A Hybrid Algorithm for Identifying and Categorizing Plagiarised Text Documents

Victor U. Thompson and Christo Panchev

Abstract: **Advancement in internet technology has made information resources more readily available and much easier for plagiarism to be carried out. Detecting plagiarism is by no means a trivial task because of the sophisticated tactics by which plagiarist disguise their sources. In this paper we present a hybrid algorithm for identifying and categorizing plagiarised text documents. We built our algorithm by combining the potentials of three standard textual similarity measures used in information retrieval (IR). We used the back propagation neural network (BPNN) for combining the measures and the PAN@Clef 2012 text alignment corpus for experimental purpose. We experimented with four categories of plagiarism with each category representing a degree of textual similarity. We measured performance in terms of precision, recall and f-measure. Comparative analysis using the same corpus revealed that our hybrid algorithm (HA) outperformed each of the base similarity measures (BSM) in detecting three out of the four categories of plagiarism, and stood at a virtual tie in the fourth category: [highly similar: HA-96.6183%, BSM-96.5517%, lightly reviewed: HA-84.1321%, BSM-80.9636%, heavily reviewed: HA-68.1188%, BSM-67.1255%, highly dissimilar: HA-70.6280%, BSM-69.7%].**

Keywords**: Information Retrieval, Plagiarism Detection, Similarity Measures, Artificial Neural Networks.**

I. Introduction

The rapid advancement in internet technology has brought about different forms of abuse of information resources such as document duplication, mirroring of websites and plagiarism [1]. Plagiarism is a problem that is often talked about in the academic and commercial sectors, and detecting plagiarism has received considerable attention by researchers in IR and natural language processing (NLP). Plagiarism is the act of copying or duplicating someone's information without referencing or acknowledging the information source or author. There are two frequently mentioned solutions to the problem of plagiarism; they include prevention and detection [2]. Preventing plagiarism means restricting access to websites and materials that could be easily used for plagiarism, enforcing strict laws that would make plagiarism a crime rather than just an ethical matter and educating students on proper referencing/citation in order to avoid plagiarism [3].

V. U. Thompson is currently a Phd student in the *Department of Computing, Engineering and Technology,* University of Sunderland, Edinburgh Building, Chester Road, Sunderland SR1 3SD (Victor.Thompson@research.sunderland.ac.uk).

C. Panchev is a senior lecturer in the Department of Computing, Engineering and Technology, University of Sunderland, Edinburgh Building, Chester Road, Sunderland SR1 3SD (christo.panchev@sunderland.ac.uk).

Detecting plagiarism on the other hand involves building automated systems that are capable of detecting plagiarised documents with a reasonable level of accuracy. This paper is in line with detecting plagiarism using automated systems.

Several techniques have been proposed in the literature for detecting plagiarised documents [4], [5], [6], [7]. Most of these techniques are based on measuring instances of overlaps (overlapping features) between text documents using some form of similarity measurement [8]. These approaches could be classified into three broad categories namely fingerprinting [4], [2], [9], [10], Vector space model (VSM) and ranking [5], [6], [12] and n-gram overlap [7], [11]. See section 2 for details about these approaches.

In most of the approaches used for detecting plagiarism, similarity measures are applied at some point to measure the degree of textual similarity between document pairs, and as argued by Zobel and Hoard [5], some similarity measures are not well suited for some similarity measurement problems. However, it is worth noting that similarity measures function differently [13], and have different potentials. It is therefore likely that a combination of two or more similarity measures will result in a better algorithm (measure) than any of the single measures used in the combination. The question then becomes, how do we combine similarity measures into a hybrid algorithm that performs equal to or greater than the single similarity measures used? In this study, we combined the potentials of three standard similarity measures (Cosine similarity, Jaccard index, Pearson correlation coefficient) into a hybrid algorithm that can automatically search, identify and categorise plagiarised documents based on degree of textual similarity. We worked on four categories of plagiarism taking from the PAN@Clef 2012 text alignment corpus (highly similar, lightly reviewed, heavily reviewed and highly dissimilar plagiarism categories). We compared documents in vector space and used ranking method to retrieve similar documents. We used Artificial Neural Network (ANN) technology to combine the similarity measures and to categorize document pairs based on degree of textual similarity. We measured performance in terms of precision, recall and f-measure; we also measured the error rate of the BPNN by computing its confusion matrix (which is a measure of how often the BPNN misclassifies). We concluded by comparing our hybrid algorithm with the base similarity measures.

II. Previous Research

This section discusses approaches that have been successfully used in the literature for identifying plagiarised documents. Popular approaches include vector space model (VSM) [14] and ranking [6], [5], fingerprinting [4], [2], [9], [10], and n-gram overlap [7], [11]. The VSM approach

ISBN:978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

represents queries and documents as vectors in space and then measure how close each document vector is from the query vector with the help of a similarity measure (cosine similarity in many cases). Documents can then be ranked in decreasing order based on their similarity scores, and document pairs with similarity scores above a predefined threshold are considered potential plagiarised documents. The VSM was used by Sanderson [6] to identify duplicate documents, Hoad and Zobel [5] used ranking technique with modified variants of cosine similarity in the identity measure for identifying co-derivatives (versions and plagiarised documents). The VSM is the most widely used document representation model in IR, it has been successfully used in plagiarism, and in duplicate and near duplicate detection. However, the efficiency of the VSM decreases as the size and number of documents increases, due to increase in dimension.

The fingerprinting approach was first introduced by Manber [4] for identifying similar files in large file systems. The idea is to represent documents as digital fingerprints, and use the amount of overlaps in their fingerprints as a measure of their similarity [15]. Overlapping sections of fingerprints indicate areas of copy (or plagiarism). Fingerprinting itself is a coding technique for mapping large sized documents into smaller sizes using a hash function (e.g. MD5). A hash function divides a document into chunks (substrings or sequence of words) and assigns unique integer (hash value) to each chunk, a document's fingerprint is therefore the collection of its hash values. One drawback of this approach is that, in order to ensure efficiency, only a subset of a document's fingerprint should be used. Hence the question of which chunk to use for fingerprint generation remains an issue, as important chunks that could result in overlap between documents could be easily discarded resulting in inaccuracy in similarity measurements.

In the n-gram approach, the idea is to model documents as n-grams (sequence of words) and use the amount of overlaps between two document's n-grams as a measure of their similarity. Lyon et al., [7] developed a small scale plagiarism detection system based on word trigrams, n-gram overlap was used in [11] for identifying plagiarism in short text passages. The n-gram overlap method is quite effective for verbatim (word for word) similarity analysis, easy to implement and quite efficient for comparing documents. However, one drawback of the n-gram overlap method is that equal weights are assigned to all items in a document without considering the fact that some items are better discriminators and should be assigned higher weights.

III. Overview of Plagiarism Detection

There are basically two types of plagiarism analysis in practice; they include external (or extrinsic) and intrinsic plagiarism [11]. External plagiarism analysis involves searching for passages of texts in a document that were been taken from other documents, while Intrinsic plagiarism analysis involves searching for portions of texts that differ in style and consistency (explanatory details) from other parts of a document. Intrinsic analysis is usually undertaken when there are no reference (source) documents. In plagiarism analysis, the document under investigation is often referred to as the suspicious document, while documents from which portions of texts were taken out of are referred to as source or reference documents [16].

Plagiarists often alter text passages before using them. Some alteration techniques used includes; shuffling and replacement of words with their synonyms, complete removal of some words and phrases, and paraphrasing of passages etc. [16], [17]. The alteration process is called obfuscation, and the degree to which an original passage is altered before being used determines the degree of textual similarity between a document pair. Detecting plagiarism usually begins with the selection of candidate sets (potential plagiarised documents), and then proceeds to a more detailed analysis on the selected candidates in order to accurately confirm plagiarism. The detailed analysis stage is computationally expensive; it requires exhaustive similarity search between pairs of documents. Hence candidate selection is often used to reduce the workload in the detailed analysis stage. One commonly used technique for selecting candidate sets is inverted indexing (inverted file) [18], [19]. In inverted indexing, indexed terms in source documents are stored in a table (database), and all relevant documents to a query are retrieved by running a quick search on the database using terms in a suspicious document as query (i.e. such as in google). Document ranking used in IR for relevance feedback is sometimes applied to filter of less relevant candidates.

In this paper, we addressed the problem of external plagiarism detection, and we considered four categories (degrees) of plagiarism. The categories include highly similar (no-obfuscation), lightly reviewed (low-obfuscation), heavily reviewed (high-obfuscation) and highly dissimilar (no-plagiarism). The degree of textual similarity is highest in the highly similar texts and lowest in the highly dissimilar texts.

IV. Similarity measures

Similarity measures are functional tools used for measuring the similarity between objects (where objects means texts in this study). When used for text similarity measurements, similarity measures output similarity scores (integers) that indicate how similarity two texts are. Similarity scores are usually in the range of 0 and 1; where 0 represents absolute dissimilarity and 1 represents absolute similarity, scores between 0 and 1 are intermediate levels of similarity [20]. The similarity measures used in this study includes; Cosine similarity, Jaccard-index [21] and Pearson correlation coefficient [22].

Similarity measures are often used to address documents similarity measurement problems, they have been successfully applied in plagiarism detection [5], document clustering and categorization [23], duplicate and near duplicate detection [24].

V. Techniques for combining Similarity Measures

Similarity measures do have different scale of measurements, combining them is therefore not a trivial task as some similarity measures naturally output higher scores than others. Various techniques could be used to combine similarity measures; however, they do have their individual short comings.

ISBN:978-988-19253-4-3 (revised on 5 May 2015)
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Z-score and min-max normalization techniques are commonly used to address differences in scale of measurements, however using z-score does not guarantee equal range, and the min-max method is often skewed by outliers (too sensitive to outliers) [25]. Other techniques include weighted combination used in AI (weighted voting, averaging, summation and Maximum-value). Weighted combination assigns different weights to different classifiers (similarity measures in this case) based on their performance from previous experiments [26]. One major short coming of weighted combination techniques with respect to this study is that, it is difficult to obtain the right weights to assign to each similarity measure; countless numbers of different weights would have to be tried which is almost impossible.
One other method that assigns weights randomly and makes adjustments recursively until the appropriate weights are assigned to each classifier is the BPNN. BPNN seems like a viable tool that can be used to combine similarity measures, however it takes time to train a network, and BPNN sometimes gets overfitted, although several techniques have been proposed for addressing the overfitting problem. The use of BPNN for combining similarity measures is similar to stacking used in AI for combing classifiers. One major advantage of stacking is that it results in algorithms that can be highly generalization [26]. Other AI ensemble techniques such as bagging, boosting, Bayesian combiner etc [26] could have been considered in this study, but similarity measures are not real classifiers and the BPNN seem a good fit and a viable tool that can be used to address the problem.We used the BPNN for combining the similarity measures.

A. Artificial Neural Networks

Artificial Neural Networks (ANNs) are information processing systems modeled to function like the human nervous system; the brain to be specific. An ANN comprises of a network of interconnected neurons (signal processing units) working in harmony to resolve a common problem. In machine learning (ML) and Artificial Intelligence (AI), ANNs are generally used to resolve pattern recognition and classification problems such as face, voice and handwritten recognition, objects classification and categorization, etc.

The basic architecture of a Multilayer Perceptron ANN (MLP) comprises of an input and an output layer; and one or more hidden layers, where each layer comprises of one or more nodes (neurons). The network receives input data through the input nodes and passed on to the hidden layer neurons. The hidden layer is like a black box where additional computation is carried out in order to extract more statistics [27] from the input data and ultimately improve classification accuracy. The MLP is trained using the Error Back Propagation algorithm (BPNN): as input data are processed into output, the difference between the derived output and the expected out is calculated and sent back through the network in order to adjust the weights at the input and hidden layers for optimum performance. The BPNN calculates the errors at the output nodes and propagate them backwards to their respective hidden nodes for further calculation and adjustments (gradient descent). It is a recursive back and forth process that continues until the errors are reduced to minimum (see [27], [28] for details on BPNN).

ANNs (BPNN in particular) have successfully been used in several studies for categorizing text documents [29], [30], [21]. The implementation of BPNN in those studies is very similar to our implementation, the difference however is rather than using textual contents as input (feature vectors) to the network we used similarity scores generated by similarity measures.

VI. Methodology

The methodology in this study is divided into two parts, the first part involves putting together relevant methods that can be used to compare and identify plagiarised documents using single similarity measures. The second part involves developing an appropriate technique for combining the potentials of similarity measures into an algorithm that is likely to be more effective in identifying and categorizing plagiarised documents. We used the vector space model (VSM) approach for the first part because of its remarkable success in recent years, and because it is currently the most popular document model used in IR [31], [32] for ranking and categorizing documents. To implement the VSM, each document in a corpus must be transformed into a vector. To convert a document into a vector, it must first be preprocessed and indexed (weighing of terms). Document vectors can be compared, and plagiarised pairs identified using similarity measures and document ranking respectively.

A. Data Pre-Processing

Data pre-processing helps in removing noisy data and presents documents in a format that makes them comparable. Typical data preprocessing steps used in IR and NLP include; tokenization, stop-word removal and stemming [33]. Documents are tokenized in order to transform them into bag-of-words or word n-grams (sequence of words) for indebt comparison to be carried out based on overlapping words or n-grams. Stop-words (i.e. the, them, he, she) are words with low discriminating power; and are usually removed. Stemming reduces words to their root-form (“friendly” and, “friendship” can be stemmed to “friend”) which ultimately improves computational efficiency and increases the chances of overlaps (and ultimately recall) [33] during document comparison. To complete the data preprocessing step, appropriate textual features have to be chosen to optimize performance. Textual features such as words, sentences, n-grams etc. are often used in computational linguistics studies [34], and several feature selection techniques such as chi-square statistics, information gain, mutual information [35] etc. have also been proposed. However, word n-grams have shown remarkable success in previous research in documents similarity measurement [12], [17], and was the choice of feature used in this study. One benefit of word n-grams to this study is that, not only are n-grams easy to generate, they can effectively separate one category of plagiarised documents from the other [12], [17].

B. Term Weighting

After preprocessing, documents are converted to vectors by assigning weights to indexed terms. Indexing speeds up document comparison and retrieval [36], while assignment

ISBN:978-988-19253-4-3 (revised on 5 May 2015) WCE 2015
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

of weights to indexed terms ensures that each term is well represented according to its importance in a document. Popular term weighting methods include; term frequency (TF), Term frequency-inverse document frequency (TFIDF) [36] and binary weighing.

C. Document Comparison and Retrieval

Document comparison involves measuring the similarity between a suspicious document and an entire collection of source documents using an appropriate similarity measure. Documents can be ranked in decreasing order of similarity (based on similarity scores) and document pairs with similarity scores above predefined threshold can be retrieved as plagiarised.

D. Combination of Similarity Measures into Hybrid Algorithm

Combination of similarity measures can be done using the similarity scores generated by the similarity measures for each pair of document compared. In order to combine similarity measures, an appropriate combining algorithm that can normalize the scale of measurement differences of the similarity measures is required. In this study, we used the BPNN as the combining algorithm. The BPNN was used because of its ability to offset the differences in scale of measurement by assigning random weights to each similarity measure and making adjustment in the weights in order to achieve a target output. The BPNN is also beneficial in this study because it can be used to categorize plagiarised documents.

VII. Description of Task

Our main task in this study is to develop a method that can be used to combine the potentials of similarity measures and use the combined algorithm to identify and categorize plagiarised documents in a large document collection. The task can be divided into two parts; the first part involves measuring the similarity between documents using the three standard similarity measures (cosine similarity, Jaccard-index and Pearson correlation coefficient). The second part involves combining the similarity measures into a hybrid algorithm using their similarity scores and a suitable combining algorithm. The hybrid algorithm is expected to be more effective in identifying and categorising plagiarised documents than any of the single similarity measures used in this study.

VIII. Experiments

A. *Corpus:* We used the PAN@Clef 2012 texts alignment corpus in our experiments. The corpus is artificially generated and comprises of 6500 documents, of which 3000 are suspicious documents (plagiarized at different degrees) and the remaining 3500 are source documents (the original documents where the plagiarised passages in the suspicious documents were taken from). The corpus comes with its ground-truth; which are pairs of documents with their accurate categories according to human judgment. The ground-truth is relevant for evaluation purpose, and particularly useful for training purpose in this study.

B. Description of Experiments

1) Identifying and categorizing plagiarised documents using single similarity measures

Each document in the corpus was first preprocessed and transformed into a vector ready for comparison. In order to reduce the amount of document to document comparison and to ultimately scale up to a large collection of data, we applied inverted indexing and document ranking to select candidate documents. Our technique for selecting potential candidate documents is similar to probability (coarse) counting used in [9] for finding replicas of documents. We used the indexed terms in each suspicious document to retrieve all relevant source document ID's from the inverted index table and then applied ranking (and an accumulator) to select only document ID's (as candidates) that contains a certain amount of the indexed terms in each suspicious document.

We used the VSM approach for identifying plagiarised documents and word n-grams for separating one category of plagiarism from others; we used 12-grams to identify highly similar, 4-grams for lightly reviewed and 2-grams for heavily reviewed plagiarised documents. This process was carried out with each similarity measure and their performances in each plagiarism category were measured in precision, recall and f-measure.

2) Combining similarity measures into hybrid algorithm using BPNN

We used the BPNN to combine the potentials of similarity measures. We trained and tested the BPNN using similarity scores generated by the similarity measures as features. The training was based on supervised learning; pairs of documents with their accurate categories (label) were presented to the BPNN to learn from. The BPN was built on MATLAB; it was trained with 60% of the data, validated with 20% and tested with remaining 20%. The performance of the network in terms of classification accuracy was measured by calculating and plotting its confusion matrix (a measure of how often it misclassifies certain documents).The weights and bias at the point where the BPNN performed best were exported and used as parameters to combine the three similarity measures. In the final phase of our experiments, we compared the performance of the hybrid algorithm with each of the base (single) similarity measures.

3) Outline of the Hybrid Algorithm (pseudocode)

i. Data pre-processing.
ii. Candidate set selection using inverted indexing and document ranking.
iii. Document comparison using the three similarity measures experimented with.
iv. Combination of similarity measures using weights and bias parameters (from BPNN) and the similarity scores obtained from the previous stage.
v. Automatic identification and categorization of documents pairs into one of the four textual (or plagiarised) categories experimented with.

ISBN:978-988-19253-4-3 (revised on 5 May 2015)
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IX. Results and Discussions

The tables below contain the results in precision, recall and f-measure for the three similarity measures (on the four levels of textual similarity). Tables 1-4 contain the performances of the individual measures while table 5 contains the performance of the hybrid measure (combined measure). Figure 1 is a plot of the confusion matrix of the BPNN.

For the highly similar document category, the best performance (in terms of f-measure) obtained for the base (individual) similarity measures (96.5517%) matches that of the hybrid measure (96.6183). There was no significant improve in performance in this category. For the lightly reviewed, heavily reviewed and highly dissimilar categories, the hybrid measure outperformed each of the base similarity measures (84.1321%, 68.1188%, and 70,628 %). The results in these categories are a clear indication that the hybrid algorithm actually combines the potentials of the base similarity measures.

The results show similar trend shared by the hybrid measure and the base similarity measures; there was a steady decrease in performance from the highly similar document category to the heavily reviewed category, and then the performance ticked-up slightly on the highly dissimilar category. This trend indicates that, as the degree of documents similarity decreases, it becomes more difficult to accurately measure their similarity (due to very few and scattered overlaps compared to the size of the documents). The overall performance of both the base similarity measures and the hybrid measure was highest on the highly similar document category, very close to the one hundred percent mark.

The closeness and the relatively low performances at the highly dissimilar and heavily reviewed categories suggest that the two categories should have been merged together as one, as there is no real difference in similarity between documents in these categories. This can be clearly seen in the confusion matrix plot which reveals a high level of misclassification in the highly dissimilar and heavily reviewed categories as can be seen in fig 1 below.

Table 1 performance on highly similar documents

Similarity Measures	Precision %	Recall %	F-measure %
Cosine similarity	95.1456	98.0	96.5517
Jaccard-index	95.1456	98.0	96.5517
Pearson correlation coefficient	93.33	95.64	94.471

Table 2 Performance on lightly reviewed documents

Similarity Measures	Precision %	Recall %	F-measure %
Cosine similarity	79.4465	82.5397	80.9636
Jaccard-index	78.1603	82.0313	80.1414
Pearson correlation coefficient	71.3870	77.2355	74.1962

Table 3 performance on heavily reviewed documents

Similarity Measures	Precision %	Recall %	F-measure %
Jaccard-index	65.445	69.08	67.213
Cosine similarity	61.889	73.33	67.1255
Pearson correlation coefficient	58.574	71.186	63.935

Table 4 performance on highly dissimilar documents

Similarity Measures	Precision %	Recall %	F-measure %
Cosine similarity	67.8229	71.685	69.7
Jaccard-index	66.6754	70.5778	68.5711
Pearson correlation coefficient	63.375	72.0	67.4127

Table 5 performance of hybrid algorithm

Document's categories	Precision %	Recall %	F-measure %
Highly similar	93.4579	100.0	96.6183
Lightly reviewed	81.5	86.94	84.1321
Heavily reviewed	59.434	79.776	68.1188
Highly dissimilar	61.24	83.476	70.628

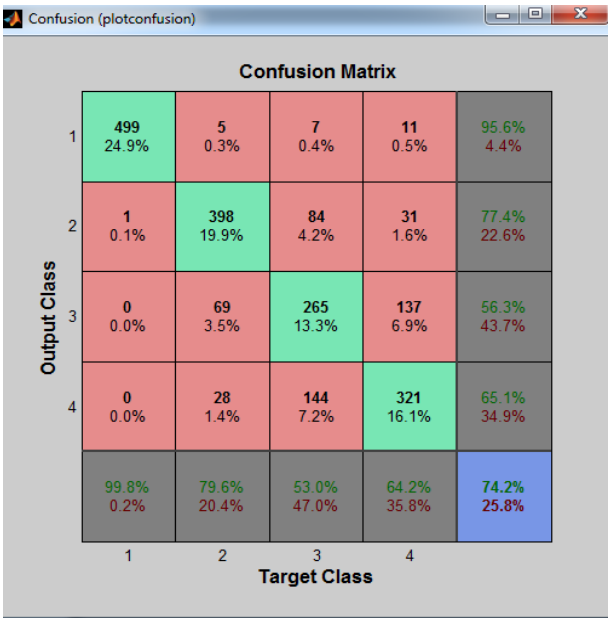

Fig 1 confusion matrix plot for the BPNN

X. Conclusion

In this paper, we present a method for combining the potentials of two or more similarity measures into a hybrid algorithm for detecting plagiarised documents. We implemented our method by combining the potentials of three standard similarity measures into a hybrid algorithm for detecting and categorizing plagiarised text documents. We used the back propagation neural network for combining the similarity measures and for categorizing plagiarised documents into four classes of textual similarity namely; highly similar, lightly-reviewed, heavily reviewed and highly dissimilar document categories. Experimental results show that outperformed the base similarity measures on three out of the four categories. Future work would be focused on generalizing the hybrid algorithm on other corpus. We also intend to swap some of the base similarity measures with other similarity measures such as kullback-Leibler divergence to find out whether or not the performance of the algorithm could be further improved.

Acknowledgements

I would like to give special thanks to Michael Oakes of the University of Wolverhampton and Valentina Plekhanova of the University of Sunderland for their enormous support on a wide range of areas with respect to research and development.

References

[1] Manku, G. S., Jain, A., & Das Sarma, A. (2007, May). Detecting near-duplicates for web crawling. In *Proceedings of the 16th international conference on World Wide Web* (pp. 141-150). ACM

[2] Brin, S., Davis, J., & Garcia-Molina, H. (1995, June). Copy detection mechanisms for digital documents. In *ACM SIGMOD Record* (Vol. 24, No. 2, pp. 398-409). ACM.

[3] Stappenbelt, B., & Rowles, C. (2010, February). The effectiveness of plagiarism detection software as a learning tool in academic writing education. In *4th Asia Pacific Conference on Educational Integrity (4APCEI)* (p. 29).

[4] Manber, U. (1994, January). Finding Similar Files in a Large File System. In*Usenix Winter* (Vol. 94, pp. 1-10).

[5] Hoad, T. C., & Zobel, J. (2003). Methods for identifying versioned and plagiarized documents. *Journal of the American society for information science and technology*, *54*(3), 203-215.

[6] Sanderson, M. (1997). Duplicate detection in the Reuters collection. " *Technical Report (TR-1997-5) of the Department of Computing Science at the University of Glasgow* G12 8QQ, UK"

[7] Lyon, C., Malcolm, J. and Dickerson, B. (2001), Detecting Short Passages of Similar Text in Large Document Collections, In *Proceedings Of 2011 Conference on Empirical Methods on Natural Language Processing*, (Pp.118-125).

[8] Stein, B., & Zu Eissen, S. M. (2006). Near similarity search and plagiarism analysis. *In From Data and Information Analysis to Knowledge Engineering* (pp. 430-437). Springer Berlin Heidelberg.

[9] Shivakumar, N., & Garcia-Molina, H. (1999). Finding near-replicas of documents on the web. In *The World Wide Web and Databases* (pp. 204-212). Springer Berlin Heidelberg.

[10] Heintze, N. (1996, November). Scalable document fingerprinting. In *1996 USENIX workshop on electronic commerce* (Vol. 3, No. 1).

[11] Clough, P., & Stevenson, M. (2011). Developing a corpus of plagiarised short answers. *Language Resources and Evaluation*, *45*(1), 5-24.

[12] Shivakumar, N., & Garcia-Molina, H. (1995). SCAM: A copy detection mechanism for digital documents.

[13] Cha, S. H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, *1*(2), 1.

[14] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.

[15] Broder, A. Z. (1997, June). On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings* (pp. 21-29). IEEE.

[16] Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., & Rosso, P. (2010, September). Overview of the 2nd International Competition on Plagiarism Detection. In *CLEF (Notebook Papers/LABs/Workshops)*.

[17] Clough, P. (2000). Plagiarism in natural and programming languages: an overview of current tools and technologies. *Research Memoranda: CS-00-05, Department of Computer Science, University of Sheffield, UK*, 1-31.

[18] Moffat, A., & Zobel, J. (1996, October). Self-indexing inverted files for fast text retrieval.*ACM Transactions on Information Systems (TOIS)*, *14*(4), 349-379.

[19] Frieder, O., Grossman, D., & Chowdhury, A. (1999). Efficiency considerations in very large information retrieval servers. In *Journal of Digital Information,(British Computer Society)*.

[20] Metzler, D., Bernstein, Y., Croft, W. B., Moffat, A., & Zobel, J. (2005, October). Similarity measures for tracking information flow. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 517-524). ACM.

[21] Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. *New phytologist*, *11*(2), 37-50.

[22] Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, *58*(347-352), 240-242.

[23] Bigi, B. (2003). *Using Kullback-Leibler distance for text categorization* (pp. 305-319). Springer Berlin Heidelberg.

[24] Yang, H., Callan, J., & Shulman, S. (2006, May). Next steps in near-duplicate detection for eRulemaking. In *Proceedings of the 2006 international conference on Digital government research* (pp. 239-248). Digital Government Society of North America.

[25] Tulyakov, S., Jaeger, S., Govindaraju, V., & Doermann, D. (2008). Review of classifier combination methods. In *Machine Learning in Document Analysis and Recognition* (pp. 361-386). Springer Berlin Heidelberg.

[26] Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1-2), 1-39.

[27] Haykin, S. (2009). *Neural networks and learning machines* (Vol. 3). Upper Saddle River: Pearson Education.

[28] Kecman, V. (2001). *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*. MIT press.

[29] Ruiz, M. E., & Srinivasan, P. (1998, October). Automatic text categorization using neural networks. In *Proceedings of the 8th ASIS SIG/CR Workshop on Classification Research* (pp. 59-72).

[30] Zaghloul, W., Lee, S. M., & Trimi, S. (2009). Text classification: neural networks vs support vector machines. *Industrial Management & Data Systems*, *109*(5), 708-717.

[31] Ghiassi, M., Olschimke, M., Moon, B., & Arnaudo, P. (2012). Automated text classification using a dynamic artificial neural network model. *Expert Systems with Applications*, *39*(12), 10967-10976.

[32] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1, p. 6). Cambridge: Cambridge university press.

[33] Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*(1), 141-188.

[34] Grieve, J. (2007). Quantitative authorship attribution: An evaluation of techniques. *Literary and linguistic computing*, *22*(3), 251-270.

[35] Michael P. Oakes, (2014). "*Literary Detective Work on the Computer*", Amsterdam: John BenjaminsBaeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463). New York: ACM press

[36] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing and management, 24(5)*, pp5

Modification date: 05/05/2015
Modification made: correction of typographical and numbering errors, and the inclusion of an acknowledgement section.