Harvard T.H. Chan School of Public Health

Harvard University

User Guide

Narrative Information Linear Extraction

NILE

Document Version:	12.10.2022
Software Version:	12.08.2022

Contents at a Glance

Welcome

NILE is an efficient and effective software for natural language processing (NLP) of clinical narrative texts. It uses a prefix tree algorithm for named entity recognition, and finite-state machines for semantic analysis, both of which were inspired by the natural reading behavior of humans. The design aims to directly translate linguistic and clinical knowledge to code, allowing for the development of functions to parse complex language patterns.

The software was developed by Tianxi Cai and Sheng Yu at Harvard T.H. Chan School of Public Health and Tianrun Cai at The Brigham and Women’s Hospital. It is distributed free of charge for academic and non-commercial research use by the President and Fellows of Harvard College.

1. PREREQUISITES

i. Database

a. Database type: The software had been tested on MSSQL and MySQL databases, but MSSQL database is recommended.

b. Access: Please ensure access to the database before running the tool. If notes are encrypted, please also have the key for decryption ready.

c. Data format: The database table should contain a single medical note for each row. At least five columns are needed for processing. Please see the example below (observe that the “Document_ID” must be unique):

Patient_ID	Document_ID	Note_date	Note_type	Note
10001	1	2018-01-04	Discharge summary	…She does not have diaphoresis associated with it. She does not have any shortness of breath…
10002	2	2010-11-25	Discharge summary	…Ways to reduce CAD risk include eating a healthy diet, regularly exercising, maintaining a healthy weight…
10001	3	2012-07-11	Visit note	…She doesn’t have any shortness of breath….

d. Connection: Please test the connection to the database before running the tool.

e. Encoding: utf-8

ii.Computer

- Hardware
  a. CPU: In order to perform multithreading to have a better processing speed, a multiple core CPU is recommended.
  b. Memory: Recommend >= 8Gb computer memory.

c. Storage: The storage space needed for processing is determined by the size of the dictionary and the number of notes. For example: the size of the results file without compression is 1.4Gb for 7.75 million notes (34K patients, 227 notes/patient) using a dictionary with size 300Kb. By default, the output file is zipped which will reduce to 11%-20% of initial size.

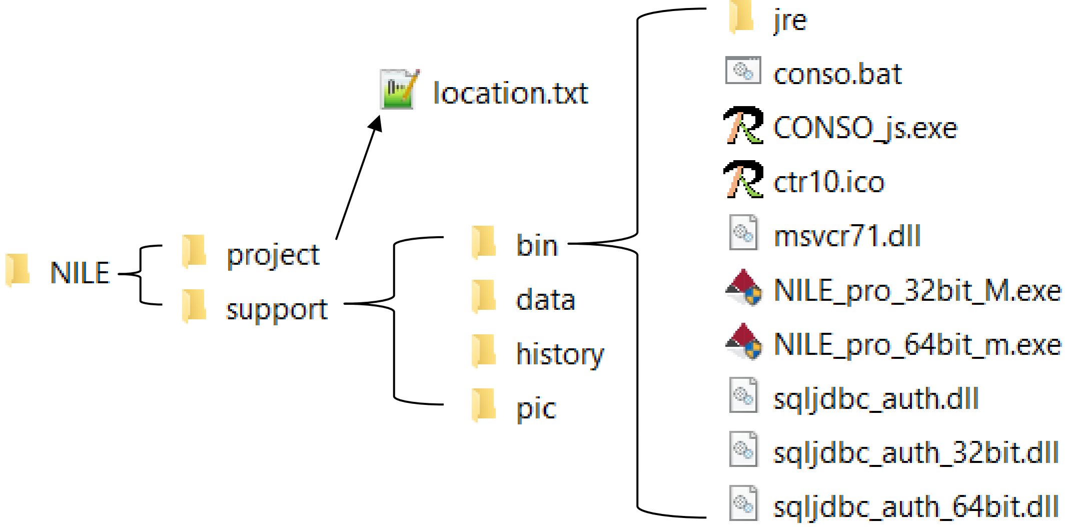
- Software

  a. Operating system: Tested on Windows 7, 8, 10 (32 bit and 64 bit), Mac, Linux

2. Installation

i. Unzip the file to the desired folder and remember the path for the next step.

ii. Structure of the NILE folder:



iii. Choose the correct version of the software.

- There are 2 versions of NILE exe files
  1. Ends with “32bit_m”: for windows 32bit OS, with multiple results files generated.
  2. Ends with “64bit_m”: for windows 64bit OS, with multiple results files generated.
- Choose the correct version, make a copy and rename it to ‘NILE.exe’, then create a shortcut on the desktop for later use.
- Make a copy of sqljdbc_auth_32bit.dll or sqljdbc_auth_64bit.dll based on the OS, then rename it to sqljdbc_auth.dll.

iv. Create a new project

- Create a new folder named by a project name. Clear the content of the file ‘location.txt’ and write the new project name into the file.
- Create a new folder named ‘input’ in the project folder.
- Copy the previously developed dictionary file and rename it to *project_name*+”_dict.txt”. e.g. for the project ‘RA’, the dictionary file would be “RA_dict.txt”
- Copy the file ‘NLPproperties_template.txt’ and paste to the folder ‘input’ created above and rename it to ‘NLPproperties.txt’ for configuration later.

v. System environment variable:

- Windows:
  a. Right click “my computer” or “This PC”
  b. Click “Properties” on the pop-up menu
  c. Click “Advanced System Setting”
  d. Click the button “Environment Variables”.
  e. Click the button “New…” under “User variables for *user name*”, enter “NLP_INTF_HOME” for the “Variable name”
  f. Copy and paste the path to the main NILE folder.
  g. Click “Ok” to save it, and click “Ok” to exit the “Environment Variable Setting”
  h. Sign out and sign in again to have the change take effect.
  i. Run “echo %NLP_INTF_HOME%” on a command prompt. It will return the path to the main NILE folder if the environment variable is saved correctly.
- Mac: *To be added for Jar version*.

vi. Create a function in database.

- At least 5 columns need to be included: patient_number, note_number, note_date, note type and note from the initial table or tables.
- Add a parameter for keys if initial notes are encrypted. The key could be used for calling previously developed SQL function to decrypt notes.
- If notes are not encrypted, just use two parameters.
- For MSSQL database, please see the example below:

```
USE [replace by database name]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- ==============================================
ALTER FUNCTION [scheme].[function name]
(
        -- Add the parameters for the function here
        @PatientOrder1 bigint,
        @patientOrder2 bigint,
        --Add a parameter for key if notes are encrypted
        @key Varchar(MAX)
)
```

```
RETURNS TABLE
AS
RETURN
(
        -- Add the SELECT statement with parameter references here
        SELECT [Patient_number]
                    ,[Note_number]
                ,[note_date]
                -- The function dbo.[ufn_Decrypt] is user defined, change to your
                own function name if it's different
                 ,dbo.[ufn_Decrypt](note, @key) as New_column_name
                ,[note_type]
        From [scheme].[table_name]
        Where  Patient_id >= @PatientOrder1
        And Patient_id <@patientOrder2
)
```

- For MySQL database, please see the example below:

```
DELIMITER $$
DROP PROCEDURE IF EXISTS Note_Processing $$
CREATE  DEFINER=`root`@`localhost`  PROCEDURE Note_Processing(
 IN PatientOrder1 bigint,
 IN PatientOrder2 bigint,
 encryptkey Varchar(255)
)
BEGIN
SELECT Patient_number,Note_number,note_date,dbo.ufn_Decrypt(note, encryptkey)
as New_column_name,note_type
Note_Processing From table_name Where  Patient_id >= PatientOrder1 And
Patient_id <patientOrder2;
END $$
DELIMITER ;
```

3. Configuration

Open the file 'NLPproperties.txt' in the 'input' folder in the desired project and provide a value for each parameter (The comment sign is leading "##". Content follows the comment sign will be ignored by the program):

a. **driver**: the value of driver could be either "mssql" or "mysql". (case sensitive)
b. **authtype**: if the drive is "mssql", please write "windows" or "sql". (case sensitive)
c. **dbname**: not case sensitive
d. **user** and **password**: Please provide user name and password if authtype is "sql"
e. **tablename**: provide scheme + tablename as the tablename when using "mssql"
f. **patientIDcol**: please enter the name for the column containing patient IDs.
g. **docIDcol**: please enter the name for the column containing document IDs.

h. **datecolumn**: please enter the name for the column containing the note dates.

i. **notetype**: please enter the name for the column containing note types.

j. **notecolumn:** please enter the name for the column containing notes.

k. **key:** please enter the key for decryption if notes are encrypted.

l. **patstart** and **patend**: please enter the start patient number and the end patient number if the value of "patientselection" is "RANGE"

m. **patincrement**: the number of patients in each thread

n. **threadlimit**: the number of concurrent threads.

o. **outputfolder**: please write the full path of output folder

p. **zip**: The value could be 'YES' or 'NO'. If the value is 'YES', the output result files and log file will be zipped in order to save space.

4. Results

- Using the example dictionary below: (CUI = Concept Unique Identifier)

TERM	CUI
diaphoresis	C0038990
hidropoiesis	C0038990
cad	C0010054
atherosclerotic heart disease	C0010054
percussion	C0030987
listening	C0004339
shortness of breath	C0013404
healthy diet	C0452415
……	……

- Using the example table in the **Section 1.i.c**
  Example output file below:

```
PatientID|DocumentID|Date|CUIs
10001|1|20180104|C0038990N
10002|2|20101125| C0010054Y, C0452415Y
10001|3|20120711| C0452415N
```

  - 4 columns separated by "|" in the results file which is indicated in the first row
  - The format of date is YYYYMMdd
  - The format of CUIs is a CUI ("C" + 7 digits) followed by a letter "Y", "N" or "U".
    "Y" = Yes (positive mention of a CUI)
    "N" = No (negative mention of a CUI)
    "U" = Unclear (the certainty information is unclear)
  - Multiple CUI mentions are separated by ","

5. Troubleshooting

- If you are getting an error message from running NILE, a few things could be the cause:
    - The NLPproperties file is not set up correctly. This is usually the case. Check the names of the database, the table/function, and the columns in the database.
    - There is no environmental variable “NLP_INTF_HOME”.
    - There is no “location” file in the NLP interface folder.
    - The SQL table/function doesn’t exist or there is an error when it is queried.
- If you run NILE and no error pops up but nothing seems to be happening:
    - First, check the task manager to see if the background process “NILE” is running.
    - Check your “location” folder to see if it matches the project you want.
    - Check the outputfolder variable in the NLPproperties file.
    - Check the sql table/function to see if it returns any entries (if the table is empty then you would see log and results files for NILE, but the results file would be empty)
    - Check the environmental variable “NLP_INTF_HOME” to see if it is the correct folder.

*Note: even if none of the dictionary terms are present in the notes, the results file will still be populated, so this issue does not indicate a problem of having no terms in the note.

Narrative Information Linear Extraction
Version – 2022_1

NILE

NILE

What's new in the new version

1. Threading

   Example table with patient number 1-50000.

   - Patstart =1
   - Patincrement = 5000
     - Result size ≈ 1-1.5 Gb
   - Patend = 50001
   - threadlimit = 2
     - Note: a connection to the database in each thread
2. note length
   - notelengthlimit = 500 (Default)

NILE

What’s new in the new version

3. Cancelled:
   - Patientlist (create a view for a list of patients)
   - noteAmountEach
4. Log file
   - Main log file
     - Error information before threading: dictionary or other files missing, parameters in the property file
     - Total processing time
     - Dictionary problem (don’t worry about this)
   - Thread log file
     - Processing time in each thread

NILE

What’s new in the new version

4. Log file
   - Thread log file (continue)
     - Database connection issue
     - Brief introduction of the data structure in each result file
     - Processing summary
       1. Total notes
       2. Length limit for a note
       3. The number of notes dropped
       4. Null notes
       5. Notes processed
       6. Row number in the result file

       (#1 = #3 + #4 + #5,   #5=#6)