Functioning of the search tool

In the url the search tool is identified by “search”, ex : search the word “greet”, url = <http://greet-test.es.anl.gov/publications/index.php?content=search?research=greet>

The form to research an expression is in “publi.php”. The user can fill the box and click the submit button or press the enter key.

The click on the submit button of the form or the change of the expression in the form by JavaScript calls *findpubs(searched expression)*. This JavaScript function is located in “publication.js”. *findpubs* is a javascript function that displays the gif image of the loader while the research is performed. It completes the url with the searched word and then call “action.php”. If the connection is busy, it tries again after 5 seconds.

“action.php” is the center piece of the research feature. It uses the xml file containing all the articles data and the xml we previously created to store the keys, keywords and relevance of each article. The research algorithm is the following :

-First a metric is created, it is an instance of the *DistanceMetric* class.

-Foreach word in the research, the distance between the word and the keywords of each article is computed using the Damerau Levenstein distance.

-If the distance is 0, the algorithm put the highest possible score, which is 3 here, to the word. If the distance is less than 2, the score is 1/distance. If the distance is more than 2, the score is 0.

-The algorithm sums up the score for each article. The maximum score is: number of words in the research\*highest score possible per word (it is set to 3). The sum is the relevance of the article.

-The articles are sorted by relevance. Each key related to an article with a perfect score is added to the result list. If there is no perfect score, the 3 keys of the best rated articles are added to the result list.

-For each article in the database, the algorithm checks if its key is in the result list. If so, the article is displayed.

There is a suggestion tool in case the user misspells a word. If there is no perfect score, the algorithm looks for the closest word to the researched word in the database and displays it in a link. A click on the suggestion performs the research for the suggested words.

To extract the keywords from the article, create an instance of the class DistanceMetric (includes/DistanceMetric.php), then use the method *generate\_keywords\_xml*. The results will be an XML file with the structure *dfile->key, keywords, relevance* with all relevances set to 0. Load this file in “action.php” and don’t forget to save the file when the relevances are computed (line 93)

Functioning of the form to modify/remove/add articles

The form is in the file “articles\_form.php”, it uses “publication.js”, “article\_mapping.xml”, “insert.php” and “articles\_per\_author.php”. When the user chooses a name in the form, “publication.js” is called to use “articles\_per\_author.php”. This will fill a second drop down with the names of every article of the chosen author. “article\_mapping.xml” contains all the articles for each author. When the user wants to add an article to the database, “insert.php” is used.

When an author is chosen in the first drop down, and the article is chosen in the second drop down, the fields “author”, “title”, “abstract”, “location”, “group” are displayed filled with the data of the chosen article. The user can then modify or remove it.