2 main goals: Importance / Relevance

Importance : we are looking for important papers

A paper importance is characterized by its number of citations (cited > not cited), and by its date of publication (recent > old).

Relevance : we are looking for papers related to the input

A paper will be relevant if it talks about the same subject than the input. We can estimate the relevance of a paper through several data fields:

* Terms used in the title
* Authors’ previous work
* Mined terms present in the full term

To compare mined terms, we would have to perform a request per paper to get those terms. We are talking about tens of thousands queries to the API. A better way would be to approximate the relevance based on the title and the authors, and then perform a more precise comparison on a subset of our results.

In order to compare the titles, we would have to be able to recognize the important terms they contain. This would require to find some sort of scientific dictionary’s API we could use to tell relevant scientific words apart from usual words. We do not exclude this idea, but would like to try something else first.

We could estimate the relation of papers based on their titles on a less precise way. What really tell two papers apart from a set of papers is the presence of words in these two papers that are missing from everyone else. Based on that idea, we could attribute the relation between two papers a weight. We propose to increment the weight of a relation based on that law:

If (***W*** in ***P1.title*** && ***W*** in ***P2.title***)*weight*[***P1***][***P2***]+= 1 / (*nb\_papers\_title\_containing*[***W***] / *nb\_papers*)

That way, a word only present in two papers represent a strong link between those papers.

“The word frequencies need to be normalized in terms of their relative frequency of presence in the document and over the entire collection” [CHARU C. – A SURVEY OF TEXT CLUSTERING ALGORITHMS].

TF-IDF : Term Frequency normalized by the Inverse Document Frequency. Normalising using DF reduces the importance of terms that appears more frequently in the collection. Reduces the importance of common terms.

Stop-word: common words such as “a”, “an”, “the”… which are not discrimination from a clustering perspective. We can delete these words using a stop word list (ex: 300 – 400 words). We also can identify these words using the document frequency (DF). To minimize their weight, we use inverse document frequency (IDF).

Very infrequent words are also to remove. They do not add anything to similarities comparison. Such terms can be misspellings and so. Frequent in web (blog, tweets …) collections.

Term strength s(t): for two related documents x and y, s(t) = P(t \in x \or t \in y).

Cosine similarity function: common function used to compute similarity between two documents [Section 3 - CHARU C. – A SURVEY OF TEXT CLUSTERING ALGORITHMS].

In order to limit the size of the output, a more precise request is needed. The user should be able to restrict the search by:

* Publication year
* Number of paper returned
* Author names (?)