

# 1 Description of the Experiment

The experiment was performed for 1000 documents from Hulth2003 dataset (Training), which is a collection of abstracts from different papers each with a set of associated human labelled keywords. For each document, a score was calculated, which is defined as follows: -

$$score = \frac{number_{correct}}{number_{system}} \quad (1)$$

where  $number_{correct}$  is the total number of keyphrases extracted by the system each of which is a substring of one of the human labelled keyphrases and  $number_{system}$  is the total number of keyphrases extracted by the system.

For each document  $d_0$  in the Hulth2003 dataset, top  $k$  nearest documents were chosen ( $k = 8$  in my implementation) from the corpus using a similarity measure, string subsequence kernel in our case. Since it was difficult to process 1000 documents, top  $k$  documents were chosen from 50 documents randomly selected from the corpus (random choice is made for each document, although this may not be a great choice as it changes the space of documents under consideration for every document). Then we assign a similarity score to each of the top- $k$  documents (similarity of each of the  $k$  documents to document  $d_0$  i.e.  $sim_{doc}(d_0, d_i)$  for  $i = \{0, 1, \dots, k\}$ ). Then for each pair of words  $(v_i, v_j)$  in the document set  $\{d_0, d_1, \dots, d_k\}$ , an affinity score is calculated as follows: -

$$aff(v_i, v_j) = \sum_{p \in \{0, 1, \dots, k\}} sim_{doc}(d_0, d_p) \times count_{d_p}(v_i, v_j) \quad (2)$$

$M$  is normalized to  $\tilde{M}$  to make the sum of each row equal to 1. The following formula (similar to the graph ranking algorithm in TextRank) is iterated until convergence to get  $WordScore(v_i)$  for each word  $v_i$  in the document set: -

$$WordScore(v_i) = \mu \sum_{j \neq i} WordScore(v_j) \times \tilde{M}_{j,i} + \frac{1 - \mu}{|V|} \quad (3)$$

where  $\mu$  is the *damping factor* and is set to 0.85. Finally, we get all the noun phrases in document  $d_0$  and for each phrase, its score is calculated as the sum of the scores of the words present in the phrase (i.e. sum of  $WordScore(v_i)$ ). Top  $m$  ( $m = 10$  in our implementation) phrases are chosen as top keyphrase.

The average score (see 1) obtained over 1000 documents was 38.8%.