A Tool for Text Comparison

Scott S.L. Piao and Tony McEnery
Lancaster University

Abstract

Text reuse is commonplace in academia and the media. An efficient algorithm for automatically detecting and measuring similar/related texts would have applications in corpus linguistics, historical studies and natural language engineering.

In an effort to explore the issue of text reuse, a tool, named Crouch[1], has been developed based on the TESAS system (Piao 2001) for comparing and measuring text similarity and derivation in sets of texts. Given a set of candidate source and derived texts, this tool maps related sentences between a pair of texts employing n-gram, stemming and synonym matching approaches. Crouch examines the textual similarity of individual pairs of texts, and also clusters pairs of texts in a collection of texts according to their similarity. The comparison is directional, in that the comparison from a derived text to its source generally produces a higher score than a comparison in the opposite direction. This presents the possibility of detecting the direction of text derivation. The tool displays its comparison of a given pair of texts in a graphical interface to help users to analyse the texts. Furthermore, as the tool is written in Java and fully supports Unicode, it can be applied to many languages. At Lancaster University, it is currently being used to help detect related English newspaper articles in 17th century newspapers.

1. Introduction

It is common practice in the publishing world to reuse text in producing other texts. This practice has grown increasingly common as large amounts of electronic texts have become readily accessible to anyone possessing a networked computer. While text copying and reuse might be regarded by some as a form of plagiarism, it is often a completely legal practice. For example, in the media industry, journalists subscribe to newswire services, such as the UK Press Association (PA), and quite legitimately reuse/modify texts released from these services when writing newspaper reports (Gaizauskas *et al*. 2001, Clough et al. 2002a). Today, on the Internet, large numbers of texts carrying similar or related content are being produced everyday, some of which are produced by reusing other texts. Newspaper texts, in particular, provide interesting material for observing and gaining insight into the practice of text reuse. An efficient algorithm for automatically detecting such texts and measuring relations between them can be useful for both academic research and practical language engineering tasks. Yet, text reuse in English newspapers is as old as the newspaper industry itself. Our goal in developing Crouch was to explore text reuse in early English newspapers[2], called newsbooks, produced in the English Commonwealth[3]. In this paper we describe a tool, named Crouch, developed for this purpose at Lancaster University. Crouch is based on a text-comparison tool, TESAS, which was developed in the METER Project in Sheffield University to identify British newspaper articles reusing texts released by the PA (Piao 2001). Written in Java code, Crouch fully supports Unicode and can potentially be used on many languages.

2. Related works

Recently, a number of related works have explored issues related to text reuse. For example, Parker and Hamblen (1989) tested several algorithms for detecting student plagiarism in program assignments. Mander (1994) described a tool, called *sif*, which, given a query text, can find similar texts from a large collection of texts. Brin *et al*. (1995) designed a system based on sentence overlaps, named COPS, which can detects copies or partial copies of chunks across texts. Shivakuma *et al*. (1995) suggested a scheme for text reuse identification, named SCAM, based on word occurrence frequencies. Their similarity metric reflects both relative frequencies of words and text subsets overlapping. Another

[1] The package is named after John Crouch, an early English satirist, Royalist and newsbook publisher in the English Commonwealth.

[2] The work outlined in this paper was supported by the British Academy, grant reference SG-33825.

[3] While these are not the earliest English newsbooks, as newsbooks of sorts appeared in the reign of Henry VIII, they do come from a period in which the newsbook had become a relatively popular and stable genre of writing. See Cranfield (1978) for an excellent history of the early English newsbooks.

relevant approach is that of Cho *et al.* (2000) who applied similarity metrics to filter out similar web pages in a web search engine.

However, the project on which the research presented here is principally based is the METER Project, in which a number of text reuse algorithms were tested (Clough *et al*., 2002b). One of the main results of the METER project was a prototype tool, named TESAS, which was designed for detecting and measuring journalistic text reuse (Piao 2001). The data to be processed was a comparative corpus, the METER Corpus, which is a collection of PA newswire texts and newspaper articles occurring in nine British newspapers (Gaizauskas *et al*. 2001, Clough *et al*. 2002a). In this corpus some of the newspaper articles are derived from the PA texts while others are written independently of the PA texts. In an evaluation of the METER corpus, in which the relations between the PA texts and newspaper articles are manually marked-up by a journalist, this tool obtained 85.44% precision with 86.85% recall in distinguishing WD (wholly derived) from PD (partially derived) and ND (non-derived) texts. It achieved a 73.03% success rate with 78.97% recall when distinguishing WD+PD versus ND texts. As we decided to base our system on the TESAS algorithms, we need to outline the TESAS algorithm in some detail here. As we explain in the following section, the core algorithm driving the TESAS system is based on the concept of text alignment.

3. The TESAS algorithm

Piao (2001) proposed an algorithm for identifying text reuse based on text alignment. It is assumed that the relationship between a pair of texts can be determined by examining the relationship between sub-units of the texts. Assuming that we can detect relationships between sub-units, sentences in our case, we can assess the overall relationship between whole texts based on these sentence level relationships. Reflecting this assumption, the algorithm consists of two main stages. Given a pair of texts, namely, candidate derived text *A* and source text *B*:

i.) Align sentences in *A* to their source sentence(s) in *B*, if they exist;
ii.) Measure the likelihood for *A* being derived from *B* using the sentence alignment.

Text alignment is a concept mainly used for processing multilingual parallel corpora (Oakes and McEnery 2000, Wu 2000). Statistical and linguistically motivated approaches to alignment, such as co-occurrence coefficients and machine-readable bilingual dictionaries, are used for identifying translations across different languages. Yet in looking at newspapers, we are working on monolingual data. This allows a range of new approaches to alignment to be tried in an attempt to discover sentence alignments, including n-gram matching, stemming and synonym matching. The use of these techniques is described in the following section.

3.1. Searching for text alignments at the sentence level

In the first stage of the TESAS algorithm, the algorithm searches for alignments at the sentence level. Given a pair of candidate derived text $T_{drv} = \{x_1, x_2, \ldots, x_m\}$ and a candidate source text $T_{src} = \{y_1, y_2, \ldots, y_n\}$, where x_i and y_i denote sentences, the algorithm splits the candidate texts into sentences. For each sentence x_i in T_{drv} all of the sentences y_j in T_{src} are scanned and compared against it in order to identify source sentence(s) for x_i. Because the sequence of sentences can be changed by journalists during rewriting, sentence y_j from any part of T_{src} is considered as a potential source sentence of a given x_i.

It was observed that a single sentence can de derived from multiple source sentences, mostly shorter ones, while multiple sentences can be derived from a single source sentence, that is to say, a long source sentence can be broken down into several sentences in a derived text. Consequently, in addition to the one-to-one match, sentence combinations $y_j + y_k + \ldots$ ($j, k = 1, 2, \ldots n$) are also included in the candidate sources of x_i ($i = 1, 2, \ldots, m$)[4]. Furthermore, those combined sentences are not limited to consecutive sentences. This is because a single PA text may contain duplicated materials, so a journalist can pick up any one of the same or similar text segments from different parts of it.

For each pair of candidate derived sentence and source sentence(s), three main approaches are used for detecting shared tokens between them:

[4] To avoid an excessive computation overhead, the number of sentences in a combined source is limited to three.

1) N-gram matching (n>=2). The n-grams shared between the candidates are identified. These n-grams provide the main factor for assessing the relations between the candidates.
2) Extended Porter's English stemmer (Porter, 1980). Porter's algorithm reduces inflected variants of an English word into a base form. It is extended to cope with irregular inflectional changes to English words.
3) Synonym map extracted from WordNet. A list of English synonyms containing about 46,000 entries was extracted from WordNet. Each entry contains two or more basic synonyms.

It should be noted that the stems that Porter's algorithm produces are not necessarily grammatical forms, hence we refer to them as base forms. For example, Porter's algorithm converts the word "degree" and "degrees" into the base form "degre". As it is an efficient tool for identifying inflectional variants of a single English word, Porter's algorithm has been widely used in natural language engineering. However, the original Porter stemmer cannot deal with irregular inflectional forms such as "thought" and "drank". Consequently, it has been extended to cope with such words.

Using the three approaches listed above, three kinds of shared items between candidate sentence pairs are identified: a) n-grams, b) identical single words, and c) word pairs that are synonyms or which have the same stem. Based on the matches, the relationship between the candidates is quantified in terms of three scores: *psd* (proportion of shared terms in candidate derived text), *dc* (mono-gram based Dice score) and *psng* (proportion of n-grams among the matched terms), which are calculated as follows.

Let l_{sw} be the number of single words matched between the candidates, including identical words, those sharing the same stem, and those that are found to be synonyms of each other via the synonym list. Let l_{ng} be the sum of the length of shared n-grams in terms of the number of tokens (n >=2). Let m_1 and m_2 be the lengths of candidate source sentence(s) and the candidate derived sentence. The scores are then defined as follows:

(1) $$psd = \frac{l_{sw} + l_{ng}}{m_2}$$

(2) $$dc = \frac{2 \times (l_{sw} + l_{ng})}{m_1 + m_2}$$

(3) $$psng = \frac{l_{ng}}{l_{sw} + l_{ng}}$$

Each of these scores reflects different aspects of the relationship between the candidate sentences. Firstly, the *psd* score indicates the extent to which x_i is dependent on the candidate source sentence(s). Its value ranges between 0 and 1. Its maximum score, 1, means that every word in the candidate derived sentence x_i can be traced back to sentence(s) in T_{src}; its minimum score of zero indicates that no word in x_i has a match in T_{src}. Secondly, the *dc* score reflects the similarity between the candidates. Ranging between 0 and 1, the maximum *dc*-score indicates every word in the candidate sentences has a match in the opposite side. If *dc* is zero, it implies that the candidate sentences are completely unrelated. Thirdly, the *psng* score denotes the reliability of the matched terms, ranging between 0 and 1. The maximum score, 1, of *psng* indicates that all of the matched items are n-grams; the minimal score of zero means that all of the matched items are single words. It is assumed, generally speaking, that closely related sentences tend to share longer text strings. Consequently, a higher *psng* score implies a closer tie between the candidate sentences, and vice versa.

It was found that function words and some highly frequent common words cause noise in the matching process. Such words are omnipresent in a text, as they perform various grammatical functions. Any pair of texts, related or not, share a number of these words. Moreover, they typically make little contribution to the semantic content of the texts. In order to reduce the noise, these words have been excluded from the matching process using a stop list of about 200 words. Filtering out the words makes the scores more effective.

Based on the three scores obtained from a candidate sentence pair x and y, we assess the possibility for x being derived from y. In order to obtain a single metric for measuring the relationship, the three scores are combined together to create a weighted score (*ws*) as follows:

(4) $$ws = \delta_1 psd + \delta_2 dice + \delta_3 psng$$

where δ_1 , δ_2 and δ_3 are weights for each of the three scores. These parameters were manually estimated by testing conducted on training data from the METER Corpus: $\delta_1 = 0.85$, $\delta_2 = 0.05$ and $\delta_3 = 0.1$. A threshold of *ws*, 0.65 by default, is used as the threshold to determine whether or not a pair of candidates are truly related. Those candidate pairs which produce a *ws* score higher than the threshold are taken to be truly related.

When the sentence alignment algorithm is applied to a pair of candidate texts T_{drv} and T_{src}, it produces a map linking each sentence x_i in T_{drv} to their source sentence(s) z_i in T_{src}: *align_map* = $\{(x_1, z_1, ws_1), (x_2, z_2, ws_2), \ldots, (x_i, z_i, ws_i), \ldots, (x_m, z_m, ws_m)\}$. Here z_i can be a single sentence or a combination of sentences from anywhere in T_{src}, and ws_i is the *ws*-score for the *i*th sentence pair. If a candidate derived sentence x_i cannot be mapped to any sentence(s) in T_{src}, it is aligned to a null sentence, or z_i = *null* and $ws_i = 0$.

3.2. Assessing the likelihood of text reuse

In the first stage described in the previous section, the TESAS algorithm produces an *align_map*, mapping sentences in the candidate derived text to their source sentences, including null alignment if no match can be found. In addition, the scores reflecting the degree of relation between each of the sentence matches are extracted. Based on this map, we proceed to assess the overall likelihood of a candidate text being derived from the candidate source.

Currently, given a pair of candidate texts T_{drv} and T_{src}, the likelihood for T_{drv} being related to T_{src} is estimated by measuring the proportion of the candidate alignment between T_{drv} and T_{src}. Three scores were tested for this purpose, which are calculated as follows:

Let mw_i denote a number of matched words in the *i*th aligned sentence in candidate T_{drv}, as_i denote the length of *i*th aligned sentence in terms of number of words, ws_i denote the weighted score for the *i*th sentence alignment, and *l* denote the length of text T_{drv}, then

(5) $$p_1 = \frac{\sum mw_i}{l}$$

(6) $$p_2 = \frac{\sum (as_i \times ws_i)}{l}$$

(7) $$p_3 = \frac{\sum as_i}{l}$$

All of these scores range between 0 and 1. In an experiment on data from the METER Corpus, the p_2-score produced the best result (Piao, 2001), thus it is used as the default metric for measuring relations between whole texts. Nevertheless the possibility exists that the other scores may be more efficient in other circumstances, or there are better metrics to assess such relations between texts.

3.3. Multilinguality and user-friendly interface

The TESAS tool is coded in Java, and fully supports Unicode. Consequently, this tool has the potential to be used on data encoded in a range of writing systems, though it must be noted that for TEAS to work on languages other than English the stemmer/lemmatiser and synonym mapping tool for the corresponding language would need to be developed.

Another important feature of TESAS is its user-friendly graphical interface. In order to assist human users to further analyse the aligned texts, a graphical interface was designed to visualise the output of the package. For example, the likelihood of sentences being derived from other texts is illustrated in charts, and the mapped sentences are displayed side by side in a table, with different types of matched words highlighted in different colours. At Lancaster University, this tool has been modified and improved to explore the issue of text reuse in early newsbooks, as described in the following section.

4. Crouch, a tool for exploring text reuse in 17th century English newspapers

In a project carried out at Lancaster University, an 800,000 word corpus (the Newsbook Corpus hereafter) has been built from English newsbooks from the mid-17th century[5]. The corpus was built

[5] See http://www.ling.lancs.ac.uk/newsbooks.

specifically to investigate text reuse in newsbooks of this period. It is impractical, if not impossible, to manually carry out an exhaustively comparison of all of the texts in the corpus. Hence, while our research goal was similar to that carried out on the METER corpus, an investigation of text reuse, we were also faced with the problem of trying to identify the texts across which *sharing of text* occurred before we could carry out the alignment process. TESAS, as a system, is able to identify text reuse between a pair of candidate texts. Providing the system with texts in which text reuse occurs is relatively easy with the METER corpus, as the corpus was composed of texts in which text reuse was known to have occurred. However, in the case of the newsbook corpus, we simply had 800,000 words of newspapers in which we believed text reuse had taken place. We had neither the time nor the resources to manually identify which newsbooks had copied from which, and hence needed a program which would not simply align a candidate pair, but which could use alignment in order to identify which newsbooks had copied text from which newsbook across the entire data set automatically; we needed a tool which could automatically identify, measure and cluster related texts. Although it is unlikely that such a tool could analyse the texts as comprehensively as human experts, it can be of assistance to human experts, i.e. it will identify the bulk, if not all, of the newsbooks which reused text. We also needed to adapt the program to deal with spelling variations, and some textual quirks of the newsbooks. The following sections outline our approaches to dealing with these three issues.

4.1. Clustering texts by similarity

The algorithms described so far examine and measure the derivational relation between a pair of texts. But before we examine texts in detail, we first need to collect related texts to examine. While this may be a trivial manual task if we are studying a dozen texts, it becomes very difficult in the context of an 800,000 word corpus. In order to search for related texts in a large collection of texts, we need a tool to automatise this process. In response to this need, we developed a tool for clustering texts based on the p_2 score (see formula 6).

The first step is to extract a matrix of similarity distances between the texts. Given a collection of texts $t_1, t_2, \ldots, t_n$, each of them is matched against the other. In this way, a matrix of similarity distance e_{ij} is obtained based on p_2 scores, as shown in Fig. 1.

Fig. 1: Matrix of text related

In this matrix, each element e_{ij} is calculated by the formula

(8) $$e_{ij} = 1 - p_2(i, j),$$

where $p_2(i, j)$ is the score of derivation between the ith text (as the candidate derived text) and the jth text (as the candidate source text). Ranging between 0 and 1, a smaller e-score indicates a greater similarity between texts, while larger scores indicate greater difference between texts.

In terms of text clustering, we chose Ward's (1963) approach as used by Phillips (1985: 72-84) who suggested that this algorithm is effective for clustering words by their collocational relationship. We use the same clustering procedure, though we replace the Euclidean distances with the similarity distance e_{ij}. In this algorithm, the Error Sum of Squares (*ESS*) is used as metric for measuring the tightness of clusters, which is calculated as follows:

(9) $$ESS = \sum_{clusters} \sum_{\substack{within \\ clusters}} \left[\sum_{k=1}^{p} \left(x_{ik} - x_k \right) \right]$$ (Phillips, 1985: 83)

where x_k denotes the mean value of the kth variables of all the entries within a cluster. The value of *ESS* increases as the strength of clustering decreases. Beginning from single texts, this algorithm tests all possible pairs, aiming to find the pair whose fusion results in the minimal increase of the total value of

ESS over all clusters. This is a recursive algorithm. As the algorithm proceeds, clusters are further clustered into larger clusters until only a single super cluster is left.

Fig. 2 shows an extract of a sample output of the clustering program, running on a set of newsbooks from January 1654. As shown in this figure, the grouped texts are enclosed in layers of curly brackets. The nodes in the deepest layer, i.e. those nodes numbered 1, contain the closest text pairs, or single texts left ungrouped. Again, these sub-nodes are clustered into a higher layer of groups, enclosed by brackets numbered 2, and so on, until all of the texts are included in a single top cluster, numbered 4 in this sample. Obviously, those text pairs enclosed in the clusters of deeper layer tend to be more tightly related than those linked through higher level clusters. We assume, therefore, that the text pairs in the clusters of bottom layer provide the most useful data for the study of text reuse.

Although far from perfect (see section 5 for more details of the effectiveness of the algorithm), this clustering algorithm provides a useful tool for human experts to collect candidate text pairs/groups for further examination. In particular, when we deal with large number of texts, such a tool becomes indispensable.

```
{ n=4
 { n=3
   { n=2
     { n=1 DutchDiurn03_B.utf8 TruePerfInf01_B.utf8 }
     { n=1 WPost$127_B.utf8 }
   }
   { n=2
     { n=1 PerfAcc158_B.utf8 WIntell149#2_B.utf8 }
   }
 }
 { n=3
   { n=2
     { n=1 MPol187_B.utf8 PerfAcc157_B.utf8 }
     { n=1 MPol188_B.utf8 PerfDiurn215_B.utf8 }
   }
 }
}
```

Fig. 2: A sample of clustered texts.

4.2. Mapping orthographic variants of the same words

One significant difficulty that we faced when attempting to identify text reuse in early modern English newsbooks was the problem of variant spelling. While some modern word forms are still subject to spelling variation (notably between British and American English, but also with respect to word endings such as *ise/ize*) the problem in early modern English is much more pronounced and pervasive. Individual authors differed, often quite markedly, with respect to how they spelt various words. Indeed, individual authors may themselves have been inconsistent over time in their spelling of a specific word. Standard spelling is a relatively modern concept (Robertson and Willet, 1991). To give a few examples of spelling variants from the Newsbook Corpus, the modern word form *useful* does occur, but it is also spelt as *usefull*. Similarly, the modern word form *coming*, occurs, but it is also spelt as *comming*.

While a human reader can identify such spelling variants with little difficulty, they present a challenge to an automatic text alignment system. In order to efficiently detect related texts in historical documents, a program needs to be able to identify spelling variants of the same words. One possible solution would be to collect such word variants into a mapping list. However, it would be difficult to exhaustively collect such word variants, as the productivity of the variant spellings, while not infinite, is certainly very large. Moreover, even if it were possible to collect all such word variants from a particular corpus representing a particular time period, that list would be less useful for other data from the same time period and certainly less useful for other time periods. Hence while a mapping list may provide a corpus specific solution for this problem, we rejected this solution as we wished to develop a solution to the program that would allow Crouch to work with different corpora from different periods of time.

To develop the more general solution to spelling variation, we reviewed recent research on spelling variation in historical databases. One of the major research projects was reported by Robertson *et al*. (1992, 1993). They tested several techniques for conflating modern and 17th-century English word equivalents, including a digram and trigram based Dice coefficient approach, SPEEDCOP, Dynamic

programming and machine learning using neural networks. In their comparative study, they reported that the digram matching and dynamic programming methods obtained the best results in matching modern and variant word forms. However, dynamic programming was found to be a very time-consuming algorithm due to its complex comparing algorithm, taking about 30 times as long as the digram approach. As the results of this research seemed to provide a promising approach to matching variant spellings, we decided to explore the use of the digram-based Dice coefficient for matching variant spellings in text alignment. The Dice coefficient is calculated as follows:

Given a pair of words w_1 and w_2, each of them is broken into digrams – subunits containing two adjacent letters. Next, the number of digrams shared between w_1 and w_2 is counted. Let d be the digram-based Dice coefficient, let l_1 and l_2 be the lengths of w_1 and w_2, and let k be the number of shared digrams, then

$$d = \frac{2 \times k}{l_1 + l_2} \tag{10}$$

Experiments show that this coefficient is effective in identifying similar word forms (McEnery *et al*. 1996, 1997). Nonetheless, spelling similarity alone is not sufficient for identifying true word variants. For example, in the Newsbook Corpus the words *gratifie* and *ratifie* have very similar forms yielding a *d*-score of 0.923, but in fact they are different words (corresponding to the modern word forms *gratify* and *ratify* respectively). In order to examine the efficiency of the dice-coefficient, an experiment was carried out as follows. Firstly, a pair of newsbooks sharing some text reuse, *Mercurius Politicus, Issue 189* and *Severall Proceedings of State Affaires, Issue 226*, were selected from the Newsbook Corpus. They contain 6,340 and 6,446 words respectively (we denote these text as *A* and *B* in the following discussion). Next, these texts were passed to a program which collects word matches between them which produce *d*-scores higher than a given threshold, 0.9 in this case. This program compares each word in text *A* against every word in text *B*, solely by *d*-score, without considering any contextual information. Table 1 lists the word pairs collected in this way. As shown in this table, altogether 26 word pairs were collected, of which 13 are true matches, resulting in a success rate of 50%.

Word from A	Word from B	check	Word from A	Word from B	check
aforesaid	foresaid	√	herein	therein	x
approven	approve	x	Innerness	Innernesse	√
asisted	assisted	√	Intelligencers	Intelligence	x
business	businesse	√	Minister	Ministery	x
comming	Coming	√	preparation	reparations	x
delivery	deliver	x	preparations	reparations	x
difficult	difficulty	x	reason	Treason	x
Garrison	Garison	√	several	severall	√
Generall	General	√	severall	several	√
Gottenburgh	Gottenburg	√	thereof	hereof	x
gratifie	ratifie	x	useful	usefull	√
Hamburgh	Hamburg	√	Whereof	hereof	x
Highness	Highnesse	√	Whereof	hereof	x

Table 1: Word pairs matched by Dice-score with threshold of 0.9.

We assume that a contextual constraint on the search area can help to improve the accuracy of matching word variants. We assume that if a pair of similar orthographic forms occur in a context within which text reuse is occuring, then they are more likely to be true word variants. As we explained previously, in the earlier stage our system aligns the sentences of a candidate derived text to their source sentence(s) in a candidate source text. If we constrain the search areas for word variants to those sentence pairs that are initially established as possible matches, we hypothesised that we were more likely to find true word variants. This is because such candidate sentence pairs provide identical or substantially similar contexts. We tested this idea by passing the same text pairs from the Newsbook Corpus to a program which searched for word variants only within the initially aligned sentence pairs.

Table 2 shows the word variant pairs collected solely from the aligned sentence pairs. As shown in this table, while the number of extracted word pairs is reduced, all of them are true variants of the same words. Although a wide scale test of this finding is necessary before we can claim that our hypothesis is correct, this preliminary experiment demonstrates the probable usefulness of contextual constraints in searching for word variants. Such an accurate identification of word variants can help to boost the efficiency of the alignment of related sentences in early modern English texts. As a by-product, this

algorithm can also be used to accurately and automatically collect word variants on a large scale. In Crouch, this algorithm is incorporated into the text-matching algorithm, and the number of identified word variants is added to the number of matched single words denoted by l_{sw} in formulae 1, 2 and 3 (see section 3.1).

Word	Substitute	check
asisted	assisted	√
comming	coming	√
Garrison	Garison	√
Innerness	Innernesse	√
useful	usefull	√

Table 2: Word pairs matched by Dice-score from aligned sentences with threshold of 0.9.

4.3. Filtering out false source sentence matches

In the original TESAS algorithm, for a candidate derived sentence x_i, its source can be a combination of sentences $y_j + y_k + \ldots$ from the source text (refer to section 3.1). The sentences y_j, y_k, … can be non-consecutive sentences, which may come from any part of the source text. This algorithm reflects two features of the METER Corpus:

a) A single text from the PA newswire service may contain duplicated or similar sections;
b) A journalist can combine multiple sentences from different parts of the PA text to form a new sentence.

Very often a newspaper sentence is matched to one of several identical/similar sentences, which are scattered across different sections of the PA text. Technically it is almost impossible to pinpoint which of them is the true source. As a result, either the first of them, in the case of multiple identical sentences, or the one producing the highest matching score, in the case of similar sentences, is chosen as the source.

When we tried the same algorithm on the Newsbook Corpus, we found that most of the con-consecutive sentences are false matches. This is due to the fact that each newspaper in the Newsbooks Corpus is the final version for publication, hence it is unlikely that it contains duplicated text fragments, as is the case for the newswire data. There was no direct seventeenth century equivalent of newswire services[6].

To reflect this distinct feature of the Newsbook Corpus, we adjusted the original algorithm to remove the non-consecutive sentences from the matched source. Suppose a candidate derived sentence x_i is aligned to multiple source sentences $y_j + y_k + \ldots..$ Firstly the sentences in the combined source are sorted in descending order by *ws*-scores (see formula 4) they produce with x_i. Taking the top sentence as the main part of the source, the following candidate sentences are checked for their location in relation to the top one. Those candidates which are adjacent to the main sentence, either preceding or following, are kept as part of the true source, while the others are filtered out.

Another typical type of false sentence alignments the original TESAS produces are those in which all of the matched items are numerical (both cardinal and ordinal) and days of the week or names of months, such as “Monday” or “January”. Because such words frequently co-occur across journalistic texts reporting events that happened in a similar time period, they cause coincidental matches between unrelated texts. In particular, when the candidate derived sentences are short, a few such coincidental matches can result in a high matching score, producing false alignments. Therefore, if a candidate derived sentence only matches on the basis of numbers/days/months, it is assumed not to match the source text.

[6] Though there were sources of data that the newsbooks commonly copied from – letters from foreign correspondents being a very typical example. However, these letters are now almost without exception no longer in existence.

5. Evaluation

Crouch was evaluated with reference to three of its operations on test data selected from the Newsbook Corpus. We tested: a) the alignment of related sentences, b) the clustering of related texts, and c) the effectiveness of filtering those sentence alignments in which only numbers/dates matched. For the test data, we chose from the corpus newsbooks that were published in January 1654, amounting to 33 newsbooks (amounting to 127,609 words). The text clustering algorithm in Crouch was run on these texts to cluster them. The threshold of the *ws*-score for sentence alignment (refer to formula 4) was manually set to 0.8 based on our initial experiments on some sample data from 1653. As mentioned previously, we assume that text pairs in the clusters of the deepest layer provide the most useful data for the study of text reuse. Therefore, for the evaluation, we collected only those clusters in the deepest layer. This layer contained 16 text pairs altogether.

Due to the difficulty of manually finding all of the true sentence alignments contained in the data, no attempt was made to examine recall in this evaluation. This will become possible only when and if a testbed corpus with all of the true sentence alignments in the newsbooks explicitly marked-up becomes available.

With respect to the sentence alignment evaluation, texts in each of the 16 text pairs were compared using Crouch. Sentence alignments, listed side by side in a table of Crouch's report package, were manually checked. The texts were compared twice. On the first occasion, the filter for clearing false alignments that only match numbers and dates (see section 4.3) was turned off. On the second occasion, this filter was turned on. Without using the filter a total of 662 sentence alignments were found. 555 of them were found to be true alignments, resulting in a success rate of 83.84%. After the filter was turned on, no new alignments were found, but 68 out of 107 false alignments were eliminated and no new false alignments generated. As a result, the total number of alignments and false alignments were reduced to 594 and 39 respectively and the success rate was increased to 93.43%.

Next, the 16 text clusters were manually examined by comparing the texts in each pair with the help of various functions in Crouch. After the examination, we concluded that two of the text clusters were incorrect clusters, i.e. we could not find any convincing evidence that the texts in each pair were related. We assume these false clusters are caused by the greedy nature of the algorithm which forces texts to form pair-wise clusters unless a single text is left. While this algorithm ensures that each text can find a mate which has the closest distance to it among the remaining candidates, sometimes it may impose a weakly-related mate on a text when no remaining candidates are closely related to it. We assume this problem can be solved by using a threshold of relational distance to filter out weakly related clusters.

Although our evaluation involved subjective judgements and further evaluation on larger scale is necessary in order to fully test Crouch's performance, our experiment shows Crouch is capable of detecting and measuring text reuse with a reasonably high rate of precision. As it stands, Crouch provides a practical tool for exploring the issue of text reuse in corpora.

6. Conclusion

Text reuse is an interesting and challenging issue for both academic research and practical language engineering tasks. In the past few years, some corpora focused on text reuse have been built, notably the METER Corpus in Sheffield and the Newsbook Corpus in Lancaster. However, further development work, notably on tools which facilitate the exploration of such corpora, is needed.

In this paper, we have described a program, Crouch, developed at Lancaster University which detects, measures and clusters reused texts in the Newsbook Corpus, a collection of English newsbooks published in the mid-17th century. This tool is based on a prototype tool, TESAS, developed on the METER project of Sheffield University. In Crouch, we have modified and improved existing TESAS functions and added new functions to TESAS. Our evaluation shows that Crouch, as it stands, already performs with a reasonably high rate of accuracy, providing both the corpus community and historians with a practical tool for exploring the issue of text reuse.

References:

Brin, Sergey, James Davis and Hector Garcia-Molina 1995 Copy detection mechanisms for degital documents. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, San Francisco, CA, pp. 398-409.

Cho, Junghoo, Narayanan Shivakumar and Hector Garcia-Molina 2000 Finding replicated web collections. In *Proceedings of the 2000 ACM SIGMOD Conference*, Dallas, Texas, pp. 355-366.

Clough, Paul, Robert Gaizauskas, S. L. Piao 2002a Building and annotating a corpus for the study of journalistic text reuse. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, vol. 5, Los Palmas de Gran Canaria, Spain, pp. 1678-1691. .

Clough, Paul, Robert Gaizauskas, Scott Piao, Yorick Wilks 2002b METER: MEasuring TExt Reuse, In *Proceedings of the ACL-2002*, University of Pennsylvania, Philadelphia, USA, pp. 152-159.

Cranfield, G.A. (1978) *The Press and Society*, Longman, London.

Gaizauskas, Robert, Jonathan Foster, Yorick Wilks, John Arundel, Paul Clough, and Scott Piao 2001 The METER Corpus: a corpus for analysing journalistic text reuse. In *The Proceedings of the Corpus Linguistic 2001*, Lancaster University, UK, pp. 214-223.

Levy, M. David 1993 Document reuse and document systems. *Electronic Publishing*, 6(4), pp 339-348.

Manber, Udi 1994 Finding similar files in a large file system. In *Proceedings of the USENIX Winter 1994 Technical Conference*, San Francisco, CA, USA, pp. 1-10.

McEnery, Tony and Michael Oakes 1996 Sentence and word alignment in the CRATER project. In Jenny Thomas and Mick Short (eds.), *Using Corpora for Language Research*, Longman, London, pp. 211-231.

McEnery, Tony, Jean-Marc Langé, Michael Oakes and Jean Véronis 1997 The exploitation of multilingual annotated corpora for term extraction. In Roger Garside, Geoffrey Leech and Anthony McEnery (eds.), *Corpus Annotation --- Linguistic Information from Computer Text Corpora*, Longman, London, pp. 220-230.

Oakes, M. and A.M. McEnery 2000 Bilingual text alignment -- an overview. In S.P. Botley and, A. M. McEnery and A. Wilson (eds.), *Multilingual Corpora in Teaching and Research*, Rodopi, Amsterdam-Atlanta, pp.1-37.

Parker, Alan and James O. Hamblen 1989 Computer algorithms for plagiarism detection. *IEEE Transactions on Education*, 32(2), pp. 94-99.

Phillips, Martin 1985 *Aspects of Text Structure – An Investigation of the Lexical Organisation of Text*, Elsevier Science Publishers B.V., Amsterdam.

Piao, Scott S.L. 2001 Detecting and measuring text reuse via aligning texts. Technical paper: CS-01-15, Department of Computer Science, University of Sheffield, UK.

Porter, M.F. 1980 An algorithm for suffix stripping. *Program*, 14(3), pp. 130-137.

Robertson, Alexander M. and Peter Willett 1991 Digram and trigram matching for the identification of word variants in historical text databases. In Tony Mcenery (ed.), *Proceedings of the British Computer Society 13th Information Retrieval Colloquium*, University of Lancaster, UK, pp. 12-21.

Robertson, Alexander M. and Peter Willett 1993 Evaluation of techniques for the conflation of modern and seventeenth century English spelling. In *Proceedings of the BCS 14th Information Retrieval Colloquium*, University of Lancaster, UK, pp. 155-168.

Shivakumar, N. and H. Garcia-Molina 1995 SCAM: A copy detection mechanism for digital documents. In Tony McEnery and Chris Paice (eds.) *Proceedings of 2st International Conference in Theory and Practice of Digital Libraries (DL '95)*. Austin, Texas.

Ward, J.H. 1963 Hierarchical grouping to optimise an objective function. *Journal of the American Statistical Association* 58(301), pp. 236-244.

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