User Guide for the eMERGE-2 CAAD Phenotype NLP System

Version 0.5

February 11, 2015 (updated)

David Carrell and David Cronkite

Contents

[Contact Us 1](#_Toc411424700)

[Introduction 2](#_Toc411424701)

[Choosing the Right NLP System for Your Reports 2](#_Toc411424702)

[Using the NLP System 2](#_Toc411424703)

[Software Download 2](#_Toc411424704)

[Computing Environment Requirements 3](#_Toc411424705)

[Launching the NLP System 4](#_Toc411424706)

[Processing the Test Data 4](#_Toc411424707)

[Exiting the NLP System 8](#_Toc411424708)

[Processing Documents from the File System 8](#_Toc411424709)

[Processing Documents from a Database System 8](#_Toc411424710)

[Error Analysis 12](#_Toc411424711)

[APPENDIX A: Exam Type Regular Expressions 15](#_Toc411424712)

# Contact Us

To assure a prompt response, please contact all of the following people if you have questions about using this NLP system.

David S. Carrell (carrell.d@ghc.org, 206-287-2705)

David Cronkite (cronkite.d@ghc.org, 206-287-2153)

Aaron Scrol (scrol.a@ghc.org, 206-287-2150)

# Introduction

Carotid artery atherosclerosis disease (CAAD) is measured quantitatively by Doppler ultrasound and related imaging studies and reported as a percent stenosis. An easy to use, self-installing natural language processing (NLP) software package is provided to all eMERGE-2 sites for extracting these stenosis measures from unstructured clinical text reports. This document explains how to obtain, set up, and use the CAAD NLP system.

# Choosing the Right NLP System for Your Reports

There are two versions of the NLP system available in the attacked files section of the CAAD PheKB page. Choose the system (from PheKB’s “Files” section) that is most suitable for your reports based on the following descriptions.

* **Version 0.5 of CAAD Portable NLP System (assumes correct exam type)**
This version of the NLP system assumes that each report to be processed contains relevant information from an imaging study of carotid artery stenosis. This is the system you should use if you are able to determine in advance and with reasonably high certainty which reports contain relevant information. As long as the reports contain relevant information it does not matter whether they are based on carotid ultrasounds, CT scans of the head and neck, or other relevant imaging studies.
* **Version 0.5 of CAAD Portable NLP System (checks exam type)**
This version of the NLP system assumes that incoming reports may or may not be relevant imaging studies and therefore attempts to determine, as the first step in processing each report, whether or not the report is relevant. Relevant reports are determined by whether or not they report contains any of several potential exam type descriptions (e.g, CAROTID ULTRASOUND). If the report contains such a description the NLP system will continue to process the report using the same algorithms used in the version of the system that assumes correct exam type). Your reports may have exam type descriptors that are not recognized by the current version of the NLP system. See Appendix A for additional information about exam type regular expressions.

Other than the test for exam type, both NLP systems process carotid stenosis information found in reports in the exact same way.

# Using the NLP System

## Software Download

The CAAD NLP software is available for download from PheKB’s ([www.phekb.org](http://www.phekb.org)) CAAD phenotype page (<http://www.phekb.org/phenotype/caad-carotid-artery-atherosclerosis-disease>). Look for the most recent version of a zip archive file that includes “CAAD portable NLP” in its name or label.

Currently, there is an unlabeled “vanilla” version as well as a “Marshfield” version, which assumes that only appropriate laboratory tests are provided as input. If you are unsure of which version to use, select the “vanilla” version.

Download the zip archive (with a name like PortableNLP\_v*n.n*.zip) to the local hard drive of the machine from which you want to execute the NLP system. Open the archive and drill into its top-level folder to view its contents:



## Computing Environment Requirements

The following are required to run the CAAD NLP system:

1. Tested on Windows, Linux, and Mac operating system
2. Java version 1.7 or higher. The Windows 7 Control Panel display for a qualifying version of Java looks like this:

3. Access, from the local machine, to either a file system or a data base from which the NLP system will read relevant text documents to be processed.
	1. If input files are provided by a database, the database must be MySQL, PostgreSQL, or Microsoft SQL Server, and you must know how to compose an ODBC connection string to link to your database server. You must also know the name of the database table in which the text is stored.
	2. If connectivity to other database systems is desired, please contact David Carrell (carrell.d@ghc.org).
4. Ability to write output files, either to the local file system or to the database (MySQL, PostgreSQL, or Microsoft SQL Server—this does not have to be the same database or type of database from which text is read into the software system).

## Launching the NLP System

Start the CAAD NLP system as follows:

1. Open (double-click) the archive file “PortableNLP.zip” then click and drag (or otherwise copy) fold “v0.4” to your local computer. Copying this folder to the desktop or other location on the local machine’s hard drive will avoid potential issues that may arise if you try to run this application on a network drive.
2. Open folder “v0.4” and double-click the JAR file named “PortableNLP.jar”. This launches the application. Be patient while the application loads—it may take a minute and may appear as though nothing is happening. (On Windows, you can look in the Windows Task Manager’s Applications tab to confirm that the application named eMERGE Application v0.5” is opening.) Once the application opens it will display a window similar to the one shown below.
	1. There are menu headings for File, Settings, Run, and Help.
	2. Below the menu are buttons used to Process Text, Clear Results, Clear Documents, and Debug (more on these later).
	3. The tall narrow window on the left displays system messages. After successfully launching the system it will display the message “Application Loaded.”
	4. The large window on the right is for viewing information about the documents read into the system (see the Documents tab), and viewing data generated by the NLP system (see the Result Data, Result Matches, and Matches tab). For now, all of these tabs are empty.



## Processing the Test Data

The application ships with a small set of de-identified/fictitious text documents which can be processed to test the NLP system. The test data consist of a set of individual text documents in a folder named “data” that is part of the archive. You can now process the test data as follows.

1. Inside the folder named “v0.5” find the folder named “data.” It contains 14 plain files with names like CAAD\_Test\_Report\_1001.txt, as shown below. You may open these reports in a text editor if you want to see what they look like.

2. Launch the NLP system BY double-clicking the JAR file (as in Step 5 above).
3. When the application opens, confirm that “Application Loaded” is displayed in the left pane. There may be other messages displayed regarding driver setup, but you can ignore these for purposes of processing the test data.
4. From the application’s **File** menu select **Load Data** 🡪 **From File …**. This will display an Open file dialog. Use the Open file dialog to navigate to the data folder containing the 14 test notes (described above). Once you have drilled into the “data” folder the open dialog should look similar to this:


5. You may open one file by clicking on it, then clicking the Open button. You may open several files by using the Shift or Ctrl keys in conjunction with mouse clicks (e.g., clicking the first file in the list, then holding down the Shift key while clicking the last file in the list selects all 14 documents). After selecting the files you want to process, click Open.
6. The application’s Documents tab will now display the list of files just opened. Notice that the log window reports the number of documents loaded (14 in this example), and the total number of documents loaded since launching the application (also 14 in this example).


7. Process the loaded documents with the CAAD NLP system by clicking the **Process Text** button. The documents will be processed, usually quickly. Notice the message in the log window that reports processing time in milliseconds. It generally takes only a few dozen milliseconds to process the 14 test documents.
8. After documents have been processed you may inspect results on the three results tabs—Result Data, Result Matches, and Matches. Tab Matches provides the most granular data. The Result Matches tab provides data slightly less granular. The Result Data tab sometimes (but not always) provides a higher level of summary data than the Result matches tab.
9. To save the NLP-generated results, use the menus to select **File** 🡪 **Export Data** … 🡪 **Save to File**. This opens a dialog as shown below.
	1. Give the file a name in the **File Name** field.
	2. Specify the **Tab to Save** (you may choose between Documents, Result Data, Result Matches, and All Matches). The Result Data tab contains the data the NLP system was intended to extract.
	3. Specify the delimiter of your choice and a file name, then click Save, as shown below.

	
10. The save data can be opened in other applications, such as Excel, shown below:


11. Exit the program by selecting from the **File** menu the **Exit** command.

## Exiting the NLP System

To exit the system:

1. Select from the menu **File 🡪 Exit**.

## Processing Documents from the File System

The procedure for processing documents from the file system, where each document is contained in an individual text file uniquely identified by its filename, follow the instructions provided above for Processing the Test Data.

## Processing Documents from a Database System

As noted above, the portable NLP system can read documents from and write results to popular database systems, including MySQL, PostgresSQL, Microsoft SQL Server, and Oracle. This section describes how to configure the NLP system for processing documents from and storing results to a database system.

We illustrate this using Microsoft SQL Server 2008 and a table containing 1,000 imaging reports.

In addition to the requirements described above for running the portable NLP system, your computing environment must also meet the following requirements:

1. It must be possible to connect to the database server from the local machine on which you are running the NLP system.
2. The person under whose credentials the NLP system is running must have at least read permissions to the relevant tables in the database. Additionally, if you want the NLP system to write its results to a database table, the person under whose credentials the NLP system is running must also have database create table permissions and write permissions.
3. You must know how to compose an ODBC connection string for connecting to your database server. A connection string used at Group Health to connect to a Microsoft SQL Server 2008 database using integrated security (i.e., the user’s Windows credentials are used to access the SQL database) looks like this:

jdbc:sqlserver://GHRINLP;databaseName=eMERGE;integratedSecurity=true;

 where
 “GHRINLP” is the name of the server on which the database resides, and
 “eMERGE” is the name of the database.

This information is provided to the NLP system by entering it into the Preferences dialog accessed by selecting from the NLP systems interface **Settings** 🡪 **Preferences**. When entered this information should look like this:



Clicking the “Remember preferences?” option will store this information and make it available the next time you use it. Storing these preferences will also make the database configuration available if you want to write the NLP system’s results to a database table, later on.

Clicking the OK button will instruct the NLP system to attempt a connection to the database using the configuration information supplied.

*Note: If there are problems establishing a connection to the database the NLP system interface may appear to hang for 30 seconds or so while the connection attempt fails.*

1. You must know the name of the database table from which you will read documents, and you must know the names of the columns in that table from which you will read the following information:
	1. Column name for a primary identifier, typically a unique document identifier.
	2. Column name for the document text (i.e., the textual content of the report to be processed).

	*Optionally, if you want them to be present in the NLP system’s output tables, you must know :*
	3. *Optionally, if you want it to be present in the NLP system’s output tables you must know the* column name for a secondary identifier (e.g., a unique patient ID, which can be very useful for post-processing the results of the NLP system).
	4. *Optionally, if you want it to be present in the NLP system’s output tables you must know the* column name for a date associated with each document.

*Note: You may read from a view instead of a table. Just use the view name in place of the table name.*

This information is provided to the NLP system by entering it into the Load Data From Table dialog accessed by selecting from the NLP systems interface menu: **File** 🡪 **Load Data** 🡪 **From Table…**. When entered into the dialog the information should look like this:



Clicking the Load button in the above dialog instructs the NLP system to load the documents from the database. Once the documents are loaded the NLP system will report in its Documents tab summary data for each document, as shown below. Note that the log window (left pane) reports the number of documents loaded, and the Documents tab (in the right pane) provides a scrollable list of information about each document obtained from the database.



1. At this point your documents are ready to be processed by the NLP system using the same procedure described in step 13 on page 5. Once the documents have been processed and the results are available you may save the results to a database table.
2. Saving results to a database table is accomplished as follows:
	1. From the NLP system’s interface menu select **File**🡪 **Export Data** 🡪 **Create Table…**.
	2. Enter a name for the table of results to be created (e.g., CAAD\_NLP\_Results) in the Destination Table field, and select the Results Data option as shown below. The table will be created in the database and under the user credentials you entered in the Preferences dialog (as shown in step 3 above). Clicking Export will write the results to the database table.

	

	*Note: If you export results to a table that already exists they will be appended to the existing data in the table. This may be desirable if you are processing reports in batches. If you do not want to append results you must either delete the existing records from the database table or by deleting the existing table. Deleting records and/or tables must be done outside this NLP system’s interface.*

# Error Analysis

While the NLP system is designed to be as robust as possible, variation in free text reports at different sites will likely cause the system some problems. Resolving these problems requires error analysis. The NLP system has some built-in features intended to facilitate local error analysis by the local user. Findings from your error analysis may be shared with the NLP system developers (taking care to avoid sharing any patient identifying information in the process), who will then use this information to make needed changes to the system.

If the NLP system does not produce correct results on your reports, please follow the guide below to help us figure out what the particular cause may be.

1. Check the “TextLength” column in the “Documents” tab to make sure the values are not getting truncated. You can also check the actual text that’s being loading by going to **Settings 🡪 Columns…** and selecting the check mark next to “Preview of Text” to look at the text that’s being loaded.



1. Clear any existing results by clicking the “Clear Results” button, and then select the “Debug Regex” button. The “Matches” tab should now be populated.



1. Next, go to **Settings 🡪 Columns…**, navigate to the “All Matches” tab and make sure only “Regex Category”, “Regex Subcategory”, and “Source Primary Name” (only if the SourceName column does not contain unique identifiers!—we just want to know which regular expressions have matched in which reports) are checked.



1. Finally, output the “Matches” tab to a text file (**File 🡪 Export Data 🡪 Save to File…**) and send it to David Carrell (carrell.d@ghc.org) for review. Make sure the “All Matches” radio button is selected.



1. If possible, also send some *de-identified* examples of **text snippets** an actual carotid ultrasound report. Snippets of interest would be those which describe the **type of exam** (examples from our reports are “*CAROTID DUPLEX ULTRASOUND*” and “*EXAMINATION: Bilateral carotid ultrasound*”) and **the findings** (examples from our reports are “*narrowing of 60% to 70%*”, “*50-75% stenosis of the right internal carotid artery*” and “*Negative for any significant stenosis in the left carotid artery*”).

# APPENDIX A: Exam Type Regular Expressions

The following eight regular expressions are used to determine which reports are eligible for further NLP system processing by the NLP system that check for exam type.

Exam Type Regular Expressions:

(?:exam(?:ination)?:\s\*((?:left\s+|right\s+|bilateral\s+)?carotid\s+(?:duplex\s+)?(?:doppler\s+)?((?:u/s|us|ultrasound))?))

(carotid\s+(?:(?:doppler:|ultrasound:|us:)|((?:duplex\s+|doppler\s+)ultrasound)))

((?:dup\w\*\s+)?(?:dop\w\*\s+)(?:carotid\s+)(?:[A-Za-z0-9\_>]+\s+){0,1}ultrasound:)

(?:exam(?:ination)?:\s\*((?:head|neck)\s+(?:u/s|us|ultrasound)))

(?:exam(?:ination)?:\s\*((?:head|neck)\s+(?:CT|computed\s+tomography)))

(extracranial\s+carotid\s+(?:duplex\s+)?(?:doppler\s+)?(?:u/s|us|ultrasound))

(?:technique:\s\*((?:(?:\W+\w+){0,5}\W\*\s+)(?:doppler\s+)?(?:ultrasound|examination)(?:(?:\W+\w+){0,5}\W\*\s+)carotid(?:(?:\W+\w+){0,2}\W\*\s+)(?:artery|arteries)))

((?:catheter|CT|MR)(?:\s+angiography))

You may use a regular expression development utility (e.g., [regexpal.com](http://regexpal.com/)) to evaluate whether exam type descriptors in your reports are captured by any of these regular expressions. If any of your exam types are not captured by one or more of these expressions please send an email to the addresses listed in Contact Us above.