Paraphrase Acquisition via Crowdsourcing and Machine Learning

STEVEN BURROWS
and
MARTIN POTTHAST
and
BENNO STEIN

Web Technology and Information Systems, Bauhaus-Universität Weimar [*]

To paraphrase means to rewrite content whilst preserving the original meaning. Paraphrasing is important in fields such as text reuse in journalism, anonymising work, and improving the quality of customer-written reviews. This paper contributes to paraphrase acquisition and focuses on two aspects that are not addressed by current research: (1) acquisition via crowdsourcing, and (2) acquisition of passage-level samples. The challenge of the first aspect is automatic quality assurance; without such a means the crowdsourcing paradigm is not effective, and without crowdsourcing the creation of test corpora is unacceptably expensive for realistic order of magnitudes. The second aspect addresses the deficit that most of the previous work in generating and evaluating paraphrases has been conducted using sentence-level paraphrases or shorter; these short-sample analyses are limited in terms of application to plagiarism detection, for example. We present the Webis Crowd Paraphrase Corpus 2011 (Webis-CPC-11), which recently formed part of the PAN 2010 international plagiarism detection competition. This corpus comprises passage-level paraphrases with 4 067 positive samples and 3 792 negative samples that failed our criteria, using Amazon's Mechanical Turk for crowdsourcing. In this paper, we review the lessons learned at PAN 2010, and explain in detail the method used to construct the corpus. The empirical contributions include machine learning experiments to explore if passage-level paraphrases can be identified in a two-class classification problem using paraphrase similarity features, and we find that a k-nearest-neighbor classifier can correctly distinguish between paraphrased and non-paraphrased samples with 0.980 precision at 0.523 recall. This result implies that just under half of our samples must be discarded (remaining 0.477 fraction), but our cost-analysis shows that the automation we introduce results in a 18% financial saving and over 100 hours of time returned to the researchers when repeating a similar corpus design. On the other hand, when building an unrelated corpus requiring say 25% training data for the automated component, we show that the financial outcome is cost-neutral, whilst still returning over 70 hours of time to the researchers. The work presented here is the first to join the paraphrasing and plagiarism communities.

Categories and Subject Descriptors: H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*computer-supported cooperative work*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*language generation*

General Terms: human factors, experimentation

Additional Key Words and Phrases: paraphrase generation, plagiarism, mechanical turk, corpus, cost analysis

[*] E-Mail: <first name>.<last name>@uni-weimar.de

1. INTRODUCTION

The recognition and generation of paraphrases forms the heart of numerous analysis and synthesis tasks in information retrieval, information extraction, and natural language processing. Given two texts, t_1 and t_2, then t_1 is a paraphrase of t_2 if a fact ϕ that can be inferred from t_1 can also be inferred from t_2, and vice versa:

$$(t_1 \wedge \alpha \models \phi) \quad \Leftrightarrow \quad (t_2 \wedge \alpha \models \phi) \tag{1}$$

The symbol α refers to a world (a domain theory or background knowledge) in the form of a set of relations that readers of t_1 and t_2 agree upon. Paraphrase *recognition* means to analyze whether two texts t_1 and t_2 fulfill Equivalence (1); paraphrase *generation* means to construct a text t_2 given a text t_1 such that Equivalence (1) is fulfilled; paraphrase *acquisition* refers to the task of compiling a set of paraphrase pairs. The difficulty of recognizing paraphrases results from the complexity of semantic entailment [Androutsopoulos and Malakasiotis 2010], as well as from the fact that it is impossible to state α completely.[1] This argumentation also applies to paraphrase generation and paraphrase acquisition.

Typically, t_1 and t_2 are of similar length, and most of the existing research is in fact on sentence-level paraphrasing. Purely syntactical approaches to paraphrase recognition (and generation as well) disregard semantic relationships and compare t_1 and t_2 using surface-level features. If semantics are brought in, they are often in the form of α being a thesaurus or a synonym dictionary.

Humans outperform machines in paraphrasing and thus finally decide whether t_1 and t_2 stand in a paraphrase relation. This fact explains the utmost importance of well-formed paraphrase corpora, since they are necessary to evaluate and benchmark the progress of research groups working on the foundations of paraphrasing, on new algorithms, and on new tools. But, as pointed out by Dolan and Brockett [2005], the compilation of reference corpora is more difficult than in other fields:

> *"Since paraphrase is not apparently a common 'natural' task—under normal circumstances people do not attempt to create extended paraphrase texts—the field lacks a large readily identifiable dataset comparable to, for example, the Canadian Hansard corpus in SMT that can serve as a standard against which algorithms can be trained and evaluated."*

In our research, we investigated whether crowdsourcing can be exploited in order to compile a reference corpus of realistic order of magnitude. Our undertaking was not of purely theoretical interest, but driven by the PAN 2010 International Plagiarism Detection Competition [Potthast et al. 2010a] for which an extensive and high-quality corpus had to be developed.

Plagiarism detection is a paraphrase recognition task that has received much recent attention. Plagiarism refers to the use of another author's information, language, or writing, when done without proper acknowledgment of the original source. The detection of plagiarism refers to the task of unveiling such cases. Note in this regard that most paraphrase recognition tasks are of "external" nature: one is given a text t_1 along with a corpus D wherein a paraphrase t_2 of t_1 is to be found. For intrinsic plagiarism detection tasks even

[1] Let α and α' be two relevant worlds for a task in question, then t_1 and t_2 might be paraphrases under α but not under α'. Moreover, without loss of generality we refer to symmetrical paraphrases throughout the paper, which can be considered as the more common case compared to asymmetrical paraphrases [Cordeiro et al. 2007a].

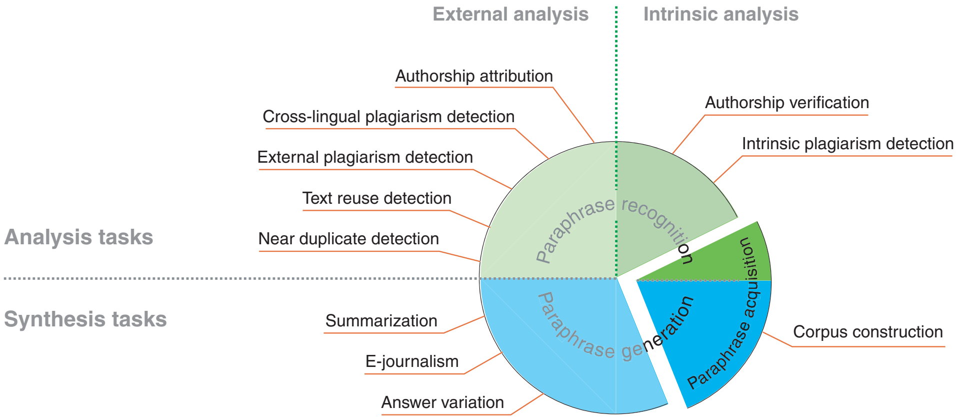

Fig. 1. Overview of text analysis tasks (upper half) and text synthesis tasks (lower half) and their relationship to paraphrasing. The analysis tasks are subdivided with regard to external (left) and internal tasks (right) and arranged by increasing difficulty: from similarity analysis over discrimination analysis up to one-class classification. Corpus construction tasks are of synthetic nature but can involve analysis aspects too.

t_1 is not given, but must be identified due its different writing style. A particularity of plagiarism detection is that t_1 and t_2 are assumed to be of passage length or section length.

Figure 1 organizes a broad spectrum of relevant text analysis and text synthesis tasks that deal with paraphrase recognition or generation. Paraphrase acquisition, which is the focus of our paper, combines aspects from both.

1.1 Outline and Contributions

This paper discusses the role of paraphrasing in plagiarism detection (Section 2), outlines the construction of a paraphrase corpus via crowdsourcing (Section 3), and reports on the parts that can be automated and the associated cost trade-offs (Section 4). Its main contributions can be summarized as follows:

(1) We identify plagiarism detection as a relevant paraphrase recognition task that, in particular, deals with passage-level text reuse.

(2) We construct a paraphrase corpus with 7 859 positive and negative samples, paraphrased by human editors via a crowdsourcing platform. The corpus is unique with respect to its size, its sample lengths, and its construction principle.

(3) We tackle the problem of automatic quality assurance and turn crowdsourcing into an effective means for paraphrase acquisition. Our paraphrase recognition work treats the aspect that negative samples are not only rejected because the content is unrelated, but also because it is too closely related.

(4) Our classification model combines the best state-of-the-art features and different machine learning paradigms. An important finding (which we will explain) is the fact that the k-nearest neighbor classifier can achieve 0.98 precision at around 0.50 recall.

(5) We give a comprehensive cost and time analysis showing excellent savings that can be made.

We consider our research also as a gap-bridging piece of work between different communities, as it shows common ground between paraphrasing and plagiarism research.

1.2 Related Work

Existing Corpora. The MSRPC corpus [Dolan and Brockett 2005], the KMC corpus [Knight and Marcu 2002], an unnamed corpus by Cohn et al. [2008], and the METER corpus [Clough et al. 2002] have all featured prominently in the paraphrasing community and closely related areas. The first three corpora are all comprised of sentence-level paraphrases [Madnani and Dorr 2010]. The MSRPC[2] (Microsoft Research Paraphrase Corpus) contains 5 801 sentence pairs, of which 3 900 are positives and 1 901 are negatives. The corpus was developed using automated extraction techniques of probable sentence-level paraphrases from news articles [Dolan and Brockett 2005], and candidate paraphrases were assessed by two human judges. The KMC[3] (Knight and Marcu Corpus) corpus contains 1 067 sentence pairs, of which all are positives. This design has motivated others to supplement the corpus with negative samples [Fernando 2007]. This corpus was again developed with automated extraction techniques on news articles, and the candidate paraphrases were identified from the articles and their corresponding abstract summaries. We note that this corpus is an application of entailed sentences, therefore it is less relevant to our work, but we still mention it for completeness. The unnamed corpus by Cohn et al. [2008][4] contains 900 sentence pairs evenly taken from the MSRPC corpus, Chinese journalism stories, and a novel. Several automated and manual approaches are used to annotate the pairs as either “sure” or “possible”, but it is unclear which set should be taken as the gold standard. Finally, the METER[5] (Measuring Text Reuse) corpus [Clough et al. 2002] for journalistic text reuse has “press agency” and “newspaper” copies of the same news stories. It contains 1 717 texts judged by one professional journalist, with annotations about the level of derivation and reuse between samples. In summary, three of the corpora are limited to sentence-level paraphrases, and the last corpus was for related work dealing with larger samples in journalistic text reuse.

Crowdsourcing. Brabham [2008a] defines crowdsourcing as an “online, distributed problem-solving and production model”. It is essentially a form of human computation [Quinn and Bederson 2011]. A benefit of this distributed model is that the work can be shared amongst a wide variety of demographics, skill sets, and political investments [Brabham 2008b], where such diversity would be difficult to obtain otherwise. A disadvantage is that with such anonymity, there are some that will exploit the model, and work is always needed to review submissions to separate the legitimate work from the rest. However, there are many genuine workers out there, which are motivated by reasons such as money, personal development, or credibility gained amongst peers. Previous research has demonstrated the successful application of crowdsourcing in a variety of areas including user studies [Kittur et al. 2008], relevance assessment [Alonso and Mizzaro 2009], machine translation [Ambati et al. 2010; Callison-Burch 2009], ideas competitions [Leimeister et al. 2009], and annotating speech and text [Callison-Burch and Dredze 2010; Snow et al. 2008]. Amazon’s Mechanical Turk [Barr and Cabrera 2006] is one such crowdsourcing service that has gaining much popular attention, which also forms the crowdsourcing service used in this paper.

[2] `http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042`

[3] This corpus should be requested from the authors directly. See Knight and Marcu [2002] for examples.

[4] `http://staffwww.dcs.shef.ac.uk/people/T.Cohn/paraphrase_corpus.html`

[5] `http://nlp.shef.ac.uk/meter`

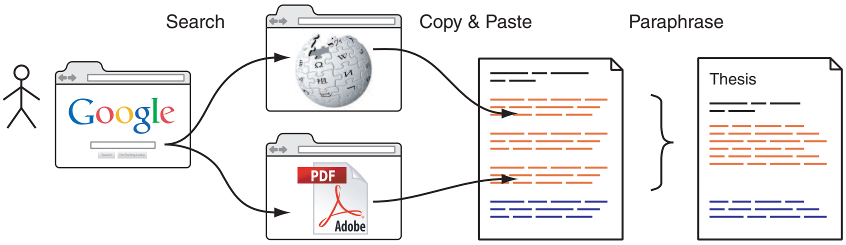

Fig. 2. Overview of the activities involved in plagiarizing.

Crowd Paraphrase Acquisition. Recently, other researchers have considered crowdsourcing to collect paraphrases, such as Buzek et al. [2010] who use Mechanical Turk to collect paraphrases of parts of sentences that are difficult to translate automatically thus making their translation easier. Denkowski et al. [2010] also study the impact of using paraphrases in machine translation, employing Mechanical Turk to filter automatically identified paraphrases of sentences to be translated. Both contributions resulted in a comparably small sample of paraphrases between 1 000 and 2 500 samples, allowing for manual double-checking of the results. By contrast, Chen and Dolan [2011] follow up on Dolan and Brockett [2005] who first observed the lack of a large paraphrase corpus: they have collected a large-scale corpus of 85 000 sentence-level paraphrases using Mechanical Turk. Paraphrases were generated independently as one-sentence descriptions of specific topics defined by a short video. Filtering was accomplished manually by discarding judging workers by checking samples of their paraphrases (i.e., not the whole data set was reviewed manually). While the order of magnitude is similar to our paper, differences include that our paraphrases are on passage-level and that we introduce an automatic means of quality control, which has been suggested as future work in the aformentioned paper.

Paraphrase Similarity and Classification. Since the list of metrics used in previous work to compute the similarity between paraphrases is very lengthy, we instead broadly describe the three main categories of paraphrase similarity metrics: word-level metrics, information retrieval metrics, and metrics specifically designed for paraphrasing. First, the word-level metrics make use of the semantic similarity between individual words, and the WordNet::Similarity software [Pedersen et al. 2004] has shown to be a highly-used implementation in the paraphrasing community. Numerous contributions have proposed the combination of individual scores obtained from measuring the similarity of individual terms between samples [Corley and Mihalcea 2005; Fernando and Stevenson 2008]. Second, information retrieval metrics have been applied directly to paraphrasing, such as the Manhattan distance, Euclidean distance, and cosine similarity by Malakasiotis [2009], and a probabilistic model by Das and Smith [2009]. Finally, other measures designed specifically for paraphrasing include measures based on n-grams [Cordeiro et al. 2007a] and asymmetric measures [Cordeiro et al. 2007b]. All the measures overviewed above have either been applied individually, or in combination as part of machine learning experiments. In summary, we stress that there has been no prior work in applying these measures for paraphrases beyond the sentence level in this body of work.

2. PARAPHRASING IN PLAGIARISM DETECTION

Previous studies on the prevalence of plagiarism are numerous. Sheard et al. [2002] summarize three studies with some of the highest rates of dishonesty reported over student university lifetimes at 88%, 90%, and 91.7%. It is unclear how often specific types of resources are plagiarized, perhaps due preferences within different discipline areas. However, a study by Dordoy [2002] about perceptions in dishonest behavior shows that “copying a few paragraphs of an essay from a book/internet” is one of the most common perceived behaviors.

Plagiarising by paraphrasing is one of many types of plagiarism identified by Maurer et al. [2006]. Hamilton et al. [2004] demonstrate that paraphrasing practices are poorly understood using an online test comprising a pool of 25 questions in a university academic integrity workshop. Results show that the correct response rate for the question that asked students to distinguish acceptable and unacceptable paraphrasing examples was the third lowest of all questions. Not only do students poorly understand how to paraphrase correctly, but other studies have shown that this academic integrity skill is not rated as very important compared to others. For example, in a list of 22 types of cheating behavior, Franklyn-Stokes and Newstead [1995] showed that paraphrasing (without references) was the most common behavior type of all, and deemed the third least serious from a combined survey of 112 students and 20 staff. McCabe [2005] discovered similar trends with perceptions of incorrect paraphrasing behavior ranking highest among a group of nine behaviors in a large USA and Canadian survey exceeding 9 000 staff.

The act of plagiarizing often happens as follows (see Figure 2): the plagiarist retrieves a source text, copies passages from that source, and optionally paraphrases the copied passages in order to disguise the plagiarism. Paraphrases found in plagiarism cases are hence clearly beyond the sentence-level.

While the plagiarist may copy from a non-digital source, these days copying from a digital source found on the Web is very quick. This is also why we deem the step of paraphrasing the copied texts to be optional as it requires quite some effort in itself. There are no statistics on the amount of verbatim plagiarism compared to paraphrased plagiarism; however, as the technology to detect plagiarism progresses, paraphrasing may become a major obstacle that stands in between detecting a case of plagiarism or not. This is one of the conclusions that can be drawn from the results of the PAN 2010 plagiarism detection competition.

2.1 Lessons Learned at PAN 2010

The annual PAN workshop and competition series is about uncovering plagiarism, authorship, and social software misuse.[6] Since the first workshop in 2007, PAN has become a platform for the interdisciplinary exchange of researchers and practitioners who address these problems from different angles, and PAN has been organized as evaluation competitions now since 2009. The automatic detection of plagiarism has been at the center of attention from the start, and 33 different plagiarism detectors have been evaluated within the 2009, 2010, and 2011 competitions, some of them thrice in a row. For the purpose of these competitions, we have researched and developed the first large-scale evaluation framework for plagiarism detection [Potthast et al. 2010b].

[6] `http://pan.webis.de`

Table I. Examples of the paraphrase generation strategies employed in PAN 2010 [Potthast et al. 2010a].

	Example paraphrases
Original sentence	The quick brown fox jumps over the lazy dog.
Random text operations	over The. the quick lazy dog <*context word*> jumps brown fox over jumps quick brown fox The lazy. the brown jumps the. quick dog The lazy fox over
Semantic word variation	The quick brown dodger leaps over the lazy canine. The quick brown canine jumps over the lazy canine. The quick brown vixen leaps over the lazy puppy.
POS-preserving word shuffling	The brown lazy fox jumps over the quick dog. The lazy quick dog jumps over the brown fox. The brown lazy dog jumps over the quick fox.

The evaluation framework consists of the PAN plagiarism corpus[7] in the versions of 2009 (PAN-PC-09), 2010 (PAN-PC-10), and 2011 (PAN-PC-11) as well as tailored detection performance measures. The corpora comprise generated plagiarism cases that have been inserted automatically into plain text documents; the 2009 version comprises 94 202 cases in 41 223 documents, the 2010 version comprises 68 558 cases in 27 073 documents, and the 2011 version comprises 61 064 cases in 26 939 documents. Several parameters have been varied during corpus construction, such as the length of a plagiarism case and the percentage of plagiarism per document; however, one of the most revealing parameters proved to be the type of paraphrase in a plagiarism case. In 2009, only automatic paraphrase generation approaches were employed, but 3 671 manually generated paraphrases were introduced for the first time in 2010 and another 938 in 2011. Our rationale for doing so was to investigate the difference in detectability of manually paraphrased plagiarism cases compared to automatically paraphrased plagiarism cases.

As an aside, three heuristics were employed to automatically generate paraphrased plagiarism. Given a source passage t_1, a paraphrase t_2 is created as follows (see Table I for examples):

—*Random Text Operations.* The paraphrase t_2 is created from t_1 by shuffling, removing, inserting, or replacing words or short phrases at random. Insertions and replacements are taken from the document in which t_2 is to be inserted, which serves as part of the world knowledge α.

—*Semantic Word Variation.* The paraphrase t_2 is created from t_1 by replacing words with one of their synonyms, antonyms, hyponyms, or hypernyms, chosen at random from WordNet, which serves as part of the world knowledge α. A word is kept if none are available.

—*POS-preserving Word Shuffling.* The sequence of parts of speech of t_1 is determined and the paraphrase t_2 is created from t_1 by shuffling words at random while retaining the original POS sequence.

To generate various degrees of similarity between sources and paraphrases, the heuristics have been adjusted by varying the number of operations made on a source passage, and

[7] `http://www.webis.de/research/corpora`

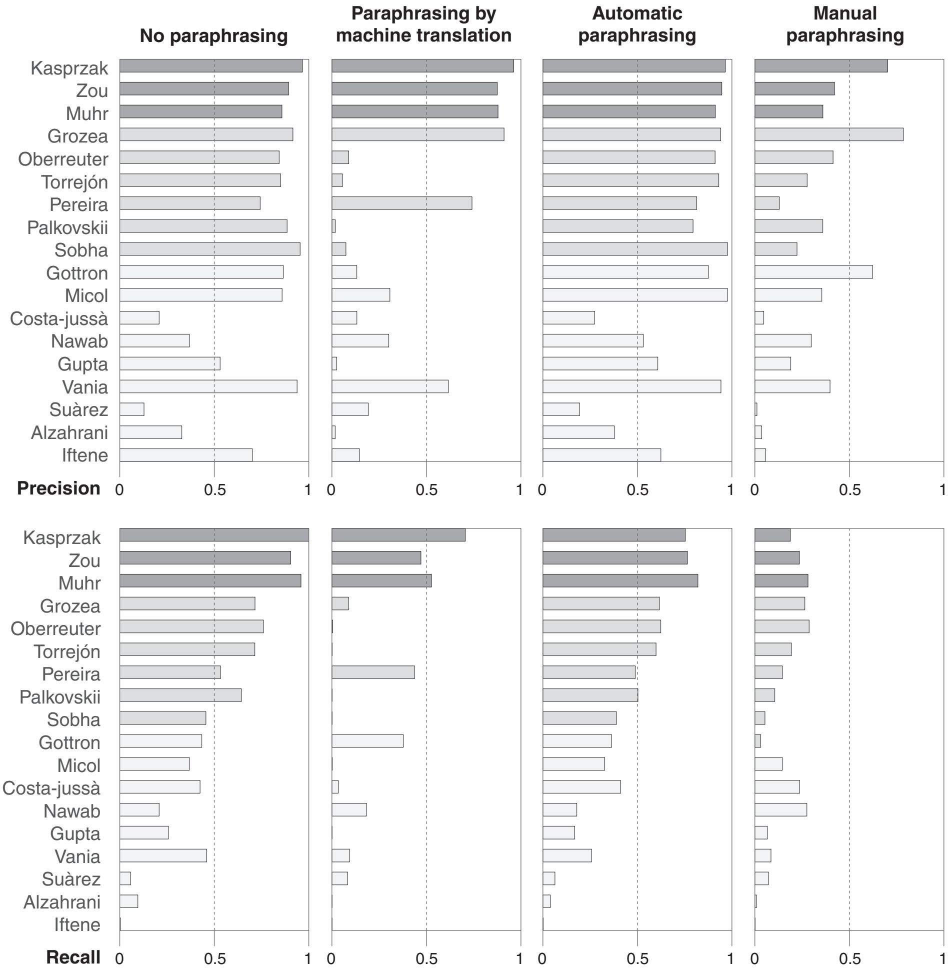

Fig. 3. Plagiarism detection performances in PAN 2010 dependent on the paraphrasing strategy. The detectors are ordered by their overall scores achieved, which is a combination of precision, recall, and a third measure called granularity, which is omitted here for brevity. Each detector is referred to by its lead developer's last name. A complete overview and full references can be found in the PAN 2010 overview paper [Potthast et al. 2010a].

by limiting the range of affected phrases. Unlike some of the more advanced paraphrase generation methods proposed in the literature [Barzilay and Lee 2003; Chevelu et al. 2009; Zhao et al. 2009], these heuristics do not produce well-formed paraphrases that are semantically equivalent to their sources. However, our rationale to use them, anyway, was to create texts that would be considered highly similar under a bag of words model, which most plagiarism detectors employ. Furthermore, most of the existing methods require significant amounts of training data, or they cannot be easily scaled, or both.

Figure 3 shows the detection precision and recall of 18 plagiarism detectors that have been evaluated in PAN 2010, dependent on the type of paraphrase: no paraphrasing, paraphrasing by machine translation from Spanish and German sources to English using the

Google Translate API,[8] automatic paraphrasing, and manual paraphrasing. The first type serves as a baseline for comparison on which the best performing detector of Kasprzak achieved more than 0.99 recall at 0.95 precision. Only two other detectors came close to these figures, whereas the others performed increasingly worse. On machine translated paraphrases, again, the detector of Kasprzak performed best, and only four of the others achieved noteworthy recall. On the automatic paraphrases, the performance characteristics of all detectors are very similar to those of the non-paraphrased plagiarism cases, while the recall is 30% lower on average. Regarding the manually paraphrased plagiarism cases, the precision performance varied greatly; however, more importantly, no detector achieved more than 0.28 recall. These results indicate that there is a wide gap between what automatic plagiarism detectors can detect and what plagiarists can do to disguise their plagiarism. Verbatim copying is easy to detect, whereas manually paraphrased plagiarism cases are quite difficult. However, automatically paraphrased plagiarism sits somewhere in the middle, since the aforementioned paraphrasing heuristics are obviously not sufficient to generate paraphrases with characteristics that come close to those of manual paraphrases.

3. CONTRUCTION OF THE WEBIS CROWD PARAPHRASE CORPUS

This section details our approach to crowdsourcing paraphrases and outlines our findings from applying it. As a result, we introduce the Webis Crowd Paraphrase Corpus 2011 (Webis-CPC-11).[9] The corpus comprises 4 067 text samples with lengths ranging from 28 to 954 words, and their corresponding paraphrases. A further 3 792 cases were rejected and form negative samples. After a brief introduction to Mechanical Turk, we detail the construction of the corpus within the two steps paraphrase acquisition and manual filtering.

3.1 Amazon's Mechanical Turk

Creating a corpus of over 4 000 paraphrases is still an enormous task, which can not be easily accomplished by a single person in a reasonable time frame and at reasonable cost. Even if a single person were to write that many paraphrases, such a corpus would inevitably be biased (for example, by the writing style preferences of that person). Recently, crowdsourcing has become a viable alternative to alleviate these problems: distributing paraphrase generation to a crowd of workers reduces the time and costs to accomplish this task, and it introduces variance into the paraphrases. Our approach to scale up the generation of paraphrases is based on Amazon's Mechanical Turk (AMT), a commercial crowdsourcing service [Barr and Cabrera 2006].

In short, AMT acts as a broker between workers and so-called requesters, who offer tasks and payment for their successful completion. Since real money is involved and workers have anonymity, the platform attracts scammers who try to get paid without actually working. Hence, requesters have the opportunity to check submitted results and reject those that are unsatisfactory. Besides saving money, rigorous result checking is of course a necessity to ensure quality. Crowdsourcing via AMT has gathered considerable interest in research and practice; it has also been demonstrated to be useful for proofreading, writing, and translating texts [Ambati et al. 2010; Bernstein et al. 2010].

[8] `http://code.google.com/apis/language/translate/overview.html`
[9] `http://www.webis.de/research/corpora/webis-cpc-11`

Table II. Example of a paraphrase generated with Mechanical Turk.

Original	**Paraphrase**
The emigrants who sailed with Gilbert were better fitted for a crusade than a colony, and, disappointed at not at once finding mines of gold and silver, many deserted; and soon there were not enough sailors to man all the four ships. Accordingly, the Swallow was sent back to England with the sick; and with the remainder of the fleet, well supplied at St. John's with fish and other necessaries, Gilbert (August 20) sailed south as far as forty-four degrees north latitude. Off Sable Island a storm assailed them, and the largest of the vessels, called the Delight, carrying most of the provisions, was driven on a rock and went to pieces. [Excerpt from "*Abraham Lincoln: A History*" by John Nicolay and John Hay.]	The people who left their countries and sailed with Gilbert were more suited for fighting the crusades than for leading a settled life in the colonies. They were bitterly disappointed as it was not the America that they had expected. Since they did not immediately find gold and silver mines, many deserted. At one stage, there were not even enough men to help sail the four ships. So the Swallow was sent back to England carrying the sick. The other fleet was supplied with fish and the other necessities from St. John. On August 20, Gilbert had sailed as far as forty-four degrees to the north latitude. His ship known as the Delight, which bore all the required supplies, was attacked by a violent storm near Sable Island. The storm had driven it into a rock shattering it into pieces.

3.2 Paraphrase Acquisition

As original texts we have used 4 067 excerpts chosen at random from about 7 000 books downloaded from Project Gutenberg.[10] The excerpts have been reviewed manually in order make sure they consist of passages of text which are amenable to be paraphrased, instead of, for instance, tables or enumerations.

A number of pilot experiments were conducted to refine the task design, the task description, and to determine the pay per task. Unsurprisingly the workers at AMT do not understand well technical terms like "paraphrase" so that in the end, the task was described to them as follows [Potthast et al. 2010b]:

> *Rewrite the original text found below* [on the task web page] *so that the rewritten version has the same meaning, but a completely different wording and phrasing. Imagine a scholar copying a friend's homework just before class, or a plagiarist willing to use the original text without proper quotation.*

The task web page comprised of the task description, the original text to be rewritten, and a multi-line text input field beside it for the paraphrase. Furthermore, a background script monitored the workers, recording their keystrokes. This way it was ensured that workers did not simply copy and paste text as the paraphrase.

As an example, Table II contrasts an original passage and its paraphrase obtained from AMT. The workers have rewritten the original text quite thoroughly, replacing words with synonyms and rearranging the sentence structure while maintaining the original semantics.

Workers were required to be fluent in English reading and writing, and they were informed that every result was to be reviewed. Workers from all nationalities were accepted.

[10] `http://www.gutenberg.org`

Table III. Summary of the 4 067 accepted Mechanical Turk tasks completed by 1 130 workers.

Task statistics					
Tasks per worker		*Work time (minutes)*		*Compensation*	
average	3.66	average	14.79	pay per task (min)	0.06 US$
std. deviation	9.67	std. deviation	22.68	pay per task (max)	0.50 US$
minimum	1	minimum	1	rejection rate	48%
maximum	118	maximum	240		

Worker demographics									
Age		*Education*		*Native speaker*		*Gender*		*Prof. writer*	
18–19	9%	HS	12%	yes	65%	male	39%	yes	11%
20–29	39%	College	33%	no	14%	female	39%	no	66%
30–39	16%	BSc.	19%						
40–49	8%	MSc.	12%						
50–59	5%	Dr.	1%						
60–69	1%								
n/a	22%	n/a	23%	n/a	21%	n/a	22%	n/a	23%

A questionnaire displayed alongside the task description asked about the worker's age, education, gender, native speaking ability, and whether the worker is a professional writer. Completing the questionnaire was optional in order to minimize false answers, but still, these numbers have to be taken with a grain of salt: AMT is not the best environment for such surveys. Table III overviews the worker demographics and task statistics. The average worker completing our tasks appears to be a well-educated male or female in their twenties, whose first language is English.

3.3 Filtering Crowd Paraphrases Manually

Each paraphrase submitted by a worker has been checked manually, and those which did not meet our requirements have been rejected. Every submitted paraphrase has been read to ensure that it has the same meaning as the original, and that it is well-formed using proper English. However, since only parts of every paraphrase were checked this way, the reviewer was also given additional information to help with the decisions:

- length ratio of paraphrase to original,
- n-gram vector space model similarity of original and paraphrase for $n \in \{1, 5, 10\}$,
- number of keystrokes with the full keylog, and
- work time of the worker.

No submitted paraphrases have been accepted that are grossly longer or shorter than the original, which eliminates many entailment relationships—entailment is not the focus of this study. Moreover, paraphrases with high n-gram similarities for $n > 1$ have been rejected, while the 1-gram similarity was expected to be high. Therefore we also rejected paraphrases which contained large chunks of unmodified text. Next, if a worker just copied and pasted some text as the paraphrase or worked extremely fast compared to others, the text was checked more closely in order to identify spammers. After completing the tasks on Mechanical Turk and after further double-checking by a student assistant, 4 067 pairs of original text and paraphrase remained, whereas 3 792 submitted paraphrases have been rejected.

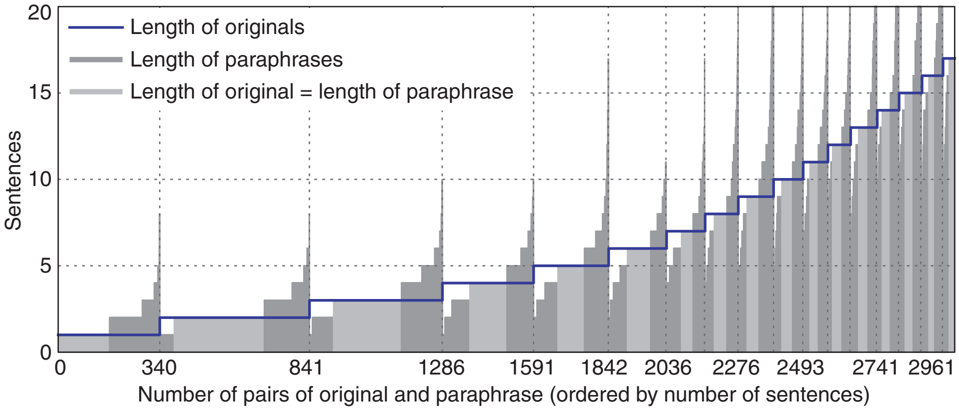

Fig. 4. Distribution of the number of sentences between original texts compared with the accepted paraphrases. The step curve (outlined blue) shows the number of sentences of the originals and the vertical bars (outlined gray) show the number of sentences of paraphrases, dependent on their respective originals. The step curve being on a level with the bars (outlined light gray) indicates that those paraphrases have the same number of sentences as the originals. For brevity, originals of length greater than 17 sentences and their respective paraphrases have been omitted (i.e., the plot shows 3 000 of the 4 067 accepted originals and paraphrases).

3.4 Post-hoc Corpus Analysis

During manual reviewing, we observed many paraphrasing patterns applied by the workers. For instance, some workers rewrote the text sentence by sentence, whereas others combined or split them. Some exchanged words on a one-for-one basis, while others reorganized sentence structures. Some stuck closely to the line of thoughts of the original, while others made the paraphrase more concise or more verbose. These observations, however, were not recorded systematically, so some doubts remained as to whether the paraphrases produced by the workers are indeed passage-level paraphrases as opposed to simply a sequence of sentence-level paraphrases. Put another way, the question is whether workers mostly resorted to repeated sentence-level paraphrasing to solve the task or not.

To gain better insight into whether the accepted paraphrases are actually worthwhile on the passage-level, we have checked the corpus with regard to the following hypothesis: if most workers applied sentence-level paraphrasing to solve our task, then most paraphrases will consist of the same number of sentences compared to their respective originals. For each original and paraphrase pair, the number of sentences was counted using the OpenNLP maximum entropy sentence extractor for plotting as Figure 4. From this experiment, we observed that 38.7% of the paraphrases have the same number of sentences as their originals, while the majority of paraphrases are of a different length. Moreover, the length distributions shown in the figure, which depend on the length of the originals, tell something about how much the paraphrases have been shortened or lengthened by the workers. Discounting errors made by the sentence extractor, and the fact that some pairs have the same length by coincidence, we can safely reject the hypothesis that workers mostly resort to sentence-level paraphrasing.

4. FILTERING CROWD PARAPHRASES AUTOMATICALLY

Having introduced the Webis Crowd Paraphrase Corpus, we now study whether and how the corpus acquisition can be further automated. After acquiring paraphrases from a crowd, the next step is to review them for quality, which turns out to be a time-consuming task as well. Automating the quality assessment bears the potential of further scaling up corpus construction in the future.

We tackle this problem with machine learning, and consider the six often applied paraphrasing patterns collated by Boonthum [2004] as a working definition of what constitutes a "good" paraphrase. These patterns are synonym use, changes between active and passive voice, changing word forms and parts of speech, breaking sentences down, replacing words with definitions or meanings, and varying sentence structures. However, in the context of crowdsourcing paraphrases it is important to also define what constitutes a "bad" paraphrase or a non-paraphrase. From the submissions rejected during human assessment, we have identified cases of duplicates, near-duplicates, unrelated texts, empty samples, automated one-for-one word substitution, and doing something unrelated due to failing to follow instructions, such as attempting to improve the quality of the text instead of paraphrasing it. By the definition of paraphrases given in the introduction, many of these examples must be considered paraphrases (e.g., duplicating a text is a trivial form of paraphrasing it), however, keeping them as positive examples in our corpus would spoil the goal of creating a useful resource for research. In our experiments, we hence investigate whether well-known similarity measures that are tailored to paraphrase identification can serve as features to distinguish between good and bad paraphrases, using the human assessments of approval and rejection as ground truth.

4.1 Similarity Analysis of Original Texts and Paraphrases

To measure the similarity between the original samples and the accepted/rejected paraphrases, we use ten established paraphrase similarity metrics from the literature. For our analysis, we chose the Edit/Levenshtein Distance (normalized), Simple Word N-gram Overlap, Exclusive LCP (Longest Common Prefix) N-gram Overlap, the BLEU Metric, and the Sumo Metric by Cordeiro et al. [2007a], as well as the Trigonometric, Parabolic, Entropic, Gaussian, and Linear metrics by Cordeiro et al. [2007b]. We note our novel application of these metrics as they were previously used for recognizing existing paraphrases, and not for generating new ones. We do not repeat the details of these metrics, since they are already explained at length in the two aforementioned papers.

The results in Figure 5 indicate many differences between the accepted and rejected groups. First, the normalized Edit Distance for the rejected group is much higher than for the accepted group, which suggests that the workers creating the rejected paraphrases generally made less changes, resulting in higher similarity. Therefore, the time taken to complete the task proportional to the length of the paraphrase may be a feature of interest.

The Simple Word N-gram Overlap scores indicate that the rejected paraphrases contain less original content, as these scores are clearly higher. This observation is reiterated with the Exclusive LCP N-gram Overlap scores, but the difference is less profound, since this metric removes much redundancy from the former metric. Considering that any matching 4-gram also includes two matching 3-grams, three matching 2-grams, and four matching 1-grams, only one matching n-gram of each type is taken for the Exclusive LCP N-gram Overlap metric.

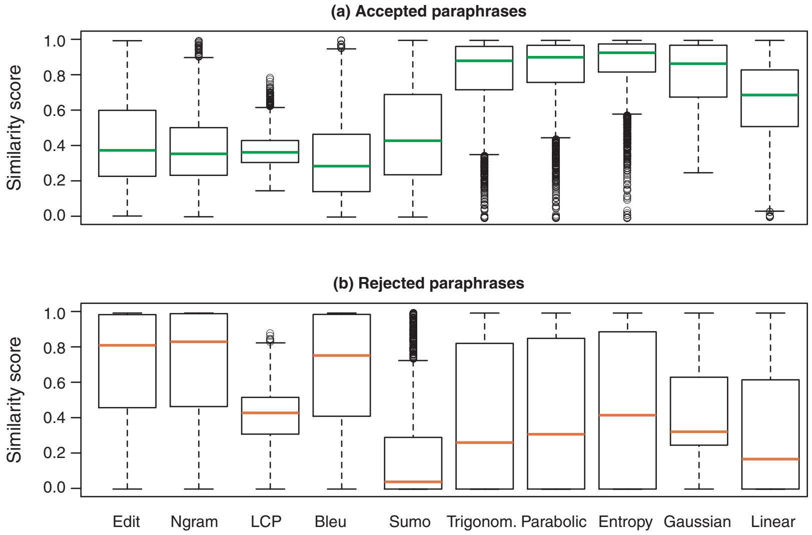

Fig. 5. Distribution of similarity scores for accepted and rejected paraphrases. The boxes represent the interquartile range (IQR), and the whiskers are plotted at 1.5×IQR.

The BLEU Metric is also based on n-gram comparisons. This metric takes into account the difference in lengths of the samples, and applies larger penalties to larger discrepancies [Papineni et al. 2002]. Like the two previous n-gram metrics, the results indicate that there is less original content in the rejected paraphrases.

The Sumo Metric also clearly separates the accepted and rejected paraphrase groups. A key feature of the Sumo Metric is that it is built with identifying duplicates and near-duplicates as a key design decision [Cordeiro et al. 2007a], which makes it ideal for evaluating the effectiveness of translation systems [Cordeiro et al. 2007b]. This is particularly important for the Webis-CPC-11 corpus, as we need to deal with dubious cases dealing with duplicates, near-duplicates, and automated single-word substitutions. We believe the results shown reflect this quality. Note that the Sumo Metric would be less suited to the MSRPC and KMC corpora, as these are not concerned with duplicates and near-duplicates.

The remaining five metrics demonstrate high variance for the rejected paraphrase group. First, the long tails of the trigonometric, parabolic, and entropic metrics for the accepted paraphrases indicate that we are likely to be dealing with data that does not follow a normal distribution, as perhaps do many of the other measurement sets. Also, the high variance in these results highlights the difficulty of our task, as we need to be able to reject submitted paraphrases that present a very wide range of similarity scores. This finding suggests that the previous work where others have "used numerous thresholds to decide definitely whether two sentences are similar and infer the same meaning" [Kozareva and Montoyo 2006] is not suitable for crowdsourcing, since almost all possible scores are appearing in the rejected paraphrase group. This does not change the fact that they may still be effective for non-crowdsourced paraphrase corpora, but the thresholds applied may not hold between corpora. This conclusion leads us to believe that machine learning is our best option for automatically classifying crowdsourced paraphrases as legitimate paraphrases or otherwise by pooling the similarity metrics together, as explored in the next section.

4.2 Classifying Accepted and Rejected HITs

Using the ten metrics explored in the previous section, we now explore using them as features for classifying crowdsourced paraphrases as legitimate or otherwise. We chose to test the five classification algorithms that we identified as having been applied to classification in the paraphrasing literature: decision trees [Fernando 2007; Wan et al. 2006], k-nearest neighbor [Kozareva and Montoyo 2006; Wan et al. 2006], maximum entropy [Kozareva and Montoyo 2006; Malakasiotis 2009], naive bayes [Wan et al. 2006], and support vector machines [Brockett and Dolan 2005; Dolan and Brockett 2005; Kozareva and Montoyo 2006; Qiu et al. 2006; Wan et al. 2006].

We ran ten iterations of a ten-fold cross validation experiment using the above classifiers for our experiments. Weka version 3.6.4 implementations for each were chosen as listed in the Table IV caption. We used the default Weka parameters for all classifiers with a few exceptions. First we wanted our classifiers to produce the best distribution of class probability estimates possible (for Section 4.3), so we built a logistic model for the support vector machine, used the distance weighting measure for the k-nearest neighbor classifier that generated the largest distribution, and used an unpruned decision tree. For the k-nearest neighbor classifier, we experimented with $k \in \{1, 2, 5, 10, 20, 50, 100, 200, 500, 1\,000\}$ and only report $k = 50$ for brevity, which had the highest positive predictive value at 0.25 true positive rate. It was clear that low values of k were poor for our data with noise. Table IV provides performance data for all five classifiers expressed in terms of confusion matrix data, positive predictive value (or precision), true positive rate (or recall), other error rates, and accuracy. With regards to our corpus construction goals, precision is the most important column in this table, as it describes the hit rate with positive samples. Negative samples are trivial to generate. Consequently, accuracy is a less suitable choice since it includes both positive and negative samples together. The Table IV results show that the k-nearest neighbor classifier has the highest precision at 0.81 (the difference with the decision tree was significant ($p = 6.27 \times 10^{-6}$) based upon a two-sample test for equality of proportions with continuity correction), but this is clearly not good enough for automating paraphrase acquisition. One option to improve this score is to trade away recall for precision. We now explore if this is indeed a viable and cost-effective proposition in the next section.

4.3 Cost Trade-Off and Analysis

Figure 6 shows the precision-recall trade-off for all five classifiers. For the generally highest precision k-nearest neighbor classifier, at 0.25 recall we get 0.987 precision, at 0.5 recall we get 0.966 precision, and precision drops away more steadily for higher recall levels. Reading this the other way, if we want an automatically generated paraphrase corpus with these precision levels, we will need to discard 75% or 50% of our legitimate paraphrases respectively.

Other results show that the highest precision classifier at the lowest recall scores is unclear, but the results are visibly less reliable at this point, as it only takes very few false positives to disturb the initial trends shown in the graphs. The other anomaly is the decision tree classifier that has the poorest distribution of class probabilities, as can be seen by the straight line sections on the graph. Using an unpruned decision tree was the best option we identified for generating good probability estimates [Su 2007].

So far, we have very high precision results in Figure 6 at many reasonable recall levels,

Table IV. Performance for all classifiers. The classifier class refers to the Weka 3.6.4 implementation that we used [Holmes et al. 1994]. (The Weka implementations are the c4.5 decision tree (trees.J48), k-nearest neighbor (lazy.IBk), maximum entropy (functions.Logistic), naive bayes (bayes.NaiveBayes), and support vector machine (functions.SMO) classifiers. The common parts of the class name are omitted for brevity; for example, the decision tree full name is "weka.classifiers.trees.J48".) The performance data is expressed in terms of the number of true positives (TP), number of false positives (FP), number of true negatives (TN), number of false negatives (FN), positive predictive value (PPV = TP / (TP + FP) = precision), true positive rate (TPR = TP / (TP + FN) = recall), false positive rate (FPR = FP / (FP + TN)), true negative rate (TNR = TN / (TN + FP)), false negative rate (FNR = FN / (FN + TP)), and accuracy (ACC = (TP + TN) / (TP + FP + TN + FN)). The number of true positives, false positives, true negatives, and false negatives sums to 78 590 for all rows.

Classifier Name	TP	FP	TN	FN	PPV	TPR	FPR	TNR	FNR	ACC
					Precision	Recall				
c4.5 decision tree	37 007	9 300	28 620	3 663	0.80	0.91	0.25	0.75	0.09	0.84
k-nearest neighbor	36 482	8 500	29 420	4 188	0.81	0.90	0.22	0.78	0.10	0.84
maximum entropy	37 146	10 546	27 374	3 524	0.78	0.91	0.28	0.72	0.09	0.82
naive bayes	36 494	12 319	25 601	4 176	0.75	0.90	0.32	0.68	0.10	0.79
support vector machine	37 835	10 870	27 050	2 835	0.78	0.93	0.29	0.71	0.07	0.83

so for anyone who wants to perform a task similar to our own, our collection can be used as training data in order to eliminate the crowdsourcing filtering phase and achieve similar precision. On the other hand, others may want to build a completely different corpus, such as a sentence-level corpus instead of a passage-level corpus. New training data is needed in this case. The experiment demonstrated in Figure 6 uses a 10-fold cross validation experiment design based on a 90% training and 10% testing split. It is now of interest to explore if smaller training partitions produce results with similar precision, so that at least part of the crowdsourcing filtering stage can be automated. Figure 7 shows that precision does not vary much when the training set size is reduced incrementally down to only 25% (1 965 samples), particularly for lower recall levels. For example, at 0.25 recall, precision drops from 1.000 to 0.982 (-0.018), and at 0.50 recall, precision drops from 0.990 to 0.952 (-0.038). The precision drop when 10% training data (786 samples) is used is however more profound.

Regardless if our data is used for training or if some new training data must be developed, it becomes necessary to analyze the monetary and time trade-offs. Table V shows a detailed breakdown of our expenditures for building WebIS-CPC-11 with each main task

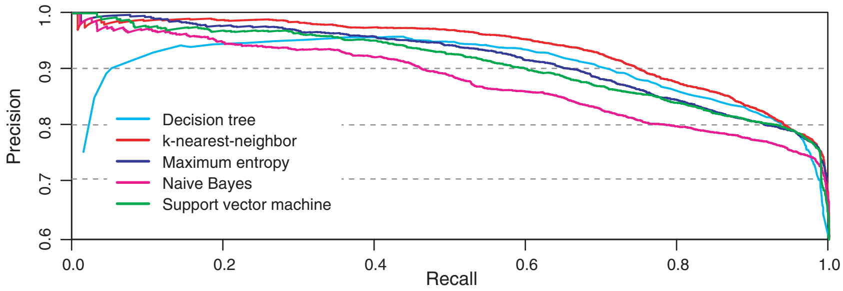

Fig. 6. Recall-precision trade-off for all five classifiers. The results show that the k-nearest neighbor classifier has the highest precision at many recall levels.

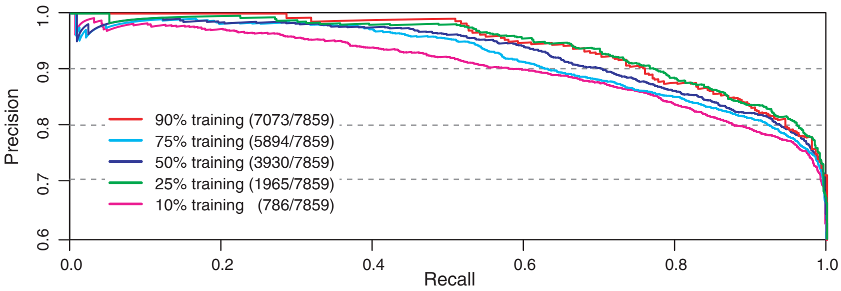

Fig. 7. K-nearest neighbor classifier ($k = 50$) using training data sets of various sizes. These results are for a single iteration of each training set and testing set size split without cross-validation (7 859 instances in total).

aggregated in the second column of Table VI. These costs are then comparable to a range of automated scenarios in the last six columns of Table VI. We note that the new "Discarded HITs" row is added to Table VI as we are proceeding with the plan to accept all bad HITs in Mechanical Turk, in order to easily eliminate most or all of the university employee time spent on manually checking off the HITs. The goal is for machine learning to replace this phase to save as much time and money as possible in building a corpus of acceptable paraphrases.

First, let's compare the costs of our corpus (second column) with the costs associated with repeating our procedure to build similar corpora (third and fourth column) in Table VI. Building similar corpora assumes that the Webis-CPC-11 corpus is used for training data. When building Webis-CPC-11, we first note that we had zero expense for rejected HITs, as manual filtering was performed completely for building the ground truth that forms the experiments in our paper. In order to save the manual filtering time, we have to pay the workers who did bad work ($1 193.06). Furthermore, we incur extra costs for discarding the acceptable HITs that are assigned low class probabilities in the precision/recall trade-off ($875.58 or $992.05) and increasing the experiment size by that proportion. However, all of the manual filtering stage is saved with the automated method ($2 782.92). Depending on the desired precision of the corpus, the cost savings are $967.24 or $850.77 for 0.950 and 0.980 precision respectively, with 111 hours of saved time for the university research staff in both cases.

It is of course possible that a corpus dissimilar to Webis-CPC-11 is required, and that our corpus is not suitable for training data. In this scenario, it becomes necessary to do some portion of the filtering manually to generate sufficient training data for a machine learning classifier to process the remainder automatically. For the scenarios in the last four columns of Table VI we still incur the same cost for paying the workers for the bad work, and in addition the costs for the discarded HITs go up because these precision levels are achieved at lower rates of recall for lower amounts of training data. Furthermore, savings for the manual filtering step are now only partial (75% or 50%). As a result, Table VI shows that our expenses are generally quite neutral when using 25% training data ($25.93 saving and $275.47 loss respectively), but the costs start to become prohibitive for 50% training data (losses of $465.81 and $946.24 respectively). Nevertheless, we still have excellent time savings in all cases, so these results provide important new options for cost-time compromises.

Table V. Cost estimates. Note that we did not pay the workers when the HITs were rejected. These costs are given as an indication of money that would be lost if no manual result filtering was done.

Item	Unit Value	Quantity	Cost
Acquisition Stage — Task Development			
Research assistant task development	$34.12	10h	$341.20
Student assistant task development	$11.80	20h	$236.00
Acquisition Stage — Accepted HITs			
Worker approved HITs	$0.06	1 065	$63.90
Worker approved HITs	$0.08	685	$54.80
Worker approved HITs	$0.10	534	$53.40
Worker approved HITs	$0.20	435	$87.00
Worker approved HITs	$0.35	466	$163.10
Worker approved HITs	$0.45	320	$144.00
Worker approved HITs	$0.50	641	$320.50
Acquisition Stage — Potential Rejected HIT Value			
Worker rejected HITs	$0.06	345	$20.70
Worker rejected HITs	$0.08	362	$28.96
Worker rejected HITs	$0.10	319	$31.90
Worker rejected HITs	$0.20	413	$82.60
Worker rejected HITs	$0.35	491	$171.85
Worker rejected HITs	$0.45	689	$310.05
Worker rejected HITs	$0.50	1 094	$547.00
Filtering Stage — Manual			
Research assistant development	$34.12	10h	$341.20
Research assistant filtering	$34.12	52h	$1 774.24
Research assistant meetings	$34.12	4h	$136.48
Student assistant development	$11.80	10h	$118.00
Student assistant filtering	$11.80	31h	$365.80
Student assistant meetings	$11.80	4h	$47.20
Filtering Stage — Automated			
Machine learning	$34.12	12h	$409.44

5. CONCLUSIONS AND FUTURE WORK

The starting point for our research was the observation that there is no benchmark corpus for paraphrases that are longer than a single sentence. However, passage-level paraphrasing is a frequent and naturally occurring phenomenon as in text plagiarism, for example. That such a benchmark corpus is necessary and that the quality of its paraphrases matters is one of the lessons learned at the PAN 2010 international plagiarism detection competition. Though the compilation of a new benchmark corpus follows a common two-step approach, namely, paraphrase acquisition followed by paraphrase filtering, we had to break new ground: for the first step no algorithms are at hand and we have applied crowdsourcing, which immediately raised the question whether the filtering step can be accomplished with a combination of the existing paraphrase analysis metrics.

This question can be answered with “yes”: in our experiments we achieved a precision of 0.980 precision at 0.523 recall. In related work, we note that Wan et al. [2006] achieved 75% accuracy in a paraphrase recognition experiment using a support vector machine with 17 features on the MSRPC corpus. However, we explained that precision is the most

Table VI. Cost projections. All repeat and new corpus scenarios provide large time savings with a mixture of cost savings or additional cost expenditures.

			Corpus Name and Properties				
Name	Webis-CPC-11	Repeat	Repeat	New	New	New	New
	Corpus	Corpus	Corpus	Corpus	Corpus	Corpus	Corpus
Training %	—	100	100	25	25	50	50
Precision	—	0.950	0.980	0.950	0.980	0.950	0.980
Recall	—	0.579	0.523	0.509	0.316	0.559	0.328
			Corpus Costs				
Task development	\$577.20	\$577.20	\$577.20	\$577.20	\$577.20	\$577.20	\$577.20
Accepted HITs	\$886.70	\$886.70	\$886.70	\$886.70	\$886.70	\$886.70	\$886.70
Rejected HITs	—	\$1 193.06	\$1 193.06	\$1 193.06	\$1 193.06	\$1 193.06	\$1 193.06
Discarded HITs	—	\$875.58	\$992.05	\$1 121.16	\$1 422.56	\$917.17	\$1 397.60
Manual filtering	\$2 782.92	—	—	\$695.73	\$695.73	\$1 391.46	\$1 391.46
Machine learning	\$409.44	\$136.48	\$136.48	\$136.48	\$136.48	\$136.48	\$136.48
TOTAL COST	\$4 636.26	\$3 669.02	\$3 785.49	\$4 610.33	\$4 911.73	\$5 102.07	\$5 582.50
			Savings from Webis-CPC-11				
Money Saved	—	\$967.24	\$850.77	\$25.93	-\$275.47	-\$465.81	-\$946.24
Time Saved	—	111h	111h	70.25h	70.25h	49.5h	49.5h

appropriate metric, so our results form a new benchmark. Moreover, since our paraphrases provide a much higher variance in regard to sample length and vocabulary compared to single sentence paraphrases, our classification model proves to be robust as well. With the approach presented in this paper we have compiled the Webis-CPC-11 corpus comprising 7 859 positive and negative pairs, which is now available to the public.[11]

With regards to economics, we have excellent cost and time savings when our corpus can be used as training data in the construction of similar corpora. In the case where only the methodology can be repeated, but around 2 000 training samples is required (about 25% of Webis-CPC-11), the costs are fairly neutral, but we still have excellent time savings. In future work, we propose to also conduct some machine learning on the bad workers, in order to try to recoup some of the money that was paid to them that they are not entitled to. Indeed, crowdsourcing could also be applied to the filtering phase, to provide another trade-off. Flagging borderline cases for manual review is also of interest.

An important avenue for future research is the analysis of the commonalities and differences between the classification of sentence-level paraphrases and passage-level paraphrases. From such an analysis we can learn to develop an even more robust classifier and better understand the paraphrase recognition process. The MSRPC corpus is a good candidate for this work.

We can also consider the classification step as a one-class problem [Tax 2001] instead of a two-class problem. This is different to the view in previous paraphrasing work where it has been stated that “paraphrase recognition reduces to a two class problem” [Kozareva and Montoyo 2006]. We suggest that our accepted paraphrases represent the target class, and all rejected paraphrases represent the outlier class. We recommend this approach because we have a large variety of negative samples as discussed above, potentially including some cases that may still be not well understood.

[11] `http://www.webis.de/research/corpora/webis-cpc-11`

A final idea for future work would be to investigate empirically the question of paraphrasing versus entailment, to find which one is better for modeling plagiarism due to being more frequent in practice.

Acknowledgements

We thank João Paulo Cordeiro and colleagues for their kind assistance with operationalizing the paraphrase similarity metrics.

REFERENCES

ALONSO, O. AND MIZZARO, S. 2009. Can we get Rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In *Proceedings of the First SIGIR Workshop on the Future of IR Evaluation*, S. Geva, J. Kamps, C. Peters, T. Sakai, A. Trotman, and E. Voorhees, Eds. IR Publications, Boston, Massachusetts, 15–16.

AMBATI, V., VOGEL, S., AND CARBONELL, J. 2010. Active Learning and Crowd-Sourcing for Machine Translation. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds. European Language Resources Association, Valletta, Malta, 2169–2174.

ANDROUTSOPOULOS, I. AND MALAKASIOTIS, P. 2010. A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Artificial Intelligence Research 38,* 1 (May), 135–187.

BARR, J. AND CABRERA, L. F. 2006. AI gets a Brain. *Queue 4,* 4 (May), 24–29.

BARZILAY, R. AND LEE, L. 2003. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *Proceedings of the Third Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, E. Hovy, M. Hearst, and M. Ostendorf, Eds. Association for Computational Linguistics, Edmonton, Canada, 16–23.

BERNSTEIN, M. S., LITTLE, G., MILLER, R. C., HARTMANN, B., ACKERMAN, M. S., KARGER, D. R., CROWELL, D., AND PANOVICH, K. 2010. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of the Twenty-Third Annual ACM Symposium on User Interface Software and Technology*, K. Perlin, M. Czerwinski, and R. Miller, Eds. ACM, New York City, New York, 313–322.

BOONTHUM, C. 2004. iSTART: Paraphrase Recognition. In *Proceedings of the Fifth ACL Workshop on Student Research*, L. van der Beek, D. Genzel, and D. Midgley, Eds. Association for Computational Linguistics, Barcelona, Spain, 55–60.

BRABHAM, D. C. 2008a. Crowdsourcing as a Model for Problem Solving. *Convergence: The International Journal of Research into New Media Technologies 14,* 1 (Feb.), 75–90.

BRABHAM, D. C. 2008b. Moving the Crowd at iStockphoto: The Composition of the Crowd and Motivations for Participation in a Crowdsourcing Application. *First Monday 13,* 6 (June).

BROCKETT, C. AND DOLAN, W. B. 2005. Support Vector Machines for Paraphrase Identification and Corpus Construction. In *Proceedings of The Third International Workshop on Paraphrasing*, M. Dras and K. Yamamoto, Eds. Kazuhide Yamamoto, Jeju, South Korea, 1–8.

BUZEK, O., RESNIK, P., AND BEDERSON, B. 2010. Error Driven Paraphrase Annotation using Mechanical Turk. In *Proceedings of the First NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, C. Callison-Burch and M. Dredze, Eds. Association for Computational Linguistics, Los Angeles, California, 217–221.

CALLISON-BURCH, C. 2009. Fast, Cheap, and Creative: Evaluating Translation Quality using Amazon's Mechanical Turk. In *Proceedings of the Fourteenth Conference on Empirical Methods in Natural Language Processing*, P. Koehn and R. Mihalcea, Eds. Association for Computational Linguistics, Singapore City, Singapore, 286–295.

CALLISON-BURCH, C. AND DREDZE, M. 2010. Creating Speech and Language Data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT First Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, C. Callison-Burch and M. Dredze, Eds. Association for Computational Linguistics, Los Angeles, California, 1–12.

CHEN, D. AND DOLAN, W. 2011. Collecting Highly Parallel Data for Paraphrase Evaluation. In *Proceedings of the Forty-Ninth Annual Meeting of the Association for Computational Linguistics: Human Language Tech-*

nologies, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds. Association for Computational Linguistics, Portland, Oregon, 190–200.

CHEVELU, J., LAVERGNE, T., LEPAGE, Y., AND MOUDENC, T. 2009. Introduction of a New Paraphrase Generation Tool Based on Monte-Carlo Sampling. In *Proceedings of the Forty-Seventh Annual Meeting of the Association for Computational Linguistics and the Fourth International Joint Conference on Natural Language Processing of the AFNLP*, K.-Y. Su, J. Su, and J. Wiebe, Eds. The Association for Computer Linguistics, Suntec, Singapore, 249–252.

CLOUGH, P., GAIZAUSKAS, R., PIAO, S., AND WILKS, Y. 2002. METER: MEasuring TExt Reuse. In *Proceedings of the Fortieth Annual Meeting on Association for Computational Linguistics*, P. Isabelle, Ed. Association for Computational Linguistics, Philadelphia, Pennsylvania, 152–159.

COHN, T., CALLISON-BURCH, C., AND LAPATA, M. 2008. Constructing Corpora for the Development and Evaluation of Paraphrase Systems. *Computational Linguistics 34,* 4 (Dec.), 597–614.

CORDEIRO, J., DIAS, G., AND BRAZDIL, P. 2007a. A Metric for Paraphrase Detection. In *Proceedings of the Second International Multi-Conference on Computing in the Global Information Technology*, M. Boicu, J. Costa-Requena, D. Thiebaut, C. Popoviciu, B. Tuy, and G. V. de Velde, Eds. IEEE Computer Society, Gosier, Guadeloupe, 1–6.

CORDEIRO, J., DIAS, G., AND BRAZDIL, P. 2007b. New Functions for Unsupervised Asymmetrical Paraphrase Detection. *Journal of Software 2,* 4 (Oct.), 12–23.

CORLEY, C. AND MIHALCEA, R. 2005. Measuring the Semantic Similarity of Texts. In *Proceedings of the First ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, B. Dolan and I. Dagan, Eds. Association for Computational Linguistics, Ann Arbor, Michigan, 13–18.

DAS, D. AND SMITH, N. A. 2009. Paraphrase Identification as Probabilistic Quasi-Synchronous Recognition. In *Proceedings of the Joint Conference of the Forty-Seventh Annual Meeting of the ACL and the Fourth International Joint Conference on Natural Language Processing of the AFNLP*, K.-Y. Su, J. Su, and J. Wiebe, Eds. Association for Computational Linguistics, Suntec, Singapore, 468–476.

DENKOWSKI, M., AL-HAJ, H., AND LAVIE, A. 2010. Turker-Assisted Paraphrasing for English-Arabic Machine Translation. In *Proceedings of the First NAACL HLT Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*, C. Callison-Burch and M. Dredze, Eds. Association for Computational Linguistics, Los Angeles, California, 66–70.

DOLAN, W. B. AND BROCKETT, C. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of The Third International Workshop on Paraphrasing*, M. Dras and K. Yamamoto, Eds. Kazuhide Yamamoto, Jeju, South Korea, 1–8.

DORDOY, A. 2002. Cheating and Plagiarism: Student and Staff Perceptions at Northumbria. In *Proceedings of the Northumbria Conference — Educating for the Future*. Northumbria University, Newcastle, United Kingdom, 1–6.

FERNANDO, S. 2007. Paraphrase Identification. M.S. thesis, Department of Computer Science, University of Sheffield, Sheffield, United Kingdom.

FERNANDO, S. AND STEVENSON, M. 2008. A Semantic Approach to Paraphrase Identification. In *Proceedings of the Eleventh Annual Research Colloquium of the UK Special-Interest Group for Computational Lingusitics*. University of Oxford, Oxford, United Kingdom, 1–7.

FRANKLYN-STOKES, A. AND NEWSTEAD, S. E. 1995. Undergraduate Cheating: Who does what and why? *20,* 2 (June), 159–172.

HAMILTON, M., TAHAGHOGHI, S. M. M., AND WALKER, C. 2004. Educating Students about Plagiarism Avoidance—A Computer Science Perspective. In *Proceedings of the Twelfth International Conference on Computers in Education*, E. McKay, Ed. Common Ground Publishing, Melbourne, Australia, 1275–1284.

HOLMES, G., DONKIN, A., AND WITTEN, I. 1994. Weka: A Machine Learning Workbench. In *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems*, J. Sitte, Ed. IEEE Computer Society Press, Brisbane, Australia, 357–361.

KITTUR, A., CHI, E. H., AND SUH, B. 2008. Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, M. Czerwinski, A. Lund, and D. Tan, Eds. ACM, Florence, Italy, 453–456.

KNIGHT, K. AND MARCU, D. 2002. Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Artificial Intelligence 139,* 1 (July), 91–107.

KOZAREVA, Z. AND MONTOYO, A. 2006. Paraphrase Identification on the Basis of Supervised Machine Learning Techniques. In *Proceedings of the Fifth International Conference on NLP*, T. Salakoski, F. Ginter, S. Pyysalo, and T. Pahikkala, Eds. Springer, Turku, Finland, 524–533.

LEIMEISTER, J., HUBER, M., BRETSCHNEIDER, U., AND KRCMAR, H. 2009. Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. *Journal of Management Information Systems 26,* 1 (July), 197–224.

MADNANI, N. AND DORR, B. J. 2010. Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods. *Computational Linguistics 36,* 3 (Sept.), 341–387.

MALAKASIOTIS, P. 2009. Paraphrase Recognition using Machine Learning to Combine Similarity Measures. In *Proceedings of the First ACL-IJCNLP Student Research Workshop*, D. M. D. Dimalen, J. R. Finkel, and B. Thomson, Eds. Association for Computational Linguistics, Suntec, Singapore, 27–35.

MAURER, H., KAPPE, F., AND ZAKA, B. 2006. Plagiarism—A Survey. *Journal of Universal Computer Science 12,* 8 (Aug.), 1050–1084.

MCCABE, D. L. 2005. Cheating Among College and University Students: A North American Perspective. *1,* 1, 1–11.

PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W.-J. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Fortieth Annual Meeting on Association for Computational Linguistics*, P. Isabelle, Ed. Association for Computational Linguistics, Philadelphia, Pennsylvania, 311–318.

PEDERSEN, T., PATWARDHAN, S., AND MICHELIZZI, J. 2004. WordNet::Similarity: Measuring the Relatedness of Concepts. In *Demonstration Papers at HLT-NAACL 2004*, D. Palmer, J. Polifroni, and D. Roy, Eds. Association for Computational Linguistics, Boston, Massachusetts, 38–41.

POTTHAST, M., BARRÓN-CEDEÑO, A., EISELT, A., STEIN, B., AND ROSSO, P. 2010. Overview of the 2nd International Competition on Plagiarism Detection. In *Notebook Papers of CLEF 10 Labs and Workshops*, M. Braschler and D. Harman, Eds.

POTTHAST, M., STEIN, B., BARRÓN-CEDEÑO, A., AND ROSSO, P. 2010. An Evaluation Framework for Plagiarism Detection. In *23rd International Conference on Computational Linguistics (COLING 10)*, C.-R. Huang and D. Jurafsky, Eds. Association for Computational Linguistics, Stroudsburg, PA, USA, 997–1005.

QIU, L., KAN, M.-Y., AND CHUA, T.-S. 2006. Paraphrase Recognition via Dissimilarity Significance Classification. In *Proceedings of the Eleventh Conference on Empirical Methods in Natural Language Processing*, D. Jurafsky and E. Gaussier, Eds. Association for Computational Linguistics, Sydney, Australia, 18–26.

QUINN, A. J. AND BEDERSON, B. B. 2011. Human Computation: A Survey and Taxonomy of a Growing Field. In *Proceedings of the Twenty-Ninth Annual Conference on Human Factors in Computing Systems*, D. Tan, G. Fitzpatrick, C. Gutwin, B. Begole, and W. A. Kellogg, Eds. ACM, Vancouver, Canada, 1403–1412.

SHEARD, J., DICK, M., MARKHAM, S., MACDONALD, I., AND WALSH, M. 2002. Cheating and Plagiarism: Perceptions and Practices of First Year IT Students. In *Proceedings of the Seventh Annual Conference on Innovation and Technology in Computer Science Education*, M. E. Caspersen, D. Joyce, D. Goelman, and I. Utting, Eds. ACM, Aarhus, Denmark, 183–187.

SNOW, R., O'CONNOR, B., JURAFSKY, D., AND NG, A. Y. 2008. Cheap and Fast—But is it Good?: Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the Thirteenth Conference on Empirical Methods in Natural Language Processing*, M. Lapata and H. T. Ng, Eds. Association for Computational Linguistics, Honolulu, Hawaii, 254–263.

SU, J. 2007. Probability Estimation Trees in Weka? Wekalist mailing list. `https://list.scms.waikato.ac.nz/pipermail/wekalist/2007-September/011343.html`.

TAX, D. 2001. One-Class Classification. Ph.D. thesis, Delft University of Technology, Delft, Netherlands.

WAN, S., DRAS, M., DALE, R., AND PARIS, C. 2006. Using Dependency-Based Features to Take the "Para-Farce" out of Paraphrase. In *Proceedings of the Fourth Australasian Language Technology Workshop*, L. Cavedon and I. Zukerman, Eds. Australasian Language Technology Association, Sydney, Australia, 131–138.

ZHAO, S., LAN, X., LIU, T., AND LI, S. 2009. Application-Driven Statistical Paraphrase Generation. In *Proceedings of the Forty-Seventh Annual Meeting of the Association for Computational Linguistics and the Fourth International Joint Conference on Natural Language Processing of the AFNLP*, K.-Y. Su, J. Su, and J. Wiebe, Eds. Association for Computer Linguistics, Suntec, Singapore, 834–842.

Received March 2011; revised August 2011; accepted November 2011