Detecting short passages of similar text in large document collections

Caroline Lyon, James Malcolm and **Bob Dickerson**
Department of Computer Science, University of Hertfordshire,
Hatfield, Hertfordshire, AL10 9AB, UK
{c.m.lyon,j.a.malcolm,r.g.dickerson}@herts.ac.uk

Abstract

This paper presents a statistical method for fingerprinting text. In a large collection of independently written documents each text is associated with a fingerprint which should be different from all the others. If fingerprints are too close then it is suspected that passages of copied or similar text occur in two documents. Our method exploits the characteristic distribution of word trigrams, and measures to determine similarity are based on set theoretic principles. The system was developed using a corpus of broadcast news reports and has been successfully used to detect plagiarism in students' work. It can find small sections that are similar as well as those that are identical. The method is very simple and effective, but seems not to have been used before

1 Introduction

Lexical patterns in text can be automatically captured and used to give a fingerprint to a piece of writing. Given a set of documents we can detect similar passages in any two documents by comparing their fingerprints and seeing if they are too close. We introduce a simple but novel system that extends the scope of textual pattern analysis, and applies it to copy detection tasks.

This tool has been used to detect plagiarism in scripts submitted by large classes of students. However, the method could be applied to other purposes, such as finding relationships between large numbers of documents in order to detect copyright infringements.

This work was developed with a corpus of 335 TV news reports. This is a good corpus to use since it contains examples of varying degrees of similarity, when the same issues are addressed in consecutive reports. This corpus is just under 1 million words long – see Table 3. We also make use of a smaller corpus, the Federalist Papers, of 85 texts, about 183,000 words, which has been investigated extensively (Hamilton et al., 1787 1788), to establish benchmarks. We describe how the plagiarism detector works on students' work, when similar but usually not identical passages a few sentences long can be detected.

Our tool will flag pairs of documents that contain similar passages. However, it can be the case that a student correctly quotes and cites source material. This can only be determined by inspection.

The rest of this paper is organised in the following way. Section 2 introduces the principles underlying our method, and discusses why word trigrams are suitable features to extract. Section 3 looks at the context of our work from the theoretical point of view, and gives an outline of the metrics that will be employed. Section 4 looks at related work in this field and Section 5 describes the domain in which we develop our prototype. Sections 6 and 7 describe the experiments undertaken and their results. Section 8 concludes the paper.

2 Principle of fingerprint extraction

A fundamental issue in language modelling for Automated Speech Recognition is the sparse data problem (Ney et al., 1997, page 176), (Gibbon et al., 1997, page 248). However, this phenomenon can be turned on its head and put to good use to fingerprint text.

The principle underlying our system is that the identifying fingerprint associated with a piece of text is based on a large number of small, easily extracted lexical features: word trigrams. Each text is converted into the set of overlapping 3 word sequences of which it is composed.

When we consider separate texts on the same subject there will be certain common words, bigrams and trigrams. Compound noun phrases provide typical examples. However, the phenomenon we exploit is the fact that common trigrams constitute a very small proportion of all the trigrams derived from independent texts – see the examples in Table 1 and Table 2.

Consider the well documented distribution of single words in English and other languages (Shannon, 1951; Manning and Schutze, 1999). We find a characteristic Zipfian distribution in which a small number of words are used very often, but a significant number are used rarely. In the Brown corpus of

A Classrooms have become inoculation centres as health workers try to stop the spread of the disease. More than *1,700 pupils and staff* were injected today to combat what's been described as a *public health emergency.*
B This morning children were queuing for injections not lessons at the school at the centre of the outbreak. Health teams have begun immunising *1,700 pupils and staff* in an attempt to stop any further cases of meningitis and bring this *public health emergency* under control.

Table 1: Examples of news text, from independent reports on the same subject. Though the semantic content is similar there are only 3 matching trigrams out of 33 in A and 43 in B. (Trigrams overlap, so 4 consecutive matching words produce 2 matching trigrams).

C *There's a lot of pressure* put *on people in* their *various capacities and if you* suddenly *find there are pressures* coming on you *that make it impossible to do your job . . .*
D *There's a lot of pressure on people in various capacities, and if you find there are pressures that make it impossible to do your job . . .*

Table 2: Examples of similar news reports where a common source is suspected. Here there are 15 matching trigrams out of 29 in C, and 23 in D. If we take 4-grams there are only 10 matches, and for 5-grams only 6 matches.

about 1 million words 40% of the word forms occur only once (Kupiec, 1992). This distribution of words is an empirical observation, but can also be understood on theoretical grounds given certain assumptions on English language production (Bell et al., 1990, chapter 4).

This distinctive distribution of words is more pronounced for word bigrams and even more pronounced for trigrams. If the probability of a word occuring is low, the probabilty of that word occurring in conjunction with others is lower still. Thus, it is usually the case that most trigrams turning up in new texts never occurred in large training corpora, even when documents are on the same subject and sometimes by the same author.

Table 4 shows the high percentage of trigrams that occur only once in the TV News corpus. Gibbon et al. give figures for large corpora from the Wall Street Journal (Gibbon et al., 1997). This corpus is in a well defined, limited domain so we might expect recurrent lexical features to become quite common as the corpus size increased. However, as the table shows, even after 38 million words, 77% of trigrams have only occurred once. In any particular article, the majority of trigrams will probably belong to that article alone. The set of trigrams derived from any one article is a distinguishing feature set.

We investigated the use of n-grams as lexical features for various n. Single words and word pairs had inadequte distinguishing power, whereas trigrams are effective, as demonstrated later.

For $n > 3$ we reduce the sensitivity of the tool, its ability to detect similar as well as exactly copied text. This is illustrated in the sample texts shown in Table 2: there are 15 matching trigrams (52% of 29 trigrams in text C), 10 matching 4-grams (36% of 28 4-grams in text C) and 6 matching 5-grams (22% of 27 5-grams in text C).

3 The context of statistical pattern recognition

Recent overviews of work in this vast field include the special issue of the journal "Pattern Analysis and Machine Intelligence", January 2000 (Jain et al., 2000). A dominant approach to pattern recognition is to take the data that is being analysed and

TV News corpus	
Number of texts	335
Total number of words	985,316
Maximum file size in words	5090
Number of files with < 1000 words	6
Average file size in words	2941
Average number of distinct trigrams per file	2818
Average % of singleton trigrams within each file	96%

Table 3: Statistics from the TV News corpus used in this work

Source	Corpus size in words	Distinct trigrams	Singleton trigrams	% of trigrams that are singletons
TV News	985,316	718,953	614,172	85%
Federalist Papers	183,372	135,830	118,842	87%
WSJ	972,868 4,513,716 38, 532,517	648,482 2,420,168 14,096,109	556,185 1,990,507 10,907,373	86% 82% 77%

Table 4: Statistics from the TV News corpus, the Federalist Papers and from the Wall Street Journal corpora (Gibbon et al., 1997, page 258)

abstract significant features from it. The resultant features are lined up in a feature vector that characterises the data. Feature vectors are then processed to achieve a given objective, such as a classifying task. Associated with this approach is the need to abstract appropriate features, get their associated weights, and find the most effective methods of processing them. The method has been successfully used in many sound and image processing tasks.

However, this approach only works within certain limits. There are relationships between the number of elements of the feature vector, and the amount of data that needs to be used to set the parameters of the system: a large number of features need a very large amount of training data. For instance, in the neural network branch of statistical pattern recognition, guidelines on the number of training examples needed for a certain size of feature vector typically quote a minimum ratio of 10 to 1 (Jain et al., 2000, page 11), (Bishop, 1995, page 380).

In language processing some tasks can be addressed within the limitations of the main stream statistical pattern recognition approach. For instance, in parsing, an indefinite number of words may be mapped onto a limited number of parts-of-speech. (Lyon and Frank, 1997). In detecting semantic similarities in texts linguistic indicators can be extracted for a feature vector (Hatzivassiloglou et al., 1999). However, there are other tasks where we cannot abstract out a reduced number of significant features without losing necessary information. When we need to use lexical information the number of words in unrestricted natural language will usually be prohibitive. The standard pattern recognition approach of processing linear feature vectors cannot be used, and document processing with large vocabulary texts is cited as a problem remaining to be addressed (Jain et al., 2000, page 58).

For copy detection we need to use lexical information. Most trigrams are *prima facie* of comparable value, and we have not abstracted out a reduced set before we started processing. Therefore, we take the different approach of classifying text using using set theoretic concepts. Instead of lining up features in a linear vector, large numbers of mini-features are grouped together in a set. Using this approach we do not weight different mini-features, nor model dependencies between them. However, we can use a much greater range of lexical information.

In spite of the advances made in statistical pattern recognition in the last 25 years comparatively little attention has been paid to this approach.

3.1 The comparison of documents based on set theoretic concepts

Our objective is to compare two documents and classify them as either "independent" or else "similar". The comparison can take place in two modes: first, we may compare items of comparable length, and in this case we will be looking for "resemblance" between texts. Secondly, a piece of text can be compared with a large body of material which could have been used as a source. In this case the texts will be of unequal size, and we will be looking for "containment". If a significant portion of the smaller text is contained within the larger, then it indicates that material has been lifted from the suspected source.

Using a method based on set theoretic concepts, we first transform each piece of text to a set of trigrams. Then, for the two documents being considered, the sets of trigrams are compared.

The measures we use come from work by Broder (Broder, 1998). The concept of resemblance, informally, is the number of matches between the elements of two sets of trigrams, scaled by joint set size.

Let S(A) and S(B) be the set of trigrams from documents A and B respectively. Let R(A,B) be the resemblance between A and B[1].

$$R = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|} \tag{1}$$

For the concept of containment, C(A,B), suppose we are measuring the extent to which set B is contained in set A. Set A might be derived from concatenated potential source material from the web, a large set. Set B might be derived from a single student essay, a small set. Informally, containment is the number of matches between the elements of trigram sets from A and B, scaled by the size of set B; in other words, the proportion of trigrams in B that are also in A.

$$C = \frac{|S(A) \cap S(B)|}{|S(B)|} \tag{2}$$

4 Related work

Lexical statistics have been widely used in speech and language processing for many years (Kukich, 1992; Gibbon et al., 1997). However, we can find no record of their use in the way we propose.

4.1 Copy detection in text

Other approaches to the copy detection task include methods based on the concept of searching for matching strings, typically much longer than trigrams.

A well known system is SCAM (Stanford Copy Analysis Mechanism) (Shivakumar and Garcia-Molina, 1996), which has two web based objectives. The first is to check the internet for copyright infringements, the second to filter out duplicates and near duplicates in information retrieval. To do this they process "several tens of millions of web pages". On this large scale, evaluation inevitably presents problems, and is very limited.

Though the scale of their current work puts their objectives into a different class from our own, their work is based on a similar concept: fingerprinting text by extracting sets of "chunks". A chunk is a string of words, based on non-overlapping sequences of various length and distribution. A sampling mechanism is chosen to reduce the number of fingerprints stored. Using non-overlapping chunks reduces storage requirements, but loses much useful information. If two chunks match in all except one word they will not match at all. Problems can arise when similar chunks get out of phase and appear not to match.

Broder (Broder, 1998) has addressed the task of clustering 30,000,000 documents into groups of those that closely resembled each other. His approach was based on the concept of matching "shingles", where a shingle is a sequence of words. The length of the shingle has to be determined. To reduce storage demands, only some shingles were selected, by a sampling process. The similarity measures: resemblance and containment were used as defined above.

A similar approach is employed by Heintze (N.Heintze, 1996), who has developed a system for analysing up to 1 million documents. He uses character strings rather than word strings as the basis for a fingerprint, and quotes effective lengths of 30 - 45 characters. His work focuses on methods of selection to produce a reduced size fingerprint from the full set. A web-based prototype is available.

These systems address copy detection on a very large scale, from 1 million to tens of millions of documents. As with all problems on this scale, evalu-

[1] also known as the Jaccard coefficient, and used in feature vector analysis (Manning and Schutze, 1999, page 299).

ation presents difficulties. For instance, the SCAM system was evaluated just by classifying a sample of 50 texts on a subjective basis (Shivakumar and Garcia-Molina, 1996, section 4.4).

In all these methods longer chunks, shingles or character strings are a less sensitive tool for detecting similar texts rather than exact copies, as illustrated in Table 2. The insertion, deletion or substitution of a small number of words can undermine the matching process. Using our trigram method underlying similarities can be better detected, and cannot be camouflaged by superficial variations in the text.

4.2 Other related work

Hatzivassiloglou et al. (Hatzivassiloglou et al., 1999) have developed a method for detecting semantically similar texts by extracting a heterogenous collection of morphological, syntactic and semantic features and then processing the resultant feature vector with a rule induction system. This is an example of the traditional feature vector approach described in section 3.

This approach has also been used for detecting similar sections of code, or clones, in very large programs. A successful example is a neural processor that detects clones in 26 million lines of code, the product of a telecommunications company (Barton et al., 1995). In this case the feature vector is a heterogenous collection of items, derived from information on keyword frequency counts, numbers of parameters, formatting and other factors. The limited number of features makes a neural approach appropriate.

Phelps and Wilensky (Phelps and Wilensky, 2000) have shown that texts can be identified by very small signatures, no more than 5 words long. This is of use in producing robust URLs, so when a web page is moved it can still be traced if the URL is augmented with a content based signature. This however is a different task to copy detection: documents with different signatures can still contain copied material.

5 The TV News corpus

The principal corpus we used to develop our prototype was a set of 335 TV news reports, taken over periods from 1999 to 2000. Some statistics on the corpus are given in Table 3.

There were usually texts from 3 news programmes each day in our corpus. Often the same semantic content was described in different words, as illustrated in Table 1. Sometimes the language used was very similar, as illustrated in Table 2. Sometimes there was verbatim copying. As we ran our diagnostic tests over this corpus more examples than expected of copied text came to light.

Though there are clear instances of independent or similar text, there is no definitive objective boundary for borderline cases. Eventually, the classification of documents as similar or independent becomes a subjective judgement.

Where the boundary is set also depends on the purpose for which the tool is being used. More possible copies can be flagged if some false positives are tolerated. To avoid this, the threshold can be set higher so that there are no false positives, but some short similar passages may go undetected.

We decided that texts like those in Table 2 should count as sufficiently close to be flagged.

6 Experiments and results

Preprocessing

In all our experiments, some pre-processing of the text was done first. For words starting sentences the initial letter was decapitalised, providing it was not a proper noun. Punctuation was removed. Numbers were replaced with a symbol, so that typical word patterns like “inflation rose x percent” were preserved. Next, each text was transformed into its associated set of trigrams.

6.1 Experiments on Resemblance

We first compared each pair of files in the original corpus. All except 6 scores for resemblance were below 0.1. There was one aberrant result discussed below, where $R = 0.16$

The number of matches for programmes not on the same day was typically in the range 20-60, giving $R < 0.01$

The scores above 0.1 were all for news programmes for the same day. On inspection we found that these texts included sections copied verbatim, or very similar copies, varying from 150 to 250 words in length. In some cases several sections were copied.

The exceptional case where $R = 0.16$ indicated that news on 10.3.99 resembled a programme on 27.4.99. On inspection it turned out that an item 440 words long from the first date was re-broadcast verbatim in a programme 3080 words long on the second date, about 14% copied. This item, on environmental pollution, was not particularly topical.

On closer inspection we found that for pairs of texts where $R > 0.06$ there were copied sections of varying lengths, and varying degrees of similarity. This was taken as a preliminary threshold above which copying was indicated, but was subsequently revised downwards to $R > 0.03$.

6.2 Identifying known copying

A blind test on identifying known copying was done with 10 files, which were put aside by one experimenter. For each of them, half their contents were used to replace the text in another file in the main body of the corpus, then they were put back in the main corpus. The other experimenter then ran the

Number of files compared	335
Original files: For 6 pairs out of 55945 R > 0.1 *Maximum* R	R-max = 0.16
Doctored files: *Minimum* R between doctored files and counterpart files from which passages had been copied	R-min > 0.30

Table 5: Scores for resemblance, R, on original TV News corpus files compared to scores for doctored files.

Number of files compared	10
Maximum containment C between undoctored files and rest-of-corpus *Minimum* containment C between doctored files and rest-of-corpus	0.33 > 0.49

Table 6: Scores for containment, C, between 10 files from original TV News corpus and rest-of-corpus compared to scores for 10 doctored texts.

fingerprinting programs over the whole corpus, including the doctored files. They were very easily identified, see Table 5.

Similar tests were carried out using the containment metric. We compared single files to a large body of material that could be a source. The whole body of texts was concatenated, apart from 10 that were reserved to be assessed. These were taken one at a time and compared to the rest-of-corpus. Results in Table 6 show that there is a clear distinction beween the C scores for the doctored files, and the exceptional case, compared with the normal ones.

However, these experiments with doctored files only show that gross plagiarism can be very easily detected.

6.3 Establishing thresholds

The Federalist corpus

The Federalist Papers are texts from a completely different domain. They are the celebrated collection of 85 papers, written in 1787-1788, on the proposed American Constitution. They are easily accessible and have been extensively studied (Mosteller and Wallace, 1984). We considered 81 of the papers, written by Madison or Hamilton, totalling 183,372 words. The average paper length is 2,300 words.

In this corpus the same subjects are addressed repeatedly. The papers are all written by one of the 2 authors, under the same pseudonym.

The threshold for R

When we ran our prototype over these papers the maximum R score was 0.03. This confirmed our view that $R = 0.03$ was a reasonable level above which similar text would be found. If R is reduced, as in some of our investigations into students' work, copying is likely to be found, but some false positives may arise.

The threshold for C

We then investigated containment scores between some individual files and the concatenated rest-of-corpus. We took out 9 files spaced through the corpus to assess. They averaged 2,300 words each, and 162,655 words were left in the rest-of-corpus. The resulting scores for C, containment, ranged from $0.23 < C \leq 0.37$. All except one were below 0.34.

We conducted a similar experiment with files extracted from the News corpus, chosen so that no other news programmes on the same day were in the rest-of-corpus. Some of them were on consecutive days. News reports on the same day frequently treat the same subjects, and there are often copied sections embedded in them. Taking reports on different days reduces, but does not eliminate, this risk.

The results in Table 7 were obtained, showing $0.25 < C < 0.31$

Number of words in concatenated corpus	959772
Number of files to be assessed	9
Average number of words in each assessed file	2838
Range of C scores	$0.25 < C < 0.31$

Table 7: Results from experiment to establish base lines for thresholds when comparing one text to a large potential source

From this we propose a provisional C threshold of 0.31, but are aware that false positives may slip through.

7 Detecting plagiarism in students' work

The plagiarism detector has been used on students' work submitted electronically. The vast majority of scripts are clearly independently written or clearly flagged as potential copies. Once the assignments have been loaded the processing time is very quick: for instance, less than 30 seconds for 280 files The preprocessing has been simplified: all letters are converted to lower case, and non-alphabetic characters are ignored.

One example of its use was to investigate 124 reports. 103 students submitted their reports in a single text file, averaging 4000 words, 200 sentences. 21 submitted their reports in sets of smaller HTML files. Each of these averaged 400 words.

This is a robust system and handles HTML as well as ordinary text: non-alphabetic symbols are ignored, and if parts of embedded HTML commands are left as words this does not undermine the system.

14 cases of matching sections of text were found. Two of these were significant, several paragraphs long. The others were matching sentences sprinkled through the scripts. It seems that some students cut and paste fragments from the web and from each other. Matching passages were mostly similar rather than identical, with occasional words and phrases deleted, substituted or inserted. There is of course no way of telling the direction of the copying between students.

Of the two significant cases one had $R = 0.19$ with 1400 matching trigrams in files of 4000 and 5000 words. There were 6 similar paragraphs. The other had $R = 0.19$ with 805 matching trigrams in files of 2000 and 3000 words. There were 8 similar paragraphs.

Of the other cases with 4 to 10 matching sentences, R ranged from 0.025 to 0.1. When the threshold for R is reduced below 0.03 some false positives arise. In this assignment there could be a small number of matching trigrams without any plagiarism - for instance from the table of contents. However, taking the threshold R at 0.03 there were no cases where we did not find matching passages in scripts where R was above the threshold.

In another case 54 reports were analysed, averaging 2,800 words each. The highest reading for R was 0.027. When we looked at 4 cases where $R > 0.023$ we found the same 3 matching sentences each time: the students had quoted the assignment brief.

An interface is being developed for this plagiarism detection tool. It will display the names of pairs of texts ranked in order, those most likely to include similar text coming first. Then the contents of the files will be displayed if required, with similar passages of text highlighted, side by side. At this point a subjective decision can be made on whether work has been copied, or is suspiciously similar.

8 Conclusion

In this work we have introduced another way of using lexical information for language processing, and produced an application. The method is very simple and effective, but seems not to have been used before.

The mainstream of textual pattern analysis has been restricted by the focus on linear feature vectors as described in section 3, and in some examples in section 4. We have used a different approach, based on set theoretic principles.

Our system has been developed using material with subtle similarities so the distinction between independent and non-independent text was hard to determine. This makes it a good basis on which to develop a robust tool.

The system has been evaluated to the extent that it is of proven effectiveness in detecting copied material in students' work. Similar passages a few sentences long have been found in files of several thousand words. Passages that are similar but not identical have been identified, so slight editing does not mask plagiarism.

Other work in this field (section 4) is much more ambitious than ours, addressing the comparison of many millions of web pages. However, evaluation is difficult for these large scale tasks. Our smaller system demonstrably works, and using our trigram based fingerprint we are more likely to detect similar passages that are not identical.

However, we need to compare our system with other methods, and the whole issue of evaluation of large scale language processing tools needs much more attention. This will be integrated into future work which will focus on scaling up our system, in particular to use material from the web.

We also plan to investigate whether this approach can be extended to comparing programs: in this case non-alphanumeric symbols would be candidates for fingerprinting. We will again start by investigating data, in particular the distribution of symbol tuples.

Our work illustrates how empirical investigations can lead to a new intepretation of data. We can claim that we have carried out the well known injunction (Wittgenstein, 1945, pages 31,47) on acquiring knowledge of language "don't think, but look! ... problems are solved ... by arranging what we have always known".

References

P Barton, N Davey, R Frank, and D Tansley. 1995. Dynamic competitive learning applied to the clone detection problem. In *International Workshop on Applications of Neural Networks to Telecommunications*, Stockholm.

T C Bell, J G Cleary, and I H Witten. 1990. *Text Compression.* Prentice Hall.

C M Bishop. 1995. *Neural Networks for Pattern Recognition.* OUP.

A Z Broder. 1998. On the resemblance and containment of documents. In *Compression and Complexity of Sequences, IEEE Computer Society.*

D Gibbon, R Moore, and R Winski. 1997. *Handbook of Standards and Resources for Spoken Language Systems.* Mouton de Gruyter.

A Hamilton, J Madison, and J Jay, 1787-1788. *The Federalist Papers.* www.mcs.net/ knautzr/fed/fedi.htm.

V Hatzivassiloglou, J Klavans, and E Eskin. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proc. of EMNLP.*

A K Jain, R P W Duin, and J Mao. 2000. Statistical pattern recognition: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22, 1.

K Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, December.

J Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language.*

C Lyon and R Frank. 1997. Using Single Layer Networks for Discrete, Sequential Data: an Example from Natural Language Processing. *Neural Computing Applications*, 5 (4).

C D Manning and H Schutze. 1999. *Foundations of Statistical Natural Language Processing.* MIT.

F Mosteller and D L Wallace. 1984. *Applied Bayesian and Classical Inference: The case of the Federalist papers.* Springer-Verlag.

H Ney, S Martin, and F Wessel. 1997. Statistical language modelling using leaving-one-out. In S Young and G Bloothooft, editors, *Corpus Based Methods in Language and Speech Processing.* Kluwer Academic Publishers.

N.Heintze, 1996. *Scalable Document Fingerprinting.* Bell Laboratories, www.cs.cmu.edu/afs/cs/user/nch/www/koala/.

Thomas Phelps and Robert Wilensky, 2000. *Robust Hyperlinks: Cheap, Everywhere, Now.* Proc. of Digital Documents and Electronic Publishing, www.cs.berkeley.edu/ phelps/Robust/papers.html.

C E Shannon. 1951. Prediction and Entropy of Printed English. *Bell System Technical Journal*, pages 50–64.

N Shivakumar and H Garcia-Molina. 1996. Building a scalable and accurate copy detection mechanism. In *Proc. of 3rd International Conference on Theory and Practice of Digital Libraries.*

L Wittgenstein. 1945. *Philosphical Investigations.* Blackwell, 1992.