METER: MEasuring TExt Reuse

Paul Clough and **Robert Gaizauskas** and **Scott S.L. Piao** and **Yorick Wilks**

Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello Street,
Sheffield, England, S1 4DP
{initial.surname@dcs.shef.ac.uk}

Abstract

In this paper we present results from the METER (MEasuring TExt Reuse) project whose aim is to explore issues pertaining to text reuse and derivation, especially in the context of newspapers using newswire sources. Although the reuse of text by journalists has been studied in linguistics, we are not aware of any investigation using existing computational methods for this particular task. We investigate the classification of newspaper articles according to their degree of dependence upon, or derivation from, a newswire source using a simple 3-level scheme designed by journalists. Three approaches to measuring text similarity are considered: n-gram overlap, Greedy String Tiling, and sentence alignment. Measured against a manually annotated corpus of source and derived news text, we show that a combined classifier with features automatically selected performs best overall for the ternary classification achieving an average F_1-measure score of 0.664 across all three categories.

1 Introduction

A topic of considerable theoretical and practical interest is that of *text reuse*: the reuse of existing written sources in the creation of a new text. Of course, reusing language is as old as the retelling of stories, but current technologies for creating, copying and disseminating electronic text, make it easier than ever before to take some or all of any number of existing text sources and reuse them verbatim or with varying degrees of modification.

One form of unacceptable text reuse, *plagiarism*, has received considerable attention and software for automatic plagiarism detection is now available (see, e.g. (Clough, 2000) for a recent review). But in this paper we present a benign and acceptable form of text reuse that is encountered virtually every day: the reuse of news agency text (called *copy*) in the production of daily newspapers. The question is not just whether agency copy has been reused, but to what extent and subject to what transformations. Using existing approaches from computational text analysis, we investigate their ability to classify newspapers articles into categories indicating their dependency on agency copy.

2 Journalistic reuse of a newswire

The process of gathering, editing and publishing newspaper stories is a complex and specialised task often operating within specific publishing constraints such as: 1) short deadlines; 2) prescriptive writing practice (see, e.g. Evans (1972)); 3) limits of physical size; 4) readability and audience comprehension, e.g. a tabloid's vocabulary limitations; 5) journalistic bias, e.g. political and 6) a newspaper's house style. Often newsworkers, such as the reporter and editor, will rely upon news agency copy as the basis of a news story or to verify facts and assess the

Proceedings of the 40th Annual Meeting of the Association for
Computational Linguistics (ACL), Philadelphia, July 2002, pp. 152-159.

importance of a story in the context of all those appearing on the newswire. Because of the nature of journalistic text reuse, differences will arise between reused news agency copy and the original text. For example consider the following:

Original (news agency) *A drink-driver who ran into the Queen Mother's official Daimler was fined £700 and banned from driving for two years.*

Rewrite (tabloid) *A DRUNK driver who ploughed into the Queen Mother's limo was fined £700 and banned for two years yesterday.*

This simple example illustrates the types of rewrite that can occur even in a very short sentence. The rewrite makes use of slang and exaggeration to capture its readers' attention (e.g. *DRUNK, limo, ploughed*). Deletion (e.g. *from driving*) has also been used and the addition of *yesterday* indicates when the event occurred. Many of the transformations we observed between moving from news agency copy to the newspaper version have also been reported by the summarisation community (see, e.g., McKeown and Jing (1999)).

Given the value of the information news agencies supply, the ease with which text can be reused and commercial pressures, it would be beneficial to be able to identify those news stories appearing in the newspapers that have relied upon agency copy in their production. Potential uses include: 1) monitoring take-up of agency copy; 2) identifying the most reused stories ; 3) determining customer dependency upon agency copy and 4) new methods for charging customers based upon the amount of copy reused. Given the large volume of news agency copy output each day, it would be infeasible to identify and quantify reuse manually; therefore an automatic method is required.

3 A conceptual framework

To begin to get a handle on measuring text reuse, we have developed a *document-level* classification scheme, indicating the level at which a newspaper story as a whole is derived from agency copy, and a *lexical-level* classification scheme, indicating the level at which individual word sequences within a newspaper story are derived from agency copy. This framework rests upon the intuitions of trained journalists to judge text reuse, and not on an explicit lexical/syntactic definition of reuse (which would presuppose what we are setting out to discover).

At the document level, newspaper stories are assigned to one of three possible categories coarsely reflecting the *amount* of text reused from the news agency and the *dependency* of the newspaper story upon news agency copy for the provision of "facts". The categories indicate whether a trained journalist can identify text rewritten from the news agency in a candidate derived newspaper article. They are: 1) **wholly-derived (WD)**: all text in the newspaper article is rewritten *only* from news agency copy; 2) **partially-derived (PD)**: some text is derived from the news agency, but other sources have also been used; and 3) **non-derived (ND)**: news agency has not been used as the source of the article; although words may still co-occur between the newspaper article and news agency copy on the same topic, the journalist is confident the news agency has not been used.

At the lexical or word sequence level, individual words and phrases within a newspaper story are classified as to whether they are used to express the same information as words in news agency copy (i.e. paraphrases) and or used to express information not found in agency copy. Once again, three categories are used, based on the judgement of a trained journalist: 1) **verbatim**: text appearing word-for-word to express the same information; 2) **rewrite**: text paraphrased to create a different surface appearance, but express the same information and 3) **new**: text used to express information not appearing in agency copy (can include verbatim/rewritten text, but being used in a different context).

3.1 The METER corpus

Based on this conceptual framework, we have constructed a small annotated corpus of news texts using the UK Press Association (PA) as the news agency source and nine British daily newspapers[1] who subscribe to the PA as candidate reusers. The METER corpus (Gaizauskas et al., 2001) is a collection of 1716 texts (over 500,000 words) carefully selected from a 12 month period from the areas of law and court reporting (769 stories) and showbusiness (175 stories). 772 of these texts are PA copy and 944 from the nine newspapers. These texts cover 265 different stories from July 1999 to June 2000 and all newspaper stories have been manually classified at the document-level. They include 300 wholly-derived, 438 partially-derived and 206 non-derived (i.e. 77% are thought to have used PA in some way). In addition, 355 have been classified according to the lexical-level scheme.

4 Approaches to measuring text similarity

Many problems in computational text analysis involve the measurement of *similarity*. For example, the retrieval of documents to fulfil a user information need, clustering documents according to some criterion, multi-document summarisation, aligning sentences from one language with those in another, detecting exact and near duplicates of documents, plagiarism detection, routing documents according to their style and identifying authorship attribution. Methods typically vary depending upon the matching method, e.g. exact or partial, the degree to which natural language processing techniques are used and the type of problem, e.g. searching, clustering, aligning etc. We have not had time to investigate all of these techniques, nor is there space here to review them. We have concentrated on just three: ngram overlap measures, Greedy String Tiling, and sentence alignment. The first was investigated because it offers perhaps the simplest approach to the problem. The second was investigated because it has been successfully used in plagiarism detection, a problem which at least superficially is quite close to the text reuse issues we are investigating. Finally, alignment (treating the derived text as a "translation" of the first) seemed an intriguing idea, and contrasts, certainly with the ngram approach, by focusing more on local, as opposed to global measures of similarity.

4.1 Ngram Overlap

An initial, straightforward approach to assessing the reuse between two texts is to measure the number of shared word ngrams. This method underlies many of the approaches used in copy detection including the approach taken by Lyon et al. (2001).

They measure similarity using the set-theoretic measures of containment and resemblance of shared trigrams to separate texts written independently and those with sufficient similarity to indicate some form of copying.

We treat each document as a *set* of overlapping n-word sequences (initially considering only n-word types) and compute a similarity score from this. Given two sets of ngrams, we use the set-theoretic containment score to measure similarity between the documents for ngrams of length 1 to 10 words. For a source text `A` and a possibly derived text `B` represented by sets of ngrams $S_n(A)$ and $S_n(B)$ respectively, the proportion of ngrams in `B` also in `A`, the ngram containment $C_n(A, B)$, is given by:

$$C_n(A,B) = \frac{\mid S_n(A) \cap S_n(B) \mid}{\mid S_n(B) \mid} \tag{1}$$

Informally containment measures the number of matches between the elements of ngram sets $S_n(A)$ and $S_n(B)$, scaled by the size of $S_n(B)$. In other words we measure the proportion of unique n-grams in `B` that are found in `A`. The score ranges from 0 to 1, indicating none to all newspaper copy shared with PA respectively.

We also compare texts by counting only those ngrams with low frequency, in particular those occurring once. For 1-grams, this is the same as comparing the *hapax legomena* which has been shown to discriminate plagiarised texts from those written independently even when lexical overlap between the texts is already high (e.g. 70%) (Finlay, 1999). Unlike Finlay's work, we

[1] The newspapers include five *popular* papers (e.g. *The Sun*, *The Daily Mail*, *Daily Star*, *Daily Mirror*) and four *quality* papers (e.g. *Daily Telegraph*, *The Guardian*, *The Independent* and *The Times*).

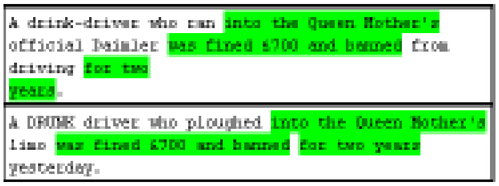
find that repetition in PA copy[2] drastically reduces the number of shared hapax legomena thereby inhibiting classification of derived and non-derived texts. Therefore we compute the containment of hapax legomena (hapax containment) by comparing words occurring once in the newspaper, i.e. those 1-grams in $S_1(B)$ that occur once with *all* 1-grams in PA copy, $S_1(A)$. This containment score represents the number of newspaper hapax legomena also appearing at least once in PA copy.

4.2 Greedy String-Tiling

Greedy String-Tiling (GST) is a substring matching algorithm which computes the degree of similarity between two strings, for example software code, free text or biological subsequences (Wise, 1996). Compared with previous algorithms for computing string similarity, such as the Longest Common Subsequence or

Levenshtein distance, GST is able to deal with transposition of tokens (in earlier approaches transposition is seen as a number of single insertions/deletions rather than a single block move).

The GST algorithm performs a 1:1 matching of tokens between two strings so that as much of one token stream is covered with *maximal* length substrings from the other (called *tiles*). In our problem, we consider how much newspaper text can be maximally covered by words from PA copy. A minimum match length (MML) can be used to avoid spurious matches (e.g. of 1 or 2 tokens) and the resulting similarity between the strings can be expressed as a quantitative similarity match or a qualitative list of common substrings. Figure 1 shows the result of GST for the example in Section 2.



A drink-driver who ran into the Queen Mother's
official Daimler was fined £700 and banned from
driving for two
years.

A DRUNK driver who ploughed into the Queen Mother's
limo was fined £700 and banned for two years
yesterday.

Figure 1: Example GST results (MML=3)

Given PA copy `A`, a newspaper text `B` and a set of maximal matches, `tiles`, of a given *length* between A and B, the similarity, `gstsim(A,B)`, is expressed as:

$$gstsim(A,B) = \frac{\sum_{i \in tiles} length_i}{\mid B \mid} \qquad (2)$$

4.3 Sentence alignment

In the past decade, various alignment algorithms have been suggested for aligning multilingual parallel corpora (Wu, 2000). These algorithms have been used to map translation equivalents across different languages. In this specific case, we investigate whether alignment can map derived texts (or parts of them) to their source texts. PA copy may be subject to various changes during text reuse, e.g. a single sentence may derive from parts of several source sentences. Therefore, strong correlations of sentence length between the derived and source sentences cannot be guaranteed. As a result, sentence-length based statistical alignment algorithms (Brown et al., 1991; Gale and Church, 1993) are not appropriate for this case. On the other hand, cognate-based algorithms (Simard et al., 1992; Melamed, 1999) are more efficient for coping with change of text format. Therefore, a cognate-based approach is adopted for the METER task. Here cognates are defined as pairs of terms that are identical, share the same stems, or are substitutable in the given context.

The algorithm consists of two principal components: a comparison strategy and a scoring function. In brief, the comparison works as follows (more details may be found in Piao (2001)). For each sentence in the candidate derived text DT the sentences in the candidate source text ST are compared in order to find the best match. A DT sentence is allowed to match up to three possibly non-consecutive ST sentences. The candidate pair with the highest score (see below) above a threshold is accepted as a true alignment. If no such candidate is found, the DT sentence is assumed to be independent of the ST. Based on individual DT sentence alignments, the overall possibility of derivation for the DT is estimated with a score ranging be-

[2] As stories unfold, PA release copy with new, as well as previous versions of the story

tween 0 and 1. This score reflects the proportion of aligned sentences in the newspaper text. Note that not only may multiple sentences in the ST be aligned with a single sentence in the DT, but also multiple sentences in the DT may be aligned with one sentence in the ST.

Given a candidate derived sentence *DS* and a proposed (set of) source sentence(s) *SS*, the scoring function works as follows. Three basic measures are computed for each pair of candidate *DS* and *SS*: SNG is the sum of lengths of the maximum length non-overlapping shared n-grams with $n \geq 2$; SWD is the number of matched words sharing stems not in an n-gram figuring in SNG; and SUB is the number of substitutable terms (mainly synonyms) not figuring in SNG or SWD. Let L_1 be the length of the candidate *DS* and L_2 the length of candidate *SS*. Then, three scores PD, PS (Dice score) and PVS are calculated as follows:

$$
\begin{aligned}
PSD &= \frac{SWD + SNG + SUB}{L_1} \\
PS &= \frac{2(SWD + SNG + SUB)}{L_1 + L_2} \\
PSNG &= \frac{SNG}{SWD + SNG + SUB}
\end{aligned}
$$

These three scores reflect different aspects of relations between the candidate *DS* and *SS*:

1. *PSD:* The proportion of the *DS* which is shared material.
2. *PS:* The proportion of shared terms in *DS* and *SS*. This measure prefers *SS*'s which not only contain many terms in the *DS*, but also do not contain many additional terms.
3. *PSNG:* The proportion of matching n-grams amongst the shared terms. This measure captures the intuition that sentences sharing not only words, but word sequences are more likely to be related.

These three scores are weighted and combined together to provide an alignment metric WS (weighted score), which is calculated as follows:

$$WS = \delta_1 \mathrm{PSD} + \delta_2 \mathrm{PS} + \delta_3 \mathrm{PSNG}$$

where $\delta_1 + \delta_2 + \delta_3 = 1$. The three weighting variables $\delta_i (i = 1, 2, 3)$ have been determined empirically and are currently set to: $\delta_1 = 0.85, \delta_2 = 0.05, \delta_3 = 0.1$.

5 Reuse Classifiers

To evaluate the previous approaches for measuring text reuse at the *document-level*, we cast the problem into one of a supervised learning task.

5.1 Experimental Setup

We used similarity scores as attributes for a machine learning algorithm and used the Weka 3.2 software (Witten and Frank, 2000). Because of the small number of examples, we used tenfold cross-validation repeated 10 times (i.e. 10 *runs*) and combined this with *stratification* to ensure approximately the same proportion of samples from each class were used in each fold of the cross-validation. All 769 newspaper texts from the `courts` domain were used for evaluation and randomly permuted to generate 10 sets. For each newspaper text, we compared PA source texts from the *same* story to create results in the form: $newspaper, class, score$. These results were ordered according to each set to create the same 10 datasets for each approach thereby enabling comparison.

Using this data we first trained five single-feature Naive Bayes classifiers to do the ternary classification task. The feature in each case was a variant of one of the three similarity measures described in Section 4, computed between the two texts in the training set. The target classification value was the reuse classification category from the corpus. A Naive Bayes classifier was used because of its success in previous classification tasks, however we are aware of its naive assumptions that attributes are assumed independent and data to be normally distributed.

We evaluated results using the F_1-measure (harmonic mean of precision and recall given equal weighting). For each run, we calculated the average F_1 score across the classes. The overall average F_1-measure scores were computed from the 10 runs for each class (a single accuracy measure would suffice but the Weka package outputs F_1-measures). For the 10 runs, the standard deviation of F_1 scores was computed for each class and F_1 scores between all approaches were tested for statistical significance using 1-way analysis of variance at a 99% confidence-level. Statistical differences between results were identified using Bonferroni analysis[3].

After examining the results of these single feature classifiers, we also trained a "combined" classifier using a correlation-based filter approach (Hall and Smith, 1999) to select the combination of features giving the highest classification score (correlation-based filtering evaluates all possible combinations of features). Feature selection was carried for each fold during cross-validation and features used in all 10 folds were chosen as candidates. Those which occurred in at least 5 of the 10 runs formed the final selection.

We also tried splitting the training data into various binary partitions (e.g. WD/PD vs. ND) and training binary classifiers, using feature selection, to see how well binary classification could be performed. Eskin and Bogosian (1998) have observed that using cascaded binary classifiers, each of which splits the data well, may work better on n-ary classification problems than a single n-way classifier. We then computed how well such a cascaded classifier should perform using the best binary classifier results.

5.2 Results

Table 1 shows the results of the single ternary classifiers. The baseline F_1 measure is based upon the prior probability of a document falling into one of the classes. The figures in parenthesis are the standard deviations for the F_1 scores across the ten evaluation runs. The final row shows the results for combining features selected using the correlation-based filter.

Table 2 shows the result of training binary classifiers using feature selection to select the most discriminating features for various binary splits of the training data.

For both ternary and binary classifiers feature selection produced better results than using all possible features, with the one exception of the binary classification between PD and ND.

Approach	**Category**	**Avg F-measure**
Baseline	WD	0.340 (0.000)
	PD	0.444 (0.000)
	ND	0.216 (0.000)
	total	**0.333** (0.000)
3-gram containment	WD	0.631 (0.004)
	PD	0.624 (0.004)
	ND	0.549 (0.005)
	total	**0.601** (0.003)
GST Sim MML = 3	WD	0.669 (0.004)
	PD	0.633 (0.003)
	ND	0.556 (0.004)
	total	**0.620** (0.002)
GST Sim MML = 1	WD	0.681 (0.003)
	PD	0.634 (0.003)
	ND	0.559 (0.008)
	total	**0.625** (0.004)
1-gram containment	WD	0.718 (0.003)
	PD	0.643 (0.003)
	ND	0.551 (0.006)
	total	**0.638** (0.003)
Alignment	WD	0.774 (0.003)
	PD	0.624 (0.005)
	ND	0.537 (0.007)
	total	**0.645** (0.004)
hapax containment	WD	0.736 (0.003)
	PD	0.654 (0.003)
	ND	0.549 (0.010)
	total	**0.646** (0.004)
hapax cont. 1-gram cont. alignment ("combined")	WD	0.756 (0.002)
	PD	0.599 (0.006)
	ND	0.629 (0.008)
	total	**0.664** (0.004)

Table 1: A summary of classification results

5.3 Discussion

From Table 1, we find that all classifier results are significantly higher than the baseline (at $p < 0.01$) and all differences are significant except between hapax containment and alignment. The highest F-measure for the 3-class problem is 0.664 for the "combined" classifier, which is significantly greater than 0.651 obtained without. We notice that highest WD classification is with alignment at 0.774, highest PD classification is 0.654 with hapax containment and highest ND classification is 0.629 with combined features. Using hapax containment gives higher results than 1-gram containment alone and in fact provides results as good as or better than the more complex sentence alignment and GST approaches.

Previous research by (Lyon et al., 2001) and (Wise, 1996) had shown derived texts could be distinguished using trigram overlap and tiling with a match length of 3 or more, respectively.

[3] Using SPSS v10.0 for Windows.

	Attributes	**Category**	**Avg F_1**
Correlation-based filter	alignment	WD	0.942 (0.008)
		ND	0.909 (0.011)
		total	**0.926** (0.010)
	alignment	PD/ND	0.870 (0.003)
		WD	0.770 (0.003)
		total	**0.820** (0.002)
	alignment	WD	0.778 (0.003)
		PD	0.812 (0.002)
		total	**0.789** (0.002)
	hapax cont.	WD/PD	0.882 (0.002)
	alignment	ND	0.649 (0.007)
	1-gram cont.	total	**0.763** (0.002)
	1-gram	PD	0.802 (0.002)
	GST mml 3	ND	0.638 (0.007)
	GST mml 1	total	**0.720** (0.004)
	alignment		
	GST mml 1	WD/ND	0.672 (0.002)
	alignment	PD	0.662 (0.003)
		total	**0.668** (0.003)

Table 2: Binary Classifiers with feature selection

However, our results run counter to this because the highest classification scores are obtained with 1-grams and an MML of 1, i.e. as n or MML length increases, the F_1 scores decrease. We believe this results from two factors which are characteristic of reuse in journalism. First, since even ND texts are thematically similar (same events being described) there is high likelihood of coincidental overlap of ngrams of length 3 or more (e.g. quoted speech). Secondly, when journalists rewrite it is rare for them not to vary the source.

For the intended application – helping the PA to monitor text reuse – the cost of different misclassifications is not equal. If the classifier makes a mistake, it is better that WD and ND texts are mis-classified as PD, and PD as WD. Given the difference in distribution of documents across classes where PD contains the most documents, the classifier will be biased towards this class anyway as required. Table 3 shows the confusion matrix for the combined ternary classifier.

	WD	PD	ND
WD	203	55	4
PD	79	192	70
ND	3	53	109

Table 3: Confusion matrix for combined ternary classifier

Although the overall F_1-measure score is low (0.664), mis-classification of both WD as ND and ND as WD is also very low, as most misclassifications are as PD. Note the high misclassification of PD as both WD and ND, reflecting the difficulty of separating this class.

From Table 2, we find alignment is a selected feature for each binary partition of the data. The highest binary classification is achieved between the WD and ND classes using alignment only, and the highest three scores show WD is the easiest class to separate from the others. The PD class is the hardest to isolate, reflecting the mis-classifications seen in Table 3.

To predict how well a cascaded binary classifier will perform we can reason as follows. From the preceding discussion we see that WD can be separated most accurately; hence we choose WD versus PD/ND as the first binary classifier. This forces the second classifier to be PD versus ND. From the results in Table 2 and the following equation to compute the F_1 measure for a two-stage binary classifier

$$\frac{WD + (PD/ND)(\frac{PD+ND}{2})}{2}$$

we obtain an overall F_1 measure for ternary classication of 0.703, which is significantly higher than the best single stage ternary classifier.

6 Conclusions

In this paper we have investigated *text reuse* in the context of the reuse of news agency copy, an area of theoretical and practical interest. We present a conceptual framework in which we measure reuse and based on which the METER corpus has been constructed. We have presented the results of using similarity scores, computed using n-gram containment, Greedy String Tiling and an alignment algorithm, as attributes for a supervised learning algorithm faced with the task of learning how to classify newspaper stories as to whether they are wholly, partially or non-derived from a news agency source. We show that the best single feature ternary classifier uses either alignment or simple hapax containment measures and that a cascaded binary classifier using a combination of features can outperform this.

The results are lower than one might like, and reflect the problems of measuring journalistic reuse, stemming from complex editing transformations and the high amount of verbatim text overlapping as a result of thematic similarity and "expected" similarity due to, e.g., direct/indirect quotes. Given the relative closeness of results obtained by all approaches we have considered, we speculate that any comparison method based upon lexical similarity will probably not improve classification results by much. Perhaps improved performance at this task may possible by using more advanced natural language processing techniques, e.g. better modeling of the lexical variation and syntactic transformation that goes on in journalistic reuse. Nevertheless the results we have obtained are strong enough in some cases (e.g. wholly derived texts can be identified with $> 80\%$ accuracy) to begin to be exploited.

In summary measuring text reuse is an exciting new area that will have a number of applications, in particular, but not limited to, monitoring and controlling the copy produced by a newswire.

7 Future work

We are adapting the GST algorithm to deal with simple rewrites (e.g. synonym substitution) and to observe the effects of rewriting upon finding longest common substrings. We are also experimenting using the more detailed METER corpus lexical-level annotations to investigate how well the GST and ngrams approaches can identify reuse at this level.

A prototype browser-based demo of both the GST algorithm and alignment program, allowing users to test arbitrary text pairs for similarity, is now available[4] and will continue to be enhanced.

Acknowledgements

The authors would like to acknowledge the UK Engineering and Physical Sciences Research Council for funding the METER project (GR/M34041). Thanks also to Mark Hepple for helpful comments on earlier drafts.

References

P.F. Brown, J.C. Lai, and R.L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Assoc. for Computational Linguistics*, pages 169–176, Berkeley, CA, USA.

P Clough. 2000. Plagiarism in natural and programming languages: An overview of current tools and technologies. Technical Report CS-00-05, Dept. of Computer Science, University of Sheffield, UK.

E. Eskin and M. Bogosian. 1998. Classifying text documents using modular categories and linguistically motivated indicators. In *AAAI-98 Workshop on Learning for Text Classification.*

H. Evans. 1972. *Essential English for Journalists, Editors and Writers.* Pimlico, London.

S. Finlay. 1999. Copycatch. Master's thesis, Dept. of English. University of Birmingham.

R. Gaizauskas, J. Foster, Y. Wilks, J. Arundel, P. Clough, and S. Piao. 2001. The meter corpus: A corpus for analysing journalistic text reuse. In *Proceedings of the Corpus Linguistics 2001 Conference*, pages 214—223.

W.A. Gale and K.W. Church. 1993. A program for aligning sentences in bilingual corpus. *Computational Linguistics*, 19:75–102.

M.A. Hall and L.A. Smith. 1999. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *Proceedings of the Florida Artificial Intelligence Symposium (FLAIRS-99)*, pages 235–239.

C. Lyon, J. Malcolm, and B. Dickerson. 2001. Detecting short passages of similar text in large document collections. In *Conference on Empirical Methods in Natural Language Processing (EMNLP2001)*, pages 118–125.

K. McKeown and H. Jing. 1999. The decomposition of human-written summary sentences. In *SIGIR 1999*, pages 129–136.

I. Dan Melamed. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, pages 107–130.

Scott S.L. Piao. 2001. Detecting and measuring text reuse via aligning texts. Research Memorandum CS-01-15, Dept. of Computer Science, University of Sheffield.

M. Simard, G. Foster, and P. Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 4th Int. Conf. on Theoretical and Methodological Issues in Machine Translation*, pages 67–81, Montreal, Canada.

M. Wise. 1996. Yap3: Improved detection of similarities in computer programs and other texts. In *Proceedings of SIGCSE'96*, pages 130–134, Philadelphia, USA.

I.H. Witten and E. Frank. 2000. *Datamining - practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann.

D. Wu. 2000. Alignment. In *R. Dale and H. Moisl and H. Somers (eds.),* A Handbook of Natural Language Processing, pages 415–458. New York: Marcel Dekker.

[4] See `http://www.dcs.shef.ac.uk/nlp/meter`.