Lang Resources & Evaluation (2011) 45:5-24
DOI 10.1007/s10579-009-9112-1

# Developing a corpus of plagiarised short answers

Paul Clough · Mark Stevenson

Published online: 16 January 2010

**Abstract** Plagiarism is widely acknowledged to be a significant and increasing problem for higher education institutions (McCabe 2005; Judge 2008). A wide range of solutions, including several commercial systems, have been proposed to assist the educator in the task of identifying plagiarised work, or even to detect them automatically. Direct comparison of these systems is made difficult by the problems in obtaining genuine examples of plagiarised student work. We describe our initial experiences with constructing a corpus consisting of answers to short questions in which plagiarism has been simulated. This corpus is designed to represent types of plagiarism that are not included in existing corpora and will be a useful addition to the set of resources available for the evaluation of plagiarism detection systems.

**Keywords** Plagiarism · Plagiarism detection · Corpus creation · Language resources

## 1 Introduction

In recent years, plagiarism (and its detection) has received much attention from both the academic and commercial communities (e.g. Hislop 1998; Joy and Luck 1999; Lyon et al. 2001; Collberg and Kobourov 2005; zu Eissen and Stein 2006; Kang et al. 2006). In academia students have used technology to fabricate texts (e.g. using

P. Clough (✉)
Department of Information Studies, University of Sheffield, Regent Court, 211 Portobello,
Sheffield S1 4DP, UK
e-mail: p.d.clough@sheffield.ac.uk

M. Stevenson
Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello,
Sheffield S1 4DP, UK
e-mail: m.stevenson@dcs.shef.ac.uk

pre-written texts from essay banks or paper mills, using word processors to manipulate texts and finding potential source texts using online search engines) and plagiarism is now widely acknowledged to be a significant and increasing problem for higher education institutions (Culwin and Lancaster 2001; Zobel 2004; McCabe 2005; Judge 2008).

A wide range of approaches to the detection of plagiarism have been suggested by the academic community, for example (Clough 2000; White and Joy 2004), and many commercial systems are also available (Bull et al. 2001). However, one of the barriers preventing a comparison among techniques is the lack of a standardised evaluation resource. Such a resource would enable a quantitative evaluation of existing techniques to plagiarism detection. Standardised evaluation resources have been very beneficial to a wide range of fields including Information Retrieval (Voorhees and Harman 2005), Natural Language Processing (Grishman and Sundheim 1996; Mihalcea et al. 2004) and authorship attribution (Juola 2006). Although proposals have been made for building such a resource, for example (zu Eissen et al. 2007; Cebrián et al. 2007) and the PAN Plagiarism Detection Competition 2009[1] (Potthast et al. 2009), little details are provided regarding their construction and they represent only specific types of plagiarism.

Unfortunately the process of creating a corpus of plagiarised documents is hampered by a number of problems that are not encountered in the majority of corpus construction tasks. Firstly, the act of plagiarism includes an element of deception; plagiarised text is not intended to be identified as such and those who plagiarise are unlikely to admit to doing so. Consequently the identification of plagiarised text may not be possible. In addition, even if it were possible to identify plagiarised documents, it is unlikely that they could be made freely available for research purposes. The document's writer is unlikely to agree to this and doing so is likely to be regarded as ethically, and perhaps also legally, unacceptable. These issues form a significant challenge to any attempt to create a benchmark corpus of plagiarised documents.

This paper describes the construction of a corpus of answers to short questions on a range of topics in Computer Science. To avoid the problems involved in collecting genuine examples of plagiarism we chose to simulate plagiarism by asking authors to intentionally reuse another document in a way which would normally be regarded as unacceptable (see Sect. 3). The corpus is not intended to comprehensively represent all possible types of plagiarism but does contain types which are not included in the resources that are currently available (see Sect. 2). The corpus is analysed both qualitatively, to gain insight into the strategies used by students when they plagiarise documents, and qualitatively, to determine how useful the various types of plagiarism contained in the documents are likely to be for the evaluation of systems (Sect. 4). It is suggested that this corpus forms a valuable addition to the set of already available resources for the plagiarism detection task. This corpus will (1) enable comparative evaluation between existing and new techniques for automated plagiarism detection, (2) help stimulate further research in the field, (3) help us to

[1] http://www.webis.de/pan-09.

understand the strategies used by students when they plagiarise, and (4) be of potential use as a pedagogical resource to provide examples of plagiarism.

## 2 Background

### 2.1 Varieties of plagiarism and their detection

A range of problems have been explored within the study of plagiarism and the type of problem influences the approach that is most appropriate for their detection. Stein (2006) distinguish *extrinsic* and *intrinsic* plagiarism analysis. In the first case the aim is to identify plagiarised portions of text *within* documents and the corresponding source; whilst the second case describes the scenario where the source does not need to be identified.

In extrinsic plagiarism analysis a key factor is the comparison of portions of text which it is suspected are plagiarised with their potential sources. This problem is made complex by the fact that there are a wide variety of "levels" of plagiarism. Martin (1994) points out that these include word-for-word plagiarism (direct copying of phrases or passages from another text without quotation or acknowledgment), paraphrasing plagiarism (when words or syntax are rewritten, but the source text can still be recognised) and plagiarism of ideas (the reuse of an original idea from a source text without dependence on the words or form of the source). Automatic approaches for detecting plagiarism within natural language originate from a diverse range of areas including file comparison (Heckel 1978; Manber 1994), information retrieval (Korfhage 1997; Sanderson 1997), authorship attribution (Woolls and Coulthard 1998; McEnery and Oakes 2000), file compression and copy detection (Brin et al. 1995; Broder 1998). These methods are typically most successful when the plagiarised texts have undergone minimal alterations, such as word-for-word plagiarism, but are unlikely to identify the source when it has been significantly changed. zu Eissen et al. (2007) and Pinto et al. (2009) also point out that the source could be written in a different language and have been translated (either automatically or manually) before being reused, a process which is likely to involve the text being significantly altered.

The problem, however, is a different one in the case of intrinsic plagiarism analysis. In this case the aim is to identify portions of text that are somehow distinct from the rest of the document in such a way that it raises suspicion in the reader, for example significant improvement in grammar or discussion of more advanced concepts than would be expected. Intrinsic plagiarism analysis is generally carried out by identifying portions of a text written in a different style from the remainder and this is often carried out using stylometric features including surface characteristics (e.g. average sentence/word length), readability measures (e.g. Flesch-Kincaid Reading Ease Flesch 1974), Coleman-Liau Index (1975) and syntactic characteristics (e.g. part of speech and syntactic structure).

There may also be variation in the number of source texts that have been plagiarised. A document may plagiarise a single source; the most extreme version of this situation is when an original document is copied verbatim and the author changed (Martin 1994). Plagiarism of this type may also include modifications to the original document or a plagiarised section being included as part of an otherwise acceptable document. Alternatively, a document may plagiarise from more than one source and, similarly, the document may consist only of plagiarised passages or plagiarised sections embedded within it and these passages may be modified or used verbatim.

### 2.2 Existing corpora

In order to evaluate approaches to plagiarism detection it is useful to have access to a corpus containing examples of the types of plagiarism that we aim to identify. Given the difficulties involved in obtaining examples of plagiarised texts, an attractive approach is to develop a corpus automatically. For example, zu Eissen et al. (2007) created a corpus for plagiarism detection experiments by manually adapting Computer Science articles from the ACM digital library that was made available to researchers with access to that collection (Web Technology & Information Systems Group 2008). Passages from other articles in the same collection were added to these documents to simulate plagiarism. Some of these passages were copied verbatim while others were altered. However, zu Eissen et al. (2007) do not describe the process of corpus creation in detail. A corpus was also automatically created for the 2009 PAN Plagiarism Detection Competition (Potthast et al. 2009). This resource contains texts of a wide range of lengths and exhibiting differing amounts of texts inserted from other documents. The reused text is either obfuscated, by randomly moving words or replacing them with a related lexical item, or translated from a Spanish or German source document. Guthrie et al. (2007) also simulated plagiarism by inserting a section of text written by another author into a document, although they did not alter the inserted text in any way.

This approach is convenient since it allows corpora of "plagiarised" documents to be created with little effort. In fact, if the inserted passages are not altered, as Guthrie et al. chose to do, the amount of documents that could be created are only limited by the size of the collection. However, it is not clear the extent to which these corpora reflect the types of plagiarism that might be encountered in academic settings.

While plagiarism is an unacceptable form of text re-use there are other forms of this practice that are not objectionable, such as the reuse of news agency text by newspapers. The METER Corpus[2] is a hand-crafted collection of 1,716 texts built specifically for the study of text reuse between newswire source texts and stories published in a range of British national newspapers (Clough et al. 2002). The corpus contains a collection of news stories between July 1999 and June 2000 in two domains: (1) law and court reporting, and (2) showbusiness and entertainment. The newspaper articles were analysed to identify the degree to which they were derived from the news agency source and annotated with a three level scheme that indicated whether the text was entirely, partially or not derived from the agency source. Almost half of the stories were analysed in more detail to identify whether the text

[2] http://www.dcs.shef.ac.uk/nlp/meter/Metercorpus/metercorpus.htm.

was extracted verbatim from the news agency text, rewritten or completely new. The METER corpus is freely available and contains detailed annotation at a level which could be very valuable in the development of plagiarism detection systems; however, the main drawback of this corpus is that the type of text reuse it represents is not plagiarism.

Plagiarism may involve attempts to disguise the source text and this may be attempted by paraphrasing (see Sect. 3.2 for further discussion). Within the field of Computational Linguistics there as been interest in the identification and generation of paraphrases over the last decade, for example (Barzilay and McKeown 2001; Callison-Burch et al. 2006). This has lead to the development of a variety of corpora containing examples of paraphrases and, while these do not represent text reuse, they are potentially valuable for evaluating some aspects of plagiarism detection. Example paraphrase corpora include, the Microsoft Research Paraphrase Corpus (MSRPC)[3] Dolan et al. (2004) contains almost 6,000 pairs of sentences obtained from Web news sources that have been manually labeled to indicate whether the two sentences are paraphrases or not. The Multiple-Translation Chinese Corpus[4] (see Pang et al. 2003) makes use of the fact that translators may choose different phrases when translating the same text. The corpus consists of 11 independent translations of 993 sentences of journalistic Mandarin Chinese text. Cohn et al. (2008) recently described a corpus[5] consisting of parallel texts in which paraphrases were manually annotated. While these resources are potentially useful in the development of plagiarism detection systems they are limited by the fact that, like the METER corpus, they consist of acceptable forms of text reuse.

The various corpora relevant to the plagiarism detection are limited since there is no guarantee that they represent the types of plagiarism that may be observed in practice. Artificially created corpora are attractive, since they allow data sets to be created quickly and efficiently, but may be limited to one type of plagiarism (insertion of reused section in an otherwise valid document) and, if the inserted text is altered, it may not be changed in the same way a student may choose to. In addition, the various resources based on acceptable forms of text reuse (including the METER corpus and paraphrase corpora) do not include the element of deception involved in plagiarism.

## 3 Corpus creation

We aim to create a corpus that could be used for the development and evaluation of plagiarism detection systems that reflects the types of plagiarism practiced by students in an academic setting as far as realistically possible. We decided to avoid the strategies used in the creation of related corpora (see Sect. 2.2) since these may

[3] http://www.research.microsoft.com/en-us/downloads/607D14D9-20CD-47E3-85BC-A2F65CD28042/default.aspx.

[4] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T01.

[5] http://www.homepages.inf.ed.ac.uk/tcohn/paraphrase_corpus.html.

A. What is inheritance in object oriented programming?
B. Explain the PageRank algorithm that is used by the Google search engine.
C. Explain the Vector Space Model that is used for Information Retrieval.
D. Explain Bayes Theorem from probability theory.
E. What is dynamic programming?

**Fig. 1** Five learning tasks used to create the corpus

not accurately represent these types of plagiarism. Attempting to create a resource that represents all of the possible types of plagiarism (see Sect. 2.1) would be a massive undertaking for which we do not have resources available. We also question how practical such a resource might be.

### 3.1 Learning tasks

A set of five short answer questions on a variety of topics that might be included in the Computer Science curriculum were created by the authors. Short answer questions were used since they provide an opportunity to show plagiarism, whilst minimising the burden placed on participants in this study; it was felt that we were unlikely to obtain good material if participants were required to do too much.

The five questions used in our study are shown in Fig. 1. This set of questions were chosen to represent a range of areas of Computer Science and also designed to be such that it was unlikely for any student to know the answer to all five questions. In addition, materials that are necessary for participants to answer these questions (see Sect. 3.2) could be easily obtained and provided to participants. The questions can essentially be answered by providing a short definition of the concept being asked about. Some of the questions allow for relatively open–ended answers, it would be possible to write quite long texts in answer to the question but could be adequately answered using a few hundred words.

### 3.2 Generation of answers

For each of these questions we aim to create a set of answers using a variety of approaches, some of which simulate cases in which the answer is plagiarised and others that simulate the case in which the answer is not plagiarised. To simulate plagiarism we require a source text in which the answer is found. For this we identified a suitable entry in Wikipedia[6] that contained an answer to the question. Wikipedia was chosen since it is readily available, generally accepted to provide information on a wide variety of topics, contains versions of pages in multiple languages (thus allowing evaluation of cross-lingual plagiarism detection) and contained answers to the type of questions used in our study.

We aimed to represent a variety of different degrees of rewrite in the plagiarised documents to enable the evaluation of different plagiarism detection algorithms. This is similar to proposals for levels of plagiarism in software code

[6] http://www.wikipedia.com.

(Faidhi and Robinson 1987), but for natural language texts. Keck (2006) discusses the following "levels" of rewrite: Near Copy, Minimal Revision, Moderate Revision, and Substantial Revision. These represent progressively more complex (and difficult) forms of rewrite identified from a set of plagiarised examples. Rewriting operations resulting from plagiarism may involve verbatim cut and paste, paraphrasing and summarising (Keck 2006).[7] Cut and paste involves lifting the original text with minor changes and is often easiest to detect. Paraphrases are alternative ways of conveying the same information (Barzilay and McKeown 2001), i.e. using different words (known as lexical paraphrases) or syntax (known as morpho-syntactic paraphrases). Campbell (1990) and Johns and Myers (1990) suggest that paraphrasing is one of a number of strategies (including summary and quotation) that students can use when integrating source texts into their writing. A summary is (typically) a shortened version of an original text. A summary should include all main ideas and important details, while reflecting the structure and order of the original. Editing operations typically used in producing summaries include (Jing and McKeown 1999): splitting up sentences from the original (sentence reduction), combining multiple sentences from the original (sentence combination), syntactic transformations (paraphrasing), lexical paraphrasing, the generalisation or the specification of concepts in the original text, and the reordering of sentences.

To generate our corpus, participants were asked to answer each question using one of four methods[8]:

**Near copy** Participants were asked to answer the question by simply copying text from the relevant Wikipedia article (i.e. performing cut-and-paste actions). No instructions were given about which parts of the article to copy (selection had to be performed to produce a short answer of the required length, 200–300 words).

**Light revision** Participants were asked to base their answer on text found in the Wikipedia article and were, once again, given no instructions about which parts of the article to copy. They were instructed that they could alter the text in some basic ways including substituting words and phrases with synonyms and altering the grammatical structure (i.e. paraphrasing). Participants were also instructed not to radically alter the order of information found in sentences.

**Heavy revision** Participants were once again asked to base their answer on the relevant Wikipedia article but were instructed to rephrase the text to generate an answer with the same meaning as the source text, but expressed using different words and structure. This could include splitting source sentences into one or more individual sentences, or combining more than one source sentence into a single sentence. No constraints were placed on how the text could be altered.

**Non-plagiarism** Participants were provided with learning materials in the form of either lecture notes or sections from textbooks that could be used to answer the relevauestion. Participants were asked to read these materials and then attempt to answer the question using their own knowledge (including what they had learned

[7] For further examples see http://www.chem.uky.edu/Courses/common/plagiarism.html and http://www.yale.edu/bass/writing/sources/plagiarism/.

[8] A pilot study with a limited number of participants used a finer grained distinction between types of plagiarism, however, we found it was difficult for participants to distinguish between them.

from the materials provided). They were also told that they could look at other materials to answer the question but explicitly instructed not to look at Wikipedia.

The aim of the final method (non-plagiarism) was to simulate the situation in which a student is taught a particular subject and their knowledge subsequently tested in some form of assessment. It is important to remember that just because a student has been taught a particular topic does not necessarily mean that they will be able to answer questions about it correctly and that one of the aims of assessment is to determine whether or not a student has mastered material they have been taught. One of our aims in including this scenario is to determine whether it is possible to distinguish between answers that are intentionally plagiarised and those where the student has attempted to understand the question before answering. A non-plagiarised answer also provides an indication of how much text one is likely to find in common between independently written texts.

### 3.3 Participation

A total of 19 participants were recruited to create texts for the corpus. Five of the participants were members of a team carrying out a group project on plagiarism detection while the remaining participants were either recruited by this group (through personal contact) or responded to an email request for volunteers. All participants were students in the Computer Science Department of Sheffield University and were studying for a degree in Computer Science at either undergraduate or postgraduate level. Participation was restricted to students with some familiarity of Computer Science since some familiarity with the topic would be required to answer the questions and also that this provided a more realistic plagiarism scenario.

Participants were presented with each of the five questions and asked to provide a single answer to each. Participants were instructed that answers should be between 200 and 300 words long and, to simplify later processing, should contain only standard (ASCII) characters and avoid using any symbols or computer code. For each question participants were instructed which approach to use to provide the answer. Two of the five questions were answered without plagiarising (the "non-plagiarism" category), one question using the near copy, one using light revision and one using heavy revision. The approach used for each question varied between participants to provide a variety of different answers to each question. To reduce learning and order effects, the tasks and categories used were arranged using a Graeco-Latin square arrangement (see, e.g. Kelly 2009, p. 54). An alternative methodology would have been to ask a single participant to provide multiple answers to each question, using a variety of approaches; however, this could have caused problems since the process of answering a question using one approach could influence subsequent answers.

All participants provided written consent to allow us to use their answers in order to make the corpus publicly-accessible. Participants were also asked to complete a short questionnaire after answering the questions. This recorded whether or not they were a native English speaker and, for each question, how familiar they were with the answer to the question being asked (1 = very familiar; 5 = not at all familiar) and how difficult they found answering the question (1 = very easy; 5 = very

**Table 1** Number of answers by learning task and plagiarism category

| Category | Learning task | | | | | Total |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | |
| Near copy | 4 | 3 | 3 | 4 | 5 | 19 |
| Light revision | 3 | 3 | 4 | 5 | 3 | 19 |
| Heavy revision | 3 | 4 | 5 | 4 | 3 | 19 |
| Non-plagiarised | 9 | 9 | 7 | 6 | 7 | 38 |
| Total | 19 | 19 | 19 | 19 | 19 | 95 |

difficult). Finally, participants were provided with a small reward for participation (electronic voucher for an on-line store).

## 4 Corpus analysis

### 4.1 Corpus properties

The corpus[9] contains 100 documents (95 answers provided by the 19 participants and the five Wikipedia source articles). For each learning task, there are 19 examples of each of the heavy revision, light revision and near copy levels and 38 non-plagiarised examples written independently from the Wikipedia source. Table 1 shows a breakdown of the number of answers in the corpus with respect to learning task (A–E) and plagiarism category. The uneven spread in the number of answers across tasks and categories results from using the Graeco-Latin square arrangement with 19 participants. The answer texts contain 19,559 words in total (22,230 unique tokens).[10] The average length of file in the corpus is 208 words (SD 64.91) and 113 unique tokens (SD 30.11). Overall, 59 (62%) of the files are written by native English speakers; the remaining 36 (38%) by non-native speakers.

### 4.2 Questionnaires

The questionnaires were analysed to determine whether there were differences between the types of participant. Figures 2 and 3 show the mean level of knowledge participants expressed as having and their level of difficulty with completing each task grouped by whether they are native or non-native speakers of English. Interestingly, the differences between the groups for knowledge (mean = 2.03 non-native; mean = 2.58 native) and difficulty (mean = 2.63 non-native; mean = 3.30 native) are statistically significant ($p < 0.01$, independent samples t-test). Overall, there is also a correlation between the scores for knowledge and difficulty ($r = 0.344$; $p < 0.01$) indicating that tasks for which participants had greater knowledge were considered easier.

[9] The corpus can be downloaded from: http://www.ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html.

[10] The Wikipedia pages total 14,242 words after conversion to plaintext using *lynx -dump* and removal of URL references.

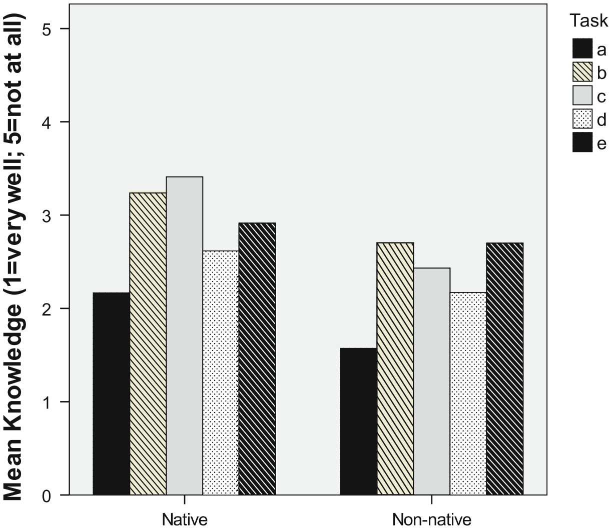


**Fig. 2** Mean level of knowledge across tasks between native and non–native participants (*1* very well, *5* not at all)

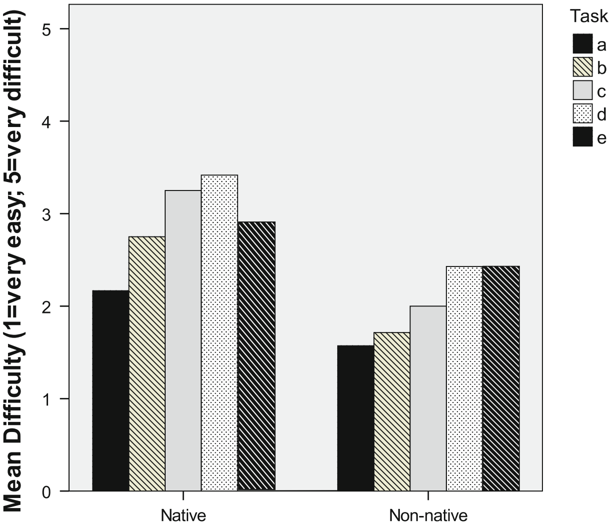


**Fig. 3** Mean level of difficulty across tasks between native and non–native participants (*1* very easy, *5* very difficult)

### 4.3 Observations

The corpus has some interesting features that are unlikely to be found in other resources. Unlike the majority of corpora that are used in language processing research, which comprise of carefully edited texts designed to reflect a specific type

**Example 1:**
Inheritance allowes classes to be categorized, similer to the way humans catagorize. It also provides a way to generalize du to the “is a” relationship between classes.

**Example 2:**
Generlisation also some time known as inheritance. The main reason behind this is a hierarchi st ructure of objects and classes. We can understand this mechanism by some examples: like fruit is aq main class and mangoes apple ,orange is child classs of the main class.So obviously inherit all the properties of fruit class.

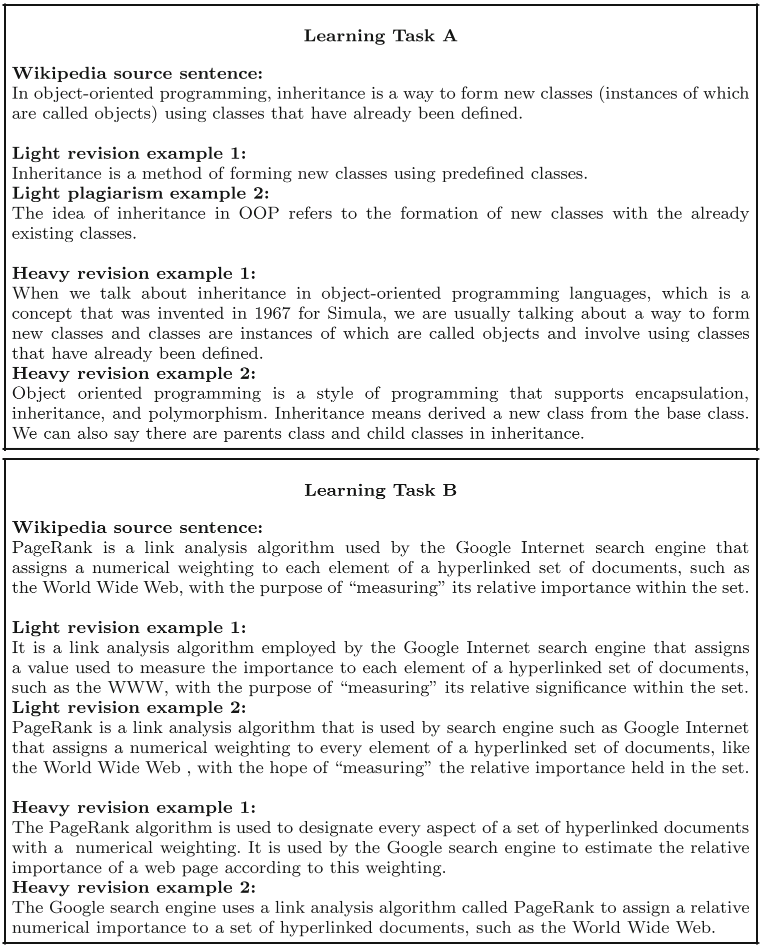
**Fig. 4** Examples of answers to learning task A containing errors

of document, our corpus includes documents that contain spelling, grammatical and typographical errors. Figure 4 shows extracts from answers provided for learning task A (see Fig. 1) containing such errors. (Note that the spacing in these examples is as provided by the participants.) These types of errors were more common in texts generated by participants who were not native speakers of English but also occurred in those generated by native speakers. It should be noted, however, these extracts represent two of the more extreme examples of errors found within texts. No attempt was made to clean up the texts to remove these errors from the texts since doing so would alter the material provided by the participants and these errors may actually complicate the task of plagiarism detection (by hampering string overlap approaches and making deeper analysis more difficult).

The simplest type of rewrite included in our study was cut and paste (near copy). Although this option did not require the participant to alter the text, they still had to decide which parts of the relevant Wikipedia article to use in their answer since the articles were longer than the 200–300 words requested. Participants used a variety of strategies including simply copying a single contiguous sequence of text of roughly the required length; others selected the portions of the text which most directly answered the relevant question. This could involve deleting isolated sentences or choosing sentences from throughout the article which are recombined into a coherent answer.

When participants were asked to perform light or heavy revision they employed similar strategies for selecting portions of the text from the Wikipedia source. Figure 5 shows examples of light and heavily revised sentences and the corresponding sentence in the Wikipedia source for learning tasks A and B. In the examples of light revision the connection between the source and plagiarised text is generally obvious (at least to the human). A number of techniques were used to obscure the connection with the source text. The first example of a lightly revised response to learning task A demonstrates deletion (the phrases “In object-oriented programming,” and “(instances of which are called objects)” are removed), substitution of words with synoymns (“way” becomes “method”) and simple paraphrases (“to form” becomes “of forming” and “classes that have already been defined” becomes “predefined classes”).

A common strategy in the examples of heavy revision is to obscure the link to the source text further by altering the amount of information contained in each sentence, either to include something from an additional sentence or to break single sentence into two separate sentences. For example, in Fig. 5, in the first example of plagiarism for learning task A information from the sentence immediately following



**Learning Task A**

**Wikipedia source sentence:**
In object-oriented programming, inheritance is a way to form new classes (instances of which are called objects) using classes that have already been defined.

**Light revision example 1:**
Inheritance is a method of forming new classes using predefined classes.
**Light plagiarism example 2:**
The idea of inheritance in OOP refers to the formation of new classes with the already existing classes.

**Heavy revision example 1:**
When we talk about inheritance in object-oriented programming languages, which is a concept that was invented in 1967 for Simula, we are usually talking about a way to form new classes and classes are instances of which are called objects and involve using classes that have already been defined.
**Heavy revision example 2:**
Object oriented programming is a style of programming that supports encapsulation, inheritance, and polymorphism. Inheritance means derived a new class from the base class. We can also say there are parents class and child classes in inheritance.

**Learning Task B**

**Wikipedia source sentence:**
PageRank is a link analysis algorithm used by the Google Internet search engine that assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set.

**Light revision example 1:**
It is a link analysis algorithm employed by the Google Internet search engine that assigns a value used to measure the importance to each element of a hyperlinked set of documents, such as the WWW, with the purpose of "measuring" its relative significance within the set.
**Light revision example 2:**
PageRank is a link analysis algorithm that is used by search engine such as Google Internet that assigns a numerical weighting to every element of a hyperlinked set of documents, like the World Wide Web , with the hope of "measuring" the relative importance held in the set.

**Heavy revision example 1:**
The PageRank algorithm is used to designate every aspect of a set of hyperlinked documents with a numerical weighting. It is used by the Google search engine to estimate the relative importance of a web page according to this weighting.
**Heavy revision example 2:**
The Google search engine uses a link analysis algorithm called PageRank to assign a relative numerical importance to a set of hyperlinked documents, such as the World Wide Web.

**Fig. 5** Examples of light and heavily revised sentences in answers to learning tasks A and B

the source (that the concept of inheritance was invented in 1967 for the Simula language) is inserted in the middle of the rewritten sentence. The second example includes information from various parts of the source article. The first example of a heavily revised answer to learning task B is an example where the source sentence has been split into two.

However, the distinction between the amounts of rewriting involved in the heavily and lightly revised examples is not clear with the amount of rewrite generally depending on the individual participant.

### 4.4 Computing similarity

Simple approaches have proven to be robust baseline methods for plagiarism detection (Wise 1992; Brin et al. 1995; Gitchell and Tran 1999; Lyon et al. 2001).

We apply two such methods, *n*-gram overlap and longest common subsequence, to our corpus to determine whether they can distinguish between the various levels of plagiarism or whether a text is created using the non-plagiarism or one of the plagiarism approaches.

### *4.4.1 N-gram overlap*

The similarity between a pair of documents can be computed by counting the number of *n*-grams they have in common and this approach is commonly used for plagiarism and copy detection (Brin et al. 1995; Shivakumar and Garcia-Molina 1996; Lyon et al. 2001). Typically set-theoretic association scores are also utilised to measure the amount of overlap between pairs of documents. For example, Broder (1998) uses the containment measure. Given an *n*-gram of length *n*, $S(A, n)$, the set of *n*-grams for document *A*, and $S(B, n)$, the set of *n*-grams for document *B*, the containment between *A* and *B*, $c_n(A, B)$ is defined following Eq. 1. Informally, containment measures the number of unique *n*-grams in *A* that are also in *B*. The score ranges between 0 and 1, with 0 indicating that none of the answer is shared with the Wikipedia source and 1 that it is completely shared. The containment measure is suitable for our evaluation since the source texts are longer than the short answers.

$$c_n(A,B) = \frac{\mid S(A,n) \cap S(B,n) \mid}{\mid S(A,n) \mid} \tag{1}$$

We compare *n*-gram sets of lengths 1–5 and use the containment measure to indicate the degree of similarity between each answer and the answer text. Before computing the containment measure the text is pre-processed by converting all letters to lowercase and comparing only unique *n*-grams.

### *4.4.2 Longest common subsequence*

Another simple approach to plagiarism and reuse detection is to compute the number of simple edit operations (insertions and deletions) required to transform one text into the other. The longest common subsequence *lcs* between two strings is the sequence of common elements such that no longer string is available. (For identical strings, the lcs is the length of the shorter string.) This can be computed using a dynamic programming solution to finding the maximum cost of transforming *a* into *b* using only insertion and deletions. Due to the quadratic time complexity of using dynamic programming, approximate solutions have been found such as the $O(nd)$ algorithm (where *n* is the sum of two strings *a* and *b*, and *d* the size of the minimum edit script to change *a* into *b*) as suggested by Myers (1986). The *lcs* measure is often normalised by computing the *lcs* between two texts and then dividing by the length of the answer text and this is referred to as $lcs_{\mathrm{norm}}$. Before computing the value of $lcs_{\mathrm{norm}}$, all letters are converted to lowercase.

The length of $lcs_{\mathrm{norm}}$ indicatives re-ordering due to paraphrasing or changes in the structure of the narrative and substitution of equivalent expressions.

**Table 2** Mean similarity between answer texts and unrelated Wikipedia article

| Task | $c_w(A, B)$ for $w$-gram | | | | | $lcs_{\text{norm}}$ |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| A | 0.48 | 0.15 | 0.08 | 0.05 | 0.03 | 0.26 |
| B | 0.65 | 0.23 | 0.12 | 0.08 | 0.05 | 0.35 |
| C | 0.49 | 0.20 | 0.11 | 0.06 | 0.03 | 0.29 |
| D | 0.60 | 0.29 | 0.17 | 0.10 | 0.06 | 0.35 |
| E | 0.61 | 0.23 | 0.13 | 0.08 | 0.05 | 0.34 |
| Avg. | 0.57 | 0.22 | 0.12 | 0.07 | 0.04 | 0.32 |

Figures are averaged across all plagiarism types for each learning task

### *4.4.3 Similarity between unrelated texts*

We begin by establishing a baseline score for these similarity measures. This is necessary since it has been shown that the vocabulary of independently written texts can overlap by as much as 50% (Finlay 1999). Each answer text is compared against the source articles for the other learning tasks (e.g. the Wikipedia article used for the task on dynamic programming is compared with all answers which are *not* related to this topic). Stopwords are not removed from the documents before applying the similarity measures.[11] Results are averaged across all answer texts for a particular learning task and shown in Table 2. The resulting matches for $c_1(A, B)$ indicates that unrelated texts share a reasonable number of common words. However, as $n$ increases the overlap between unrelated texts decreases rapidly to a point where few matches are found when $n > 3$.

### *4.4.4 Comparison of rewrite levels*

The next experiment establishes the similarity between the various levels of plagiarism and the relevant WIkipedia original (e.g. we compare the Wikipedia article about dynamic programming against all answers for this task). Table 3 shows the results averaged across the five learning tasks.

The difference in results for each category indicate that the texts in the corpus did contain varying levels of text reuse. Differences between each rewrite category are all significant (one-way ANOVA (Morgan et al. 2001) with Bonferroni post-hoc test, $p < 0.01$). As expected, the degree of similarity between the texts is lower as the level of rewriting increases (from near copy to heavy revision). The scores for the non-plagiarised answers are noticeably closer to the average baseline scores (see Table 2) than for those generated using the various plagiarism strategies. As the length of $n$ increases, the decrease in similarity is more pronounced for the heavily revised and non-plagiarised answers. This indicates that the authors are breaking up

[11] Equivalent experiments were carried out in which the stopwords were removed before computing similarity. We found a similar pattern of results to those reported here and do not report results when stopwords are removed for brevity.

**Table 3** Mean similarity between Wikipedia and answer texts for each rewrite level across all tasks

| Category | $c_n(A, B)$ for $n$-gram | | | | | $lcs_{\text{norm}}$ |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| Near copy | 0.95 | 0.89 | 0.85 | 0.81 | 0.78 | 0.88 |
| Light revision | 0.87 | 0.70 | 0.56 | 0.46 | 0.39 | 0.76 |
| Heavy revision | 0.81 | 0.52 | 0.34 | 0.26 | 0.21 | 0.58 |
| Non-plagiarised | 0.63 | 0.23 | 0.05 | 0.01 | 0.00 | 0.41 |

the longer sequences of words when the text is heavily revised. The $lcs_{\text{norm}}$ measure also indicates that the degree of ordering between the texts decreases as authors heavily revise the original version.

### 4.4.5 Comparison of learning tasks

Table 4 shows the differences in similarity scores for each of the learning tasks across all rewrite categories. The results show variation between the different tasks (the majority of which are not significant) and highlights the importance of using multiple learning tasks when developing a corpus containing examples of plagiarism.

### 4.4.6 Comparison of native and non-native speakers

Analysis of results based on the participant's mother tongue showed, overall, no significant differences between the similarity scores for any of the plagiarism levels. However, we did observe that for $n$-grams with $n \geq 3$, the containment scores were lower for the heavy revision category indicating that perhaps unfamiliarity with the language meant students were relying more heavily on the source text and carrying out less revision than native speakers. This is shown most clearly for 5-gram containment scores (Fig. 6). This is consistent with previous results (Keck 2006) which showed that non-native speakers are more likely than native speakers to use cut-and-paste as a strategy when reusing text.

**Table 4** Average similarity between Wikipedia and answer texts for each task across all rewrite categories

| Task | $c_n(A, B)$ for $n$-gram | | | | | $lcs_{\text{norm}}$ |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| A | 0.77 | 0.45 | 0.31 | 0.27 | 0.25 | 0.55 |
| B | 0.81 | 0.53 | 0.35 | 0.28 | 0.25 | 0.63 |
| C | 0.71 | 0.44 | 0.31 | 0.25 | 0.21 | 0.53 |
| D | 0.82 | 0.58 | 0.46 | 0.40 | 0.36 | 0.69 |
| E | 0.81 | 0.56 | 0.41 | 0.35 | 0.32 | 0.65 |
| Avg. | 0.79 | 0.51 | 0.37 | 0.31 | 0.28 | 0.61 |

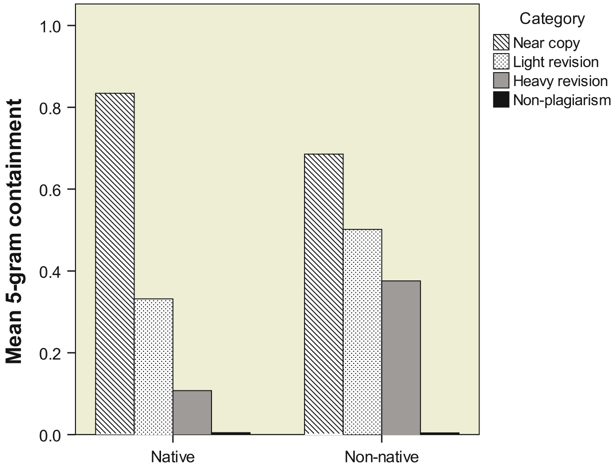


**Fig. 6** Mean 5-gram containment scores across rewrite categories for native and non-native participants

## 5 Classification task

To demonstrate how the corpus could be used to evaluate plagiarism detection systems we cast the problem as a supervised document classification task (similar to the extrinsic plagiarism detection problem outlined in Sect. 2.1). The two lexical overlap measures described in Sects. 4.4.1 and 4.4.2 were used as features. The Wikipedia source articles were excluded from the corpus and threefold cross-validation carried out over the remainder of the documents. A simple Naive Bayes probabilistic supervised learning algorithm[12] was used for classification. Classification effectiveness is measured using the $F_1$ measure (the harmonic mean of precision and recall given equal weighting) computed for each class, averaged across the three runs from cross-validation.

Results are shown in Table 5. Overall we observe that the most successful classification is for the non-plagiarised class, followed by near copy with results decreasing as the level of rewrite increases. The individual features giving highest accuracy, including $c_2(A, B)$ and $c_3(A, B)$, are consistent with previous findings (Lyon et al. 2001).

The best performance (80% accuracy) is obtained when all features are combined. The confusion matrix for classification using this set of features (Table 6) demonstrates that mis-classification occurs mainly between the light and heavy categories, indicating perhaps these could be folded into a single rewrite category.

In practice we are more likely to be interested in whether a particular answer is plagiarised or not than in labeling a text with the amount of rewriting that has taken place. A simple plagiarism detection task was created by combining all three categories of plagiarism into a single category and then carrying out a binary

[12] The WEKA 3.2 implementation was used.

**Table 5** Results ($F_1$ measure) for a supervised classification using various features

| Class | Feature | | | | | | |
|---|---|---|---|---|---|---|---|
| | $c_1(A, B)$ | $c_2(A, B)$ | $c_3(A, B)$ | $c_4(A, B)$ | $c_5(A, B)$ | $lcs_{\mathrm{norm}}$ | All |
| Near copy | 0.778 | 0.778 | 0.850 | 0.850 | 0.829 | 0.571 | 0.850 |
| Light revision | 0.605 | 0.579 | 0.571 | 0.452 | 0.357 | 0.400 | 0.629 |
| Heavy revision | 0.457 | 0.485 | 0.500 | 0.500 | 0.537 | 0.556 | 0.611 |
| Non-plagiarised | 0.895 | 0.937 | 0.911 | 0.902 | 0.925 | 0.911 | 0.937 |
| Overall accuracy (%) | 72.6 | 76.8 | 75.8 | 73.7 | 73.7 | 67.4 | 80.0 |

**Table 6** Confusion matrix for classification using all measures as features

| Classified as | Near copy | Heavy revision | Light revision | Non-plagiarised |
|---|---|---|---|---|
| Near copy | 17 | 0 | 1 | 1 |
| Light revision | 3 | 5 | 11 | 0 |
| Heavy revision | 1 | 11 | 4 | 3 |
| Non-plagiarism | 0 | 1 | 0 | 37 |

classification task using all measures as features. It was found that 94.7% of the answers were correctly classified. This figure is surprisingly high and highlights the fact that in practice even simple measures can successfully identify plagiarised examples.

## 6 Summary and future work

In this paper we have discussed the creation of a publicly-available resource designed to assist in the evaluation of plagiarism detection systems for natural language texts. Our aim was to generate a resource that represented the strategies used by students when reusing text as far as is possible. Rather than relying on automatic methods for generating plagiarised texts our resource consists of examples manually generated by students at our institution. These participants were asked to produce short answers to five questions on a range of topics in Computer Science using a variety of methods that were designed to simulate plagiarised and non-plagiarsed responses. The importance of generating realistic examples has been highlighted through a qualitative analysis of plagiarised texts where aspects such as language skills have demonstrated that the examples may contain a range of grammatical, typographical and spelling errors. Analysis of the corpus using two simple text reuse methods (*n*-gram overlap and longest common subsequence) identified clear distinctions between the answers generated for each level of plagiarism. Interestingly these simple methods can distinguish between answers generated using methods that simulate plagiarism and non-plagiarism with an accuracy of almost 95%.

Although our resource may be a useful resource in the evaluation of plagiarism detection systems it is limited in a number of ways. The manual nature of the corpus creation process has restricted the size of the corpus. Ideally we would like to be able to include further examples of short answer questions and involve more participants. In addition, the length of examples is short, compared to texts such as essays and this may limit the range of approaches that could realistically be tested using our resource. Finally, our corpus only contains examples of answers to Computer Science questions. We aim to address these limitations by collecting further examples and experiment with soliciting longer answers. In addition, we hope to develop sets of learning tasks for other academic disciplines and gather answers for these. We also hope to develop evaluation resources that represent further types of plagiarism including cases where plagiarised passages are embedded within otherwise acceptable answers and using non-English versions of the Wikipedia articles to simulate multilingual plagiarism.

## 7 Data

The corpus described in this paper is freely available for research purposes and can be downloaded from http://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html.

**Acknowledgments** We thank James Gregory, Aman Brar, Chris Bishop, Saleh Al Belwi and Congyun Long for organising the data collection and all participants involved in generating examples for the corpus.

## References

Barzilay, R., & McKeown, K. (2001). Extracting paraphrases from a parallel corpus. In *Proceedings of 39th annual meeting of the association for computational linguistics, association for computational linguistics* (pp. 50–57). Toulouse, France.

Brin, S., Davis, J., & Garcia-Molina, H. (1995). Copy detection mechanisms for digital documents. In *Proceedings of the ACM SIGMOD international conference on management of data* (pp. 398–409).

Broder, A. Z. (1998). On the resemblance and containment of documents. In *Compression and complexity of sequences*. IEEE Computer Society.

Bull, J., Collins, C., Coughlin, E., & Sharp, D. (2001). Technical review of plagiarism detection software report. Luton: Computer Assisted Assessment Centre. http://www.plagiarismadvice.org/documents/resources/Luton_TechnicalReviewofPDS.pdf.

Callison-Burch, C., Koehn, P., & Osborne, M. (2006). Improved statistical machine translation using paraphrases. In *Proceedings of the human language technology conference of the NAACL, main conference, association for computational linguistics* (pp. 17–24). New York City, USA.

Campbell, C. (1990). Writing with other's words: Using background reading text in academic compositions. In B. Kroll (Ed.), *Second language writing: Research insights for the classroom* (pp. 211–230). Cambridge: Cambridge University Press.

Cebrián, M., Alfonseca, M., & Ortega, A. (2007). Automatic generation of benchmarks for plagiarism detection tools using grammatical evolution. In *GECCO '07: Proceedings of the 9th annual conference on genetic and evolutionary computation* (pp. 2253–2253).

Clough, P. (2000). Plagiarism in natural and programming languages: An overview of current tools and technologies. Technical report on research memoranda: CS-00-05. Department of Computer Science, University of Sheffield (UK).

Clough, P., Gaizauskas, R., Piao, S., & Wilks, Y. (2002). Measuring text reuse. In *Proceedings of 40th annual meeting of the association for computational linguistics, association for computational linguistics* (pp. 152–159). Philadelphia, Pennsylvania, USA.

Cohn, T., Callison-Burch, C., & Lapata, M. (2008). Constructing corpora for development and evaluation of paraphrase systems. *Computational Lingustics, 34*(4), 597–614.

Coleman, M., & Liau, T. (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology, 60*, 283–284.

Collberg, C., & Kobourov, S. (2005). Self-plagiarism in computer science. *Communications of the ACM, 48*(4), 88–94.

Culwin, F., & Lancaster, T. (2001). Plagiarism issues for higher education. *VINE, 31*(2), 36–41.

Dolan, B., Quirk, C., & Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of 20th internatonal conference on computational linguistics (Coling 2004)* (pp. 350–356) Geneva, Switzerland.

Faidhi, J. A. W., & Robinson, S. K. (1987). An empirical approach for detecting program similarity and plagiarism within a university programming environment. *Journal of Computer Education, 11*(1), 11–19.

Finlay, S. (1999). Copycatch. Master's thesis, University of Birmingham.

Flesch, R. (1974). *The art of readable writing*. New York: Harper and Row.

Gitchell, D., & Tran, N. (1999). Sim: A utility for detecting similarity in computer programs. In *Proceedings of 13th SIGSCI technical symposium on computer science education* (pp. 226–270).

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 16th international conference on computational linguistics (COLING-96)* (pp. 466–470), Copenhagen, Denmark.

Guthrie, D., Guthrie, L., Allison, B., & Wilks, Y. (2007). Unsupervised anomaly detection. In *Proceedings of the twentieth international joint conference on artificial intelligence (IJCAI 2007)* (pp. 1626–1628). Hyderabad, India.

Heckel, P. (1978). A technique for isolating differences between files. *Communications of the ACM, 21*(4), 264–268.

Hislop, G. W. (1998). Analyzing existing software for software re-use. *Journal of Systems and Software, 54*(3), 203–215.

Jing, H., & McKeown, K. (1999). The decomposition of human-written summary sentences. In *Proceedings of SIGIR99* (pp. 129–136).

Johns, A., & Myers, P. (1990). An analysis of summary protocols of university esl students. *Applied Linguistics, 11*, 253–271.

Joy, M., & Luck, M. (1999). Plagiarism in programming assignments. *IEEE Transactions of Education, 42*(2), 129–133.

Judge, G. (2008). Plagiarism: Bringing economics and education together (with a little help from it). *Computers in Higher Education Economics Review, 20*(1), 21–26.

Juola, P. (2006). Authorship attribution. *Foundations and Trends in Information Retrieval, 1*(3), 233–334.

Kang, N., Gelbukh, A., & Han, S. (2006). Ppchecker: Plagiarism pattern checker in document copy detection. In *Proceedings of the 2006 European conference on information retrieval* (pp. 565–569).

Keck, C. (2006). The use of paraphrase in summary writing: A comparison of l1 and l2 writers. *Journal of Second Language Writing, 15*, 261–278.

Kelly, D. (2009). Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval, 3*(1–2), 1–224.

Korfhage, R. (1997). *Information storage and retrieval*. London: Wiley.

Lyon, C., Malcolm, J., & Dickerson, B. (2001) Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 conference on empirical methods in natural language processing (EMNLP-2001)* (pp. 118–125).

Manber, U. (1994). Finding similar files in a large file system. In *Proceedings of 1994 winter usenix technical conference* (pp. 1–10).

Martin, B. (1994). Plagiarism: A misplaced emphasis. *Journal of Information Ethics, 3*(2), 36–47.

McCabe, D. (2005). Research report of the center for academic integrity. http://www.academicintegrity.org.

McEnery, A. M., & Oakes, M. P. (2000). Authorship identification and Computational Stylometry. In R. Dale, H. Moisl, & H. Somers (Eds.), *Handbook of natural language processing* (pp. 545–562). New York.

Mihalcea, R., Chklovski, T., & Kilgarriff, A. (2004). The senseval-3 English lexical sample task. In *Proceedings of senseval-3: The third international workshop on the evaluation of systems for the semantic analysis of text*. Barcelona, Spain.

Morgan, G., Griego, O., & Gloeckner, G. (2001). *SPSS for windows: An introduction to use and interpretation in research*. Mahwah New Jersey: Lawrence Erlbaum Associates.

Myers, E. (1986). An O(ND) difference algorithm and its variations. *Algorithmica, 1*(2), 251–266.

Pang, B., Knight, K., & Marcu, D. (2003). Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the human language technology conference and the annual meeting of the North American chapter of the association for computational linguistics*.

Pinto, D., Civera, J., Barrón-Cedeño, A., Juan, A., & Rosso, P. (2009). A statistical approach to crosslingual natural language tasks. *Journal of Algorithms, 64*(1), 51–60.

Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., & Rosso, P. (2009). Overview of the 1st international competition on plagiarism detection. In B. Stein, P. Rosso, E. Stamatatos, M. Koppel, & E. Agirre (Eds.), *Workshop on uncovering plagiarism, authorship, and social software misuse (PAN 09)* (pp. 1–9), CEUR-WS.org.

Sanderson, M. (1997). Duplicate detection in the reuters collection. Technical Report (TR-1997-5), Department of Computing Science at the University of Glasgow.

Shivakumar, N., & Garcia-Molina, H. (1996). Building a scalable and accurate copy detection mechanism. In *Proceedings of 1st ACM conference on digital libraries DL'96*.

Stein, S. B., & zu Eissen S. M. (2006). Near similarity search and plagiarism analysis. In *Proceedings of the 29th annual conference of the GfKl*.

Voorhees, E., & Harman, D. (2005). *TREC: Experiment and evaluation in information retrieval*. Cambridge, Mass: MIT Press.

Web Technology & Information Systems Group BUW. (2008). Plagiarism Corpus Webis-PC-08. In S. M. zu Eissen, B. Stein, & M. Kulig (Eds.). http://www.uni-weimar.de/medien/webis/research/corpora.

White, D., & Joy, M. (2004). Sentence-based natural language plagiarism detection. *ACM Journal on Educational Resources in Computing, 4*(4), 1–20.

Wise, M. (1992). Detection of similarities in student programs: Yap'ing may be preferable to plague'ing. In *Presented at 23rd SIGCSE technical symposium* (pp. 268–271). Kansas City, USA.

Woolls, D., & Coulthard, M. (1998). Tools for the trade. *Forensic Linguistics, 5*(1), 33–57.

Zobel, J. (2004). Uni cheats racket: A case study in plagiarism investigation. In: *ACE '04: Proceedings of the sixth conference on Australasian computing education* (pp. 357–365). Darlinghurst, Australia: Australian Computer Society, Inc.

zu Eissen, S. M., & Stein, B. (2006). Intrinsic plagiarism detection. In *Proceedings of the ninth international conference on text, speech and dialogue* (pp. 661–667).

zu Eissen, S. M., Stein, B., & Kulig, M. (2007). Plagiarism detection without reference collections. In R. Decker & H. Lenz (Eds.), *Advances in data analysis* (pp. 359–366). London: Springer.