Distributed Similarity and Plagiarism Search

Máté Pataki
Department of Distributed Systems
Computer and Automation Research Institute of the Hungarian Academy of Sciences
Mate.Pataki@sztaki.hu

Abstract. This paper describes the different approaches of plagiarism search, the methods used by the KOPI Online Plagiarism Search and Information Portal and, shows a distributed approach for building a plagiarism search system. This architecture adds scalability to the system, by allowing placing an arbitrary number of identical components into it. To reduce network traffic and enable secure transfer of the documents between the portal and the document servers a new method of communication is introduced.

Keywords: plagiarism search, similarity, distributed, hash

1 Introduction

Access restriction and detection of plagiarism are two ways of protection for documents. It is really difficult to achieve high quality protection which on the other hand makes the access to the documents harder for everyone. This results in less people reading and referring to these sources, and the ones who really are determined to copy these documents will always find a way to by-pass the protection. Detection on the other hand does not restrict the access to the documents, they can be freely distributed and used, but if someone uses that particular text, or part of it, the plagiarism detection system - which has a copy of all documents - will determine the original source of the text. With such a system in use the risk of being caught for plagiarism will be too high for most people and so the document is protected.

Similarity search and plagiarism search among documents mean almost the same, a tiny difference between the two is that in the first case both the original author and the quantity of the copied text are stated, while in the second one or both of these are missing. It cannot be automatically determined whether something is plagiarism, similarity or quotation; however, a plagiarism search system can help to find the possible original sources to be examined.

The Department of Distributed Systems of the Computer and Automation Research Institute of the Hungarian Academy of Sciences [1] has developed a plagiarism search portal with the following goals in mind:

- Fighting and reducing plagiarism at schools and universities
- Helping conference organizers find similar work, or determine the genuinity of a given article (compared for example to other articles of the same author)
- Finding the original source of a document

The KOPI Online Plagiarism Search and Information Portal [2] is a free service, provided by the department. It is built on a portal engine with static pages (about plagiarism, laws, netiquette, FAQ, help) and forum pages for the users to discuss ideas concerning plagiarism. Users can also upload documents, which then become part of the system's database. When a search for similar documents is requested, this database - consisting of other users' documents - can also be searched to detect overlapping documents.

One could ask whether KOPI is different from the other systems, or is it yet another plagiarism search tool. The existing systems can be categorized as follows:

- Many systems use watermarks or checksums for the whole text. In most cases watermarks can be easily and automatically removed. Checksums are not good at detecting smaller overlapping parts and can be easily deceived with some small alterations in the text.
- There are programs that generate a test from the document, where a given number of words are removed, which then have to be filled in by the author [3]. This solution could work at a school or university but there the student is accused of plagiarism and has to fill in the test. This takes a lot of time from both the student and the professor, and more importantly the risk of false accusations may be too high.
- Meta systems that use search engines (like Google) for plagiarism search [4], have good results in detecting works copied from the Internet, yet in most cases the sources cannot be found on the Internet. A few people put their homework or theses on the Web, and the access to most digital libraries and collections are also restricted.
- A totally different approach for plagiarism search is authorship attribution and identification. However, it has two big disadvantages, the first is that in most cases the algorithm used for linguistic analysis is language dependent so it has to be developed for each language used. The other problem is that they need more texts from the same author, which in many cases are not available.
- There are also a couple of commercial systems, like Plagiarism Finder [5] and EVE Plagiarism Detection System [6], but because of their kind, their working mechanisms and the algorithms they use are unknown (security by obscurity). This makes it hard to rely on them, as it is not known how they can be deceived and what conditions the documents must fulfill to

be suitable for plagiarism search. Moreover, many people and institutions cannot afford paying for such services.

KOPI Plagiarism Search Portal, in contrast, uses a language independent algorithm that was published, and even with this knowledge is hard to deceive. This service is provided by our department free of charge for everybody. The development of the portal begun in early 2003 and it is open to the public since the end of May 2004. The database of the documents in the system gets larger with the increasing number of users, and the more documents the system has, the more effective the detection will be.

2 Similarity Search

The similarity search algorithm developed for this portal requires six basic steps:

1. Getting the documents
2. Converting the document into plain text
3. Chunking the document
4. Fingerprinting the chunks
5. Uploading the fingerprints into a database
6. Database query for documents with same fingerprints

The following sections will describe each step, while the next chapter will explain how these parts are put together to make up the whole system.

2.1 Getting the Documents

There are many document collections, databases and documents of different institutions which could become part of such a plagiarism search system. The most important sources are the documents uploaded by the users themselves. Considering the quantity of homework and theses written each year universities can serve as huge sources of documents. Moreover, digital libraries can also be possible partners in providing texts as they possess a large quantity of documents to be protected against plagiarism, yet with free access to anyone. Finally, the biggest source of digital texts is the Internet, which can be harvested by a robot.

2.2 Converting the Document to Plain Text

As KOPI does not use watermarking algorithms hidden in the formatting all formatting can be removed from the text. This is an important step because it allows the system to find overlapping between two texts with different formatting. The converter of KOPI is a complete subsystem in itself and accepts the following file-types: rtf, doc, pdf, html and zip. The latter is extracted and

the files in the archive are added to the system. After conversion, the language of the document is also determined, this makes it possible to put documents written in different languages into different databases.

There are a lot of document types which all need different programs to be accessed. To convert one to the other is even a bigger challenge, as the available converters differ in knowledge and system requirements. The best converter we could find for PDF files was written for both Windows and Linux platforms, but the easiest and safest way of converting DOC and RTF files is with a small program which calls Microsoft Words's own built in converter for this task (via the OLE interface) [7]. The most suitable converter for different Wiki files was written for Linux.

The above convinced us to write a distributed converter system. In this system there are servers, each with different capabilities, one of them can covert DOC and RTF files to the other and to TXT, the other can do the same while capable of converting PDF files as well, the third can just convert Wiki files to HTML and TXT and so on.

Each client that needs to use this distributed converter system has a list of servers, which it can use, it regularly asks all of them for their conversion capabilities, and when a document has to be converted it uses a randomly chosen server from its list of capable servers. This randomity is a load balancing, as in our case the conversion of DOC to TXT is done by some of the desktop computers of our colleagues where Microsoft Word is installed.

2.3 Chunking the Document

To be able to find smaller overlapping the text needs to be chunked into smaller parts. For this purpose KOPI uses a new method [8] which is the mixture of word chunking and overlapping word chunking [9], [10], [11]. Word chunking is the easiest way to chunk a text, with a parameter n at every n^{th} word a new chunk begins.

However, word chunking has a big disadvantage: if one inserts or deletes even a single word from the beginning of the text, then all chunks are altered. This so called phase shift problem is solved by the overlapping word chunking, where a chunk begins at each word, and so every possible n word piece is generated from the text.

As seen in the example (Fig. 4.), it generates n times as many chunks as the word chunking, yet the insertion or deletion of a word causes only local differences, the following chunks of the text remain unchanged.

In the KOPI portal the database is filled with word chunked documents (Fig. 3.), this results in a small database and a quicker search. To get rid of the phase shift problem the documents compared to the database are chunked with the same parameter n, but with overlapping word chunking (Fig. 4.). For example, by inserting one word at the beginning the same text as above results in the following chunks:

These five chunks (Fig. 5.) are uploaded to the database. When comparing to the other document, chunked with overlapping word chunking (Fig. 4.), the

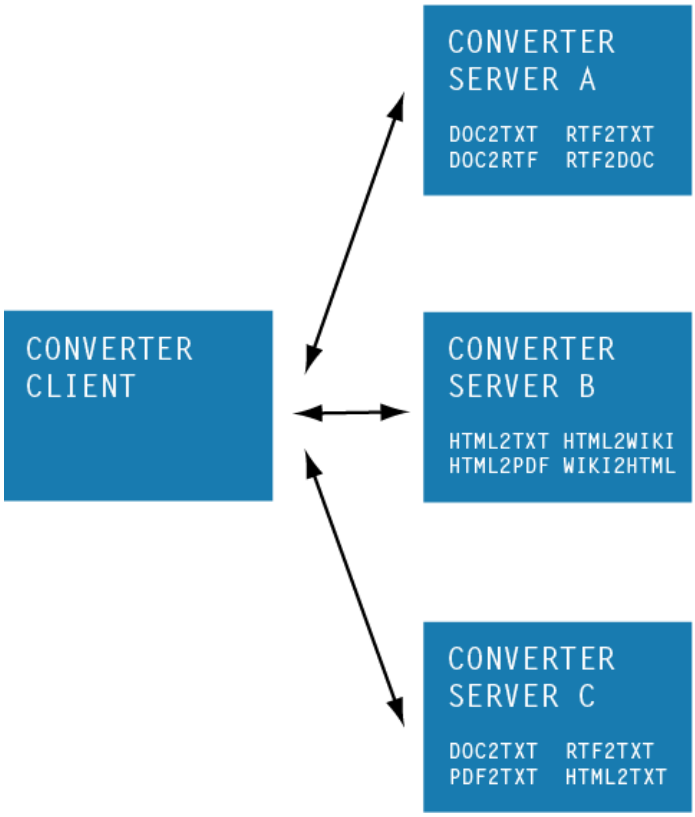

Figure 1: Converter client and tree servers with different but overlapping capabilities

first chunk will be different from all other chunks. The next chunk, and all other ones as well, exists in the overlapping word chunked version (chunk number 10, 20, 30 and 40).

With this new method, the database is much smaller, roughly nth the size of the one with overlapping word chunking. When a word is altered in the text, one chunk will differ from the original, when using only overlapping word chunking n chunks would differ from n times as many. This means that for similarity search both are equally good, but the newly developed one is much faster and requires less storage capacity.

2.4 Fingerprinting the Chunks

Working with texts is much slower than working with numbers, and that is why the chunks are fingerprinted using 32 bits of the MD5 sum of the chunks. This allows a much smaller database and as our research has shown does not end up with too many false positive results (different text but same MD5 sum) [12].

```
   The purpose of the KOPI portal is to detect plagiarism and to
protect documents from being copied, as the advance of computer
sciences not only has made it easier to create written documents
but also has made it extremely simple to copy and plagiarize whole
documents or parts of documents.
```

Figure 2: Original text

```
 1. the purpose of the kopi portal is to detect plagiarism
 2. and to protect documents from being copied as the advance
 3. of computer sciences not only has made it easier to
 4. create written documents but also has made it extremely simple
 5. to copy and plagiarize whole documents or parts of documents
```

Figure 3: A text chunked with word chunking (n=10)

Another big advantage of fingerprinting is that the fingerprints can be freely transferred through the network and the original documents cannot be reconstructed from them. This is an important issue to be taken into consideration when one plans to have a distributed search engine.

2.5 Uploading the Fingerprints Into a Database

As mentioned above the fingerprints are uploaded into a database. The table where the fingerprints are consists of only two columns, namely the fingerprint and the documents id [9], [13], [14]. The position of the chunks in the document is unimportant, because the same text in different documents could be anywhere [15]. It is also interesting to note that even if a chunk occurs more than once in the text, it will be stored only once, not forced to save the number

```
 1. the purpose of the kopi portal is to detect plagiarism
 2. purpose of the kopi portal is to detect plagiarism and
 3. of the kopi portal is to detect plagiarism and to
 4. the kopi portal is to detect plagiarism and to protect
 5. kopi portal is to detect plagiarism and to protect documents
 6. portal is to detect plagiarism and to protect documents from
...
40 simple to copy and plagiarize whole documents or parts of
41 to copy and plagiarize whole documents or parts of documents
```

Figure 4: The same text chunked with overlapping word chunking (n=10)

1. the main purpose of the kopi portal is to detect
2. plagiarism and to protect documents from being copied as the
3. advance of computer sciences not only has made it easier
4. to create written documents but also has made it extremely
5. simple to copy and plagiarize whole documents or parts of
6. ~~documents~~ (the last chunk is discarded if incomplete)

Figure 5: One word inserted into the text and chunked with word chunking (n=10)

of occurrences in the database makes the system faster, and the result will be almost the same even in these rare occasions.

3 Querying the Database for Documents with Same Fingerprints

In this last step of the process the number of identical fingerprints of the two documents are considered as the quantity of overlapping. Accidental one chunk overlapping between documents can occur even without copying, and there are also rare occasions of the earlier mentioned false positive results from MD5 algorithm, therefore one common fingerprint is not considered as overlapping at all. When the two documents have two or more chunks in common, the result is shown as percent of the documents and also in the number of words (Fig. 6.).

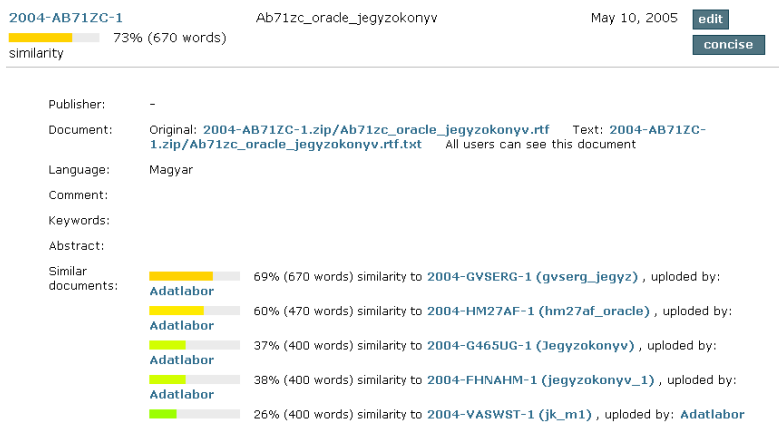

Figure 6: Visualization of the results of a similarity search in KOPI

4 The System

The KOPI Portal can be considered as an interface for the users to upload their documents and start search jobs. The "real work", chunking, fingerprinting and database queries, is done by the document servers responsible for different sources of documents. In the current system there are two servers of that kind, one includes the files uploaded by the users, the other is a collection of documents harvested from the Internet. We plan to have at least two more servers, one for a digital library and an other for university theses, but there is no limitation on the number of servers in the system architecture.

The document servers can be connected to one or more portals, in our case there is only one portal. Its architecture is shown in the figure below.

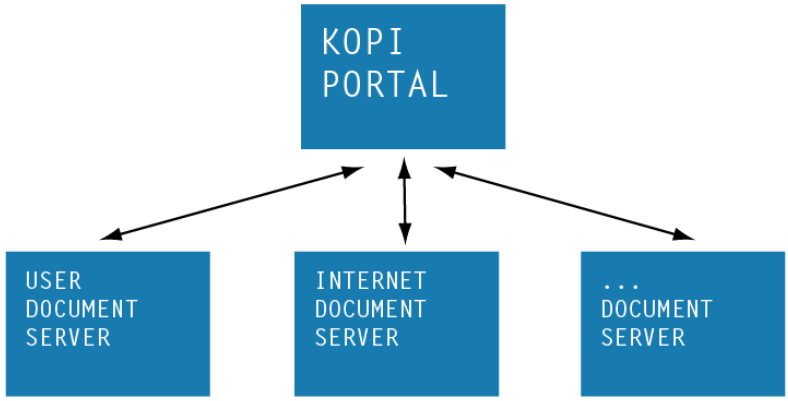

Figure 7: KOPI Portal and the document servers responsible for different document sources

The connection between the different parts of the KOPI system is done by SOAP protocol [16]. This is a W3C recommendation, which has the advantage of being available for almost any programming languages. The current system and subsystems are written in PHP, but with SOAP protocol none of the parts is bound to this or any other programming language, they could be replaced any time without changing the parts connected to them.

5 Document Servers

All document servers consist of the parts described in the previous chapter and are connected to one or more databases and to the converter server (which itself is also a distributed system).

The connection to the KOPI portal is initiated by the document server, this allows it to work on one job at the same time but for more portals in a row. The document server regularly connects to the portal and receives fingerprints of documents to be searched for in its database.

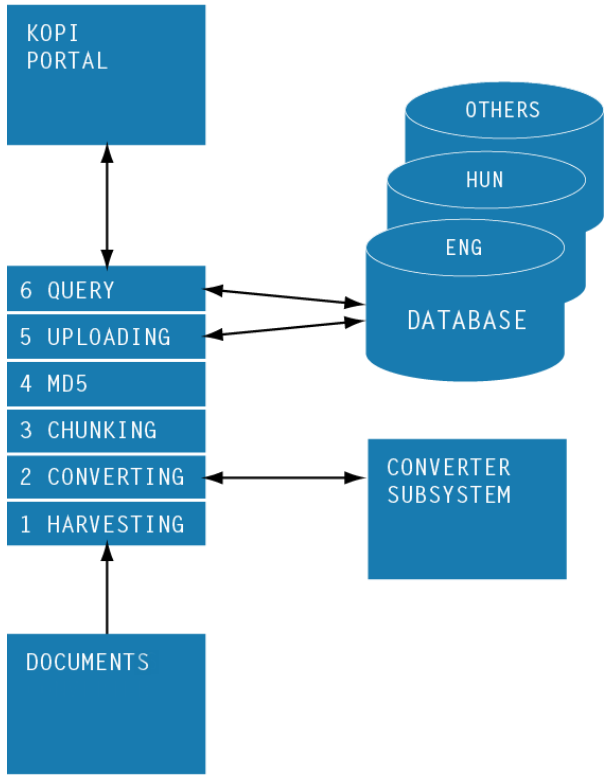

Figure 8: A document server

6 Future Plans

Many institutions would like to protect their documents and as a result they cannot and will not upload them to a system outside their network. KOPI could be a solution for them as well. If an institution installs and runs its own version of KOPI, they could easily upload all their documents to it and search for plagiarism there. These standalone versions could then be connected to each other by a special document server which would search in other portals' databases and return the results.

These document servers would transfer only the fingerprints of the documents to be compared to each other. It is safe to do so, because even if someone intercepts this information, the document cannot be restored from it.

With this new way of communication between institutions we can hopefully achieve our goal in fighting and reducing plagiarism and can also protect the original authors by giving them credit for their work.

Acknowledgement

The author would like to express his thanks to László Kovács for his support as a scientific advisor. This paper was created with financial support from the Visegrad International Fund and will be published also at www.ikaros.cz, an e-journal on information society.

References

[1] "Department of distributed systems of the hungarian academy of sciences," http://dsd.sztaki.hu.

[2] "Kopi online plagiarism search and information portal," http://kopi.sztaki.hu.

[3] "Glatt plagiarism screening program," http://www.plagiarism.com.

[4] "Plagiarism search v 1.0.0," http://baltic.cse.msu.edu/heynige1/Search.

[5] "Plagiarism finder," http://www.m4-software.de/en-index.htm.

[6] "Eve plagiarism detection system," http://www.canexus.com.

[7] REI@WDIC.ORG, "Word ole document converter,"

[8] M. Krisztián, F. Raphael, Z. Arkady, H. Gábor, and P. Máté, "Comparison of overlap detection techniques," Computer Science ICCS 2002 International Conference, April 21-24 2002.

[9] B.-Y. Ricardo and R.-N. Berthier in Modern Information Retrieval, Addison Wesley, 1999.

[10] S. Narayanan and G.-M. Hector, "Scam: A copy detection mechanism for digital documents," Proceedings of 2nd International Conference in Theory and Practice of Digital Libraries (DL'95), 1995.

[11] S. Narayanan and G.-M. Hector, "Building a scalable and accurate copy detection mechanism," 1996.

[12] P. Máté, "Darabolási technikák és másolatkeresés," in Szöveges dokumentumok darabolása és tömörítése hash-kódolással, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2002.

[13] S. G., "The state of retrieval system evaluation," 1992.

[14] B. Sergey, D. James, and G.-M. Hector, "Copy detection mechanism for digital documents," in Department of Computer Science, Stanford, 1999.

[15] H. Gábor, "Adattárolási és adatszervezési kérdések," in Szöveges dokumentumok darabolása és tömörítése hash-kódolással, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2002.

[16] "Simple object access protocol," http://www.w3.org/TR/soap.