# NLP Searches



## NLP searches

There are five types of searches in the NLP Suite: simple word/string search (either single words or colocations, i.e., combinations of two or more words commonly used together, e.g., 'coming out', 'post office'), POSTAG and DEPREL search with Part Of Speech Tag and/or Dependency Relations, N-Grams search, Word Co-Occurrences search, and KWIC search.

## Single words & collocations searches

There are three different types of word/string searches in the NLP form, with different types of input and output, and different units of analysis (sentence or entire document).

1. **Search in multiple doc, docx, rtf or txt files in a folder**
   **Input:** Any folder containing at least one doc, docx, rtf, txt file (files in non-supported formats are automatically ignored) and one or more search terms (wildcards are not supported)
   **Output:** A table with the following fields: "Frequency of word: <searched word>", "File Path", "File Type".

   If more than one search term is specified, then the field "Frequency of word: <searched word>" will appear once for each search term.

   In this search type there is also an option that allows to visualize the context of each occurrence of the searched terms. By context we mean a prefixed number of words before and after each occurrence of the search term. In this case the columns of the output table will be: "Word found", "File Path", "File Type".

2. **Search in single doc, docx, rtf, txt file**
   **Input:** Any doc, docx, rtf or txt file and one or more search terms (wildcards are not supported)

   **Output:** A table with the following fields: "Frequency of word: <searched word>", "File Path", "File Type".

3. **Search in a single txt file containing multiple documents separated by their paths enclosed in < >**
   **Input:** Any doc, docx, rtf or txt file with several documents separated by their paths and other information enclosed in < > and one or more search terms (wildcards are not supported)
   **Output:** A table with the following fields: “Frequency of word: <searched word>”, “File Path”, “File Type”.
   The file path field will contain the file path embedded inside the < > that precede each of the merged document.
   See point 1 for a reference about the content of the other fields of the table.

**Note:** In each of those search routines the input search term will be searched in the documents as a string, therefore also words that contain your search term may appear as result in the output table. For example, the search for the term “sheriff” could also consider the words: “sheriffs” or “Sheriff” because the procedure is case insensitive. In order to avoid this issue, the search term could be preceded and followed by a space.

## POSTAG & DEPREL searches

**Input:** A CoNLL table, the result of running the Stanford CoreNLP on a txt document of choice and one or more search terms (wildcards are supported) and/or DEPREL tags.
**Output:** A table with the following fields: “Form_<number>”, “Deprel_<number>”, “Sentence”, “Document Path”.

This search will be performed on one of the imported CoNLL tables in PC-ACE database, it will return a list of the sentences where the specified word(s) (co-)occur.
The user can specify a filter on each of the search terms based on the POSTAG and DEPREL which s/he wants to be associated with the search term. For example, if the user wants to search for the term “sheriff” and only display the sentences where it occurred with the DEPREL tag “nsubj” this could be done by entering “sheriff” as the only search term and by specifying the desired DEPREL tag. More than a single search term (with its DEPREL tag) can be specified for each search. In this case the columns “Form_<number>”, “Deprel_<number>” will appear as many times as the search term number.

## N-grams searchs

The N-Grams search is a Java routine that allows searches for Ngrams, i.e., key words (e.g., “nursery school” (a 2-gram or bigram), “kindergarten” (a 1-gram or unigram), and “child care” (another bigram) that occur in different documents within a selected time period (e.g., month, year). It works similarly to Google Ngram Viewer except this routine works on documents supplied by the user rather than on the millions of Google books (see https://books.google.com/ngrams/info).

The routine relies on the Stanford CoreNLP for lemmatizing words.

**The routine will display the FREQUENCY OF NGRAMS (WORDS), NOT the frequency of documents where searched word(s) appear, as with Word Co-Occurrences.**

Normalization

Hovering over each data point in the Excel line chart will display the following information: the Group size (i.e., the number of all available documents at that specific data point, regardless of whether any of the documents contain any of the searched words) and the total number of documents in the corpus.

THE INHERENT BIAS OF SUCH A SEARCH IS THAT A SPECIFIC EVENT (E.G., AN "ASSAULT") THAT IS MENTIONED REPEATEDLY IN A SINGLE DOCUMENT WILL LEAD TO A HIGH FREQUENCY, ALTHOUGH THE ACTUAL NUMBER OF DISTINCT "ASSAULT" EVENTS MAY BE LOW.

The routine uses a SET OF TEXT FILES in input and produce an Excel line chart in output.

NGRAMS DO NOT MAKE MUCH SENSE WITH A SINGLE FILE; A POINT WOULD BE PLOTTED!

## Word Co-occurrences searches

The Java routine will allow searches for word co-occurrence, i.e., key words that occur together in the same document.

The routine uses a SET OF TEXT FILES in input and produce an Excel line chart in output.

**The routine will display the FREQUENCY OF DOCUMENTS where searched word(s) appear together in the same document, NOT the frequency of the searched word(s) as with NGrams.**

Hovering over each data point in the Excel line chart will display the following information: the Group size (i.e., the number of all available documents at that specific data point, regardless of whether any of the documents contain any of the searched words) and the total number of documents in the corpus.

THE INHERENT BIAS OF SUCH A SEARCH IS THAT A SPECIFIC EVENT (E.G., AN "ASSAULT") THAT APPEARED IN MANY DCUMENTS WILL LEAD TO A HIGH FREQUENCY, ALTHOUGH THE ACTUAL NUMBER OF DISTINCT "ASSAULT" EVENTS MAY BE LOW.

The routine uses a SET OF TEXT FILES in input and produce an Excel line chart in output.

WORD CO-OCCURRENCE DO NOT MAKE MUCH SENSE WITH A SINGLE FILE; A POINT WOULD BE PLOTTED!

## KWIC (Key Word in Context) searches

**Input:** Any KWIC table, a search term and a position index (wildcards are supported)
**Output:** A table with the following fields “Searched Word”, “Word in Context”, {“-<position>”, “+position” or “Sentence”}

This search allows the user to see what are the words that co-occur in the same document, with the search term at the specified index. For example, consider a document with the following sentence: “A few weeks ago a house and a warehouse were destroyed by fire in Hinesville, and all the circumstances pointed to its being the work of an incendiary.”
If the user typed the search term: “fire” and the position index 2 s/he would see that the words “destroyed” and “Hinesville” occur respectively at position -3 and +3 with respect to the word “fire” in the document.

The user can specify any position index from 1 to 10 (if the selected KWIC table was computed including that position index, the default distance to compute word co-occurrences is 5) or the special context “Sentence”. All the words co-occurring with the search term in the same sentence will be displayed in the output table if the user selects the “Sentence” context.