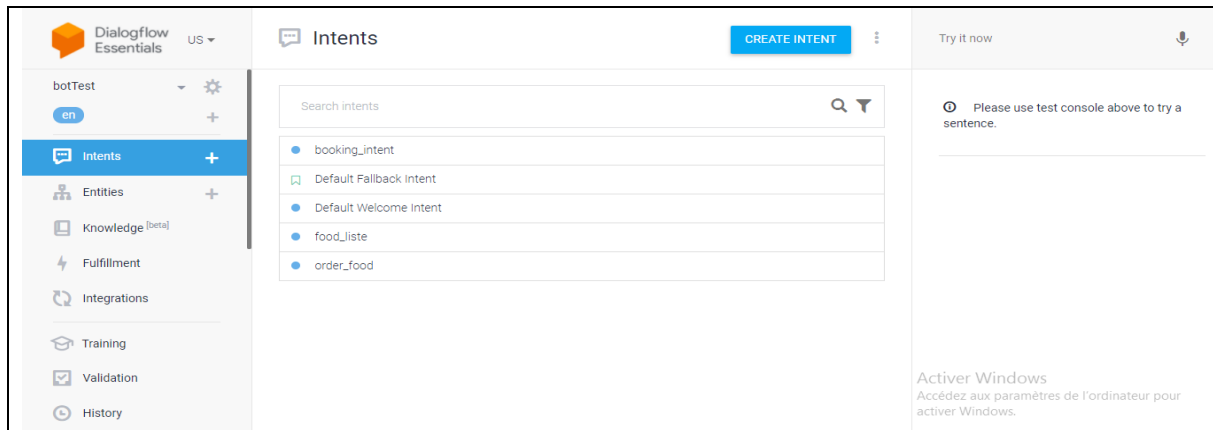
Compte rendu

Ex1 chatbot dialogflow :

Vidéo : Dans l'archive

Lien : <https://bot.dialogflow.com/297438b5-f81a-4e20-9568-ac51836022be>

J'ai créé un bottest qui permet de réserver une table à la semaine prochaine



Compte rendu

Dialogflow Essentials

US

botTest

en

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

order_food

SAVE

Try it now

Training phrases

Search training phrase

Add user expression

i want kousskous

i want food

i want to order food

order food

Action and parameters

Please use test console above to try a sentence.

Activier Windows

Accédez aux paramètres de l'ordinateur pour activer Windows.

Dialogflow Essentials

US

botTest

en

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

food_liste

SAVE

Try it now

Define synonyms

Regex entity

Allow automated expansion

Fuzzy matching

rice	rice
cousscouus	cousscouus
steak	steak
fish	fish
tuna	tuna
pizza	pizza
Enter reference value	Enter synonym

Please use test console above to try a sentence.

Activier Windows

Accédez aux paramètres de l'ordinateur pour activer Windows.

botTest

POWERED BY Dialogflow

hello

Hello! How can I help you?

book me a table please

please when you want to book a table

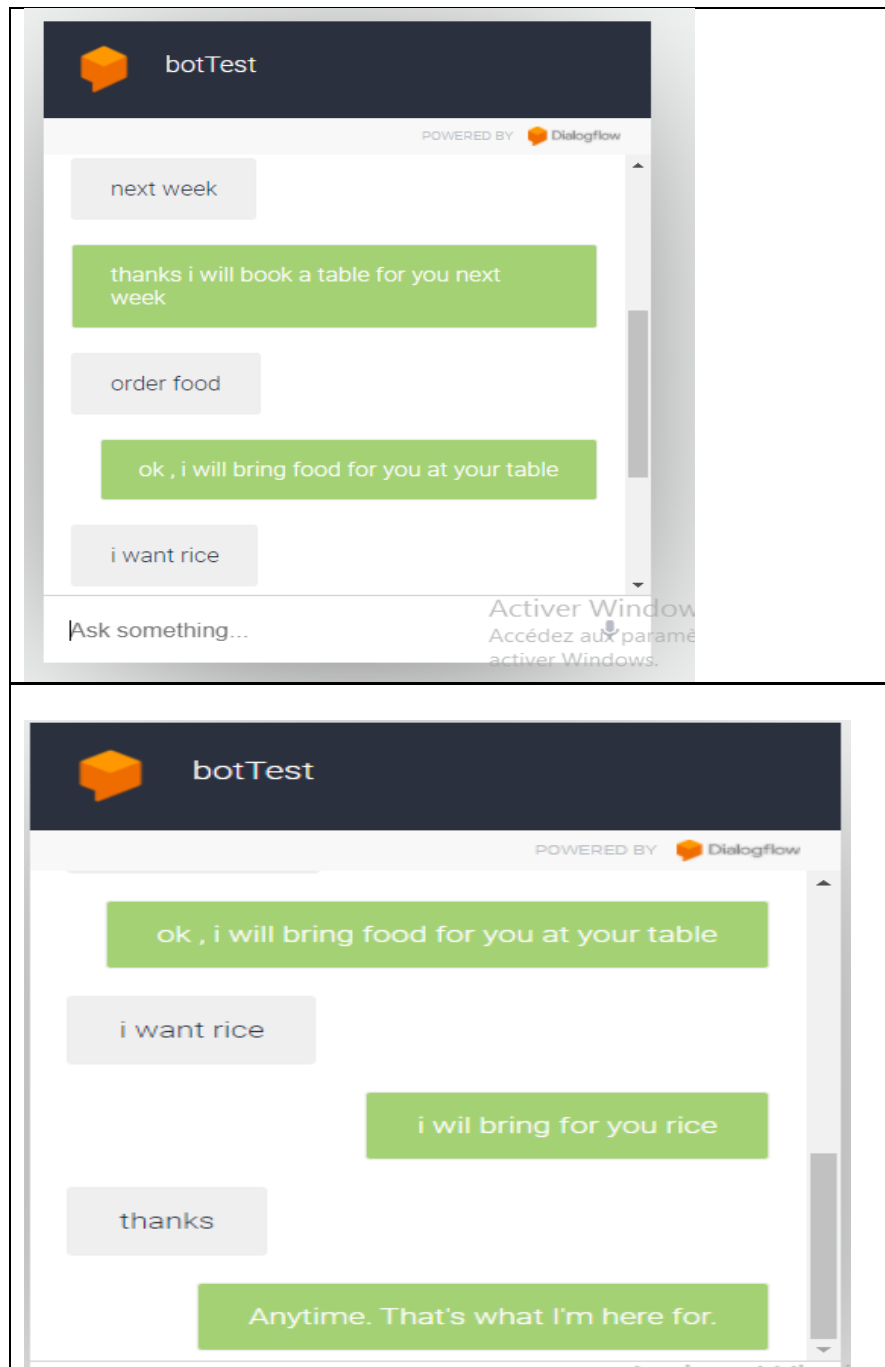
next week

Ask something...

Activier Windows

Accédez aux paramètres de l'ordinateur pour activer Windows.

Compte rendu



Compte rendu

Ex2 chatbot_python :

```
#Question :  
#please book me a table next week  
#please order food  
#i think pizza is good  
#yummy  
#thanks you
```

```
... Hi what's your name?  
>my name is khouloud  
hello khouloud can help you  
>please book me a table next week  
i am going to book a table for you next week  
>please order food  
ok , i will bring food for you at your table  
>i think pizza is good  
i wil bring for you pizza  
>yummy  
Sorry I don't know what `yummy` is?  
>thanks you  
you are welcome  
>
```

Vidéo : Dossier archive

Code :

```
#import the library  
from nltk.chat.util import Chat, reflections  
  
pairs = [  
    [  
        r"My name is (.*)",  
        ['hello %1 can help you'],  
    ],  
    [  
        r"please book me a table (.*)",  
        ['i am going to book a table for you %1'],  
    ],  
    [  
        r"please order food",  
        [' ok , i will bring food for you at your table '],  
    ],  
    [  

```

Compte rendu

```
        r"i think (.*)is good",
        ['i wil bring for you %1'],

    ],

    [
        r"thanks you",
        ['you are welcome'],
    ],

    [
        r'(.*)', # default response if no patterns from above is found
        ["Sorry I don't know what `%1` is?"],
    ],
]

def hugot_bot():
    print("Hi what's your name?")
    chat = Chat(pairs, reflections)
    chat.converse()

if __name__ == "__main__":
    hugot_bot()
```

Texte résumé :

Segmentation de phrases en mots

Normalisation

Suppression de mots vides

Lemmatisation

Tokenisation

stemming

Calcule de fréquence

Code :

Compte rendu

```
Éditeur - C:\Users\users\Desktop\projet mariem\text resume\TResume.py
temp.py TResume.py
5 @author: hp
6
7 import os
8 import nltk
9 import pandas as pd
10 from nltk.tokenize import sent_tokenize, word_tokenize
11 import spacy
12 import fr_core_news_md
13 nlp = fr_core_news_md.load()
14
15
16 testfile=open('text.txt','r',errors='ignore')
17
18 phrase= testfile.read().lower().split('\n')
19
20 def removepunctuation(testfile):
21     testfile1="".join([c for c in text if c not in string.punctuation])
22     return testfile1
23 specialCharacters=['<h1>','</h1>','<h2>','</h2>','_','-','?','!','<p>','</p>']
24
25 def removespecialCharacters(text):
26     testfile2="".join([c for c in test if c in specialCharacters])
27     return testfile2
28
29 text=removepunctuation(testfile)
30 text=removespecialCharacters(testfile)
31 text=text.lower()
32 print(text)
33
34 pattern = re.compile(r'\b('+r'|'.join(stopwords.words('french'))+r')\b\s+')
35 testfile3=pattern.sub('',text)
36 print(testfile3)
37
38 token=word_tokenize(testfile3)
39 print(token)
40
41 from nltk.probability import FreqDist
42 freqDist= FreqDist
```

```
Éditeur - C:\Users\users\Desktop\projet mariem\text resume\TResume.py
temp.py TResume.py
38 token=word_tokenize(testfile3)
39 print(token)
40
41 from nltk.probability import FreqDist
42 freqDist= FreqDist
43
44 plusFreq = freqDist.most_common(5)
45 print(plusFreq)
46
47 def return_token(sentence):
48     doc = nlp(sentence)
49     return [X.text for X in doc]
50
51 return_token(testfile)
52
53
54 from nltk.corpus import stopwords
55 stopWords = set(stopwords.words('french'))
56
57 clean_words = []
58 for token in return_token(test):
59     if token not in stopWords:
60         clean_words.append(token)
61
62 return_stem(testfile)
63
64 #Tokenisation par phrases
65 def return_token_sent(sentence):
66     doc = nlp(sentence)
67     return [X.text for X in doc.sents]
68
69 from nltk.stem.snowball import SnowballStemmer
70 stemmer = SnowballStemmer(language='french')
71
72 #stemming
73 def return_stem(sentence):
74     doc = nlp(sentence)
75     return [stemmer.stem(X.text) for X in doc]
76
77 return_stem(test)
```