

Text Technologies - Practical 3 - Plagiarism detection  
Ruaridh Thomson - s0786036

### Libraries & Reference

hashlib - python library for computing checksums (md5 is deprecated)  
re - regular expression for string manipulation  
string - also for string manipulation  
math

[1] <http://d3s.mff.cuni.cz/~holub/sw/shash/> - reference for  
simhash  
[2] <http://bibliographie-trac.ub.rub.de/browser/simhash.py> - further simhash help

### Preprocessing

Initially the top line, explaining who gave the speech, is removed so that only the speech contents remain. Furthermore all punctuation is removed and each speech is stored as a string in a list. A separate list further contains each speech as a list of words (the tokens of the speech) that have been converted to lowercase. Going a step further, another list contains each speech in token form with all stopwords ('the', 'them', 'a', 'at' etc.) removed.

### Comparison

Exact duplicates can be found by comparing the string representation of each speech, however this is slow with order  $O(n^2)$ .

Initially I tried using md5 to compute the fingerprint for each speech. Though with trouble I was unsuccessful. Following from [1] and [2] I have been able to incorporate simhash into the plagiarism detector and find near-duplicates with a similarity greater than a given threshold, 99% in this case. Without stopwords the fingerprint of near-duplicates should be almost, if not completely, identical.

Lowering the threshold gives higher recall with lower precision.

The exact duplicates are also found alongside the near-duplicates and therefore I only write the near-duplicates to file (duplicates.txt).

### Duplicates

Taking a look at a pair of very near duplicates, 270897-9347200, we can see simple wording changes, e.g.:

As I have said previously, it is important to give a certain flexibility to the Member States.

As I have said previously, it is important to give a certain flexibility of this proposal to the Member States.