A Graphical and Convenient Tool for Document Comparison and Visualization

C.M.X. Benjamin, W.L. Woon, K.S.D. Wong
Malaysia University of Science and Technology
mitm4bc1@must.edu.my

Abstract

Although there are a number of tools available for analyzing collections of documents, little work has been done in visualizing differences and elucidating similarity trends among documents to detect plagiarism. Therefore, a convenient and powerful document comparison and visualization tool is introduced. The software tool is developed with GIMP toolkit (GTK) to provide a graphical user interface (GUI) for an effortless navigation to process multiple documents, and allow the user to visualize document similarities in an intuitive way via Kohonen Maps. Moreover, the document analyzer tool incorporates Customized Term Weighting Scheme (CSTW) to capture category and semantic information for document classification.

I. INTRODUCTION

Document comparison, clustering, and visualization have been studied in the field of information retrieval (IR) for some time. The development of fast algorithms for these three components are critical in helping a user locate relevant materials among the retrieved documents as quickly as possible. A number of document organization approaches have been developed over the recent years [1-3].

Information overload and plagiarism on the Internet are both well recognized problem remain unresolved. With huge amounts of information connected to the Internet, efficient and effective discovery of resource and knowledge for IR methods has become an imminent research issue. Moreover, with the advent of new technology coupled with rapid multimedia development, digital documents can be now easily copied. Detecting the similarities between a document and a plagiarized version of a document would be difficult particularly when two documents are partly overlapped with some portion of text are (almost) similar while the rest being totally unrelated. It is often of interest, given a large collection of things for an efficient document comparison method to be available to determine quickly if some of the documents are plagiarized. Therefore, there is a need for document organization in IR systems to reduce information overload and to detect plagiarism since the increase of data sharply exacerbates the shortcomings of existing IR techniques to resolve information needs that are broad, vague, or even difficult to be expressed through a set of keywords.

For document comparison, inter-document similarity techniques are often used to define a distance score between words or a set of documents. Conventional IR approaches based on exact match search paradigm require all of the query terms and their specified textual relationships be satisfied precisely by the document representations. On the other hand, IR techniques based on inter-document similarity tend to offer a better solution in terms of accuracy, speed, consistency and ease of use because by having a measure of inter-document similarity, similar documents can be clustered together. Document comparison is useful to determine whether a document comprises a subset of plagiarized text from another document. These documents can be organized into meaningful clusters where inter-document similarities are captured in an edit-distance matrix which can then be represented using a suitable visualization technique, such as Kohonen's self organizing map. If a collection is well clustered, we can search only the cluster that will contain relevant documents. Thus, document clustering is another important document organization approach which would be useful because searching a smaller collection would improve IR effectiveness and efficiency.

Document organization will not be complete without a visualization system. Document visualization can be very helpful in data analysis for instance, for finding main topics that appear in larger sets of

[1] This work is supported in part by MOSTI grant 01-02-05-SF0003

978-1-4244-1692-9/08/$25.00 ©2008 IEEE

documents and provide a clearer picture on the inter-document similarities.

Extracting actionable insight from large high dimensional data sets, and its use for more effective decision-making, has become a pervasive problem across many fields in research and industry. When the amount of data increases, both in terms of size and dimensions, it is becoming harder to make accurate interpretations that still retain the main features of the data. An efficient visualization system for large document collections should be robust enough to provide a quick insight into the structure of the corpus and the textual relationships that exist between the documents. Moreover, it should focus on the intuitive presentation of and interactions with visual representations of the data so as to exploit the users own abilities in spotting patterns, outliers and trends [4].

II. System Architecture

Following the previous discussion, we would like to build a software tool to incorporate CSTW along with other useful features for document comparison and visualization purposes. Towards this end, CSTW and the document analyzer tool should be able to fulfill the following specifications:

1. Able to classify documents accurately based on the customized weights for different categories.
2. Visualize document similarity scores to analyze trends and behaviour of the document collection.
3. Fast document comparisons to detect for plagiarism.

The architecture of the software tool is further illustrated in Figure 1 for document classification, comparison, and visualization purposes. CSTW is being incorporated into the document analyzer to assign customized weights based on context awareness and category information. Besides that, there are several tools such as WordNet, POS tagger and an extended Kohonen Map presented by [5] will be included into the document analyzer tool. They will be useful to aid some of the implementations that will be explained later in Section III.

III. Document Analyzer: Design and Development

Within this research, it is proposed that a graphical user interface (GUI) software tool be developed. User interfaces occupy an important part of software development. In the first section, we draw the conclusions from the facts learned during the problem statement, and develop the necessary requirements to build our software tool. In the following subsections, we condensed several requirements into few goals we aim to achieve in creating the document analyzer tool. For the document analyzer tool, the GIMP Toolkit (GTK)[2] is used because it is able to build graphical user interfaces easily and effectively.

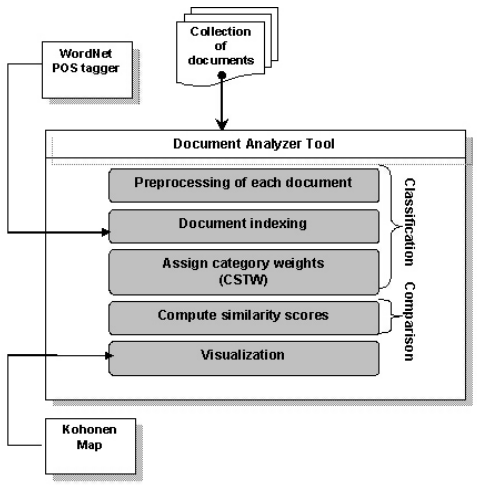

Figure 1. System Architecture

A. Requirements

1) Reliability: For fast and accurate document comparisons, we would like to develop a reliable document comparison tool that digs deep into a set of documents to find anything they have in common.

2) *Ease of Use:* The interface should be easy to navigate as possible. Design and development of a GUI is important to allow easier navigation and be able to control of multiple documents at one time.

3) Performance: The interface should instantly respond to any user request. The document comparison tool should have a fast processing time to compute weight scores and compare multiple documents simultaneously.

4) Flexibility: Features must be easily modifiable to accommodate different users preferences.

5) Scalability and Extensibility: As the document analyzer evolves, the interface must be able to scale accordingly to accommodate with this evolution. The code should be easily modifiable in order to accommodate additional requirements and features.

[2] GTK is a multi-platform toolkit for creating graphical user interfaces and it provides a complete set of widgets, which made it suitable for projects ranging from small one-off projects to complete application suites.

B. Initial Setup

1) Gimp Toolkit (GTK)

GTK+ is a C based toolkit for programming graphical applications and it is also function as a library providing a comprehensive collection of core widgets used to control the layout of a graphical user interface. At this time of writing, GTK+2.10.14 is used to build the GUI interfaces of the document analyzer tool. Although GTK uses C as its programming language, it is more like object oriented programming because it is implemented using the idea of classes and callback functions (pointers to functions). GTK+ has been designed from the ground up to support a range of languages, not only C/C++. Using GTK+ from languages such as Perl and Python (especially in combination with the Glade GUI builder) provides an effective method of rapid application development [6].

Before GTK widget toolkit can be compiled, a preliminary check of various other tools and libraries installed on the system is required. Another essential tool is needed during the build process is pkg-config.

Three of the main libraries needed before building and compiling GTK, which are GLib, Pango, and ATK. A description for each library is provided below to explain their functions with respect to GTK.

1. The GLib library is needed prior to the build process because it provides the core non-graphical functionality such as high level data types, Unicode manipulation, and an object and type system to C programs.
2. Pango is needed for internationalized text handling.
3. ATK stands for Accessibility Toolkit that provides a set of generic interfaces to allow accessibility technologies such as screen readers providing a GUI interaction.

2) WordNet Installation and Setup

Unlike the installation for GTK, installing WordNet is a more straightforward process with less libraries to consider. Although WordNet Version 3.0 is the current release from Princeton University's Cognitive Science Laboratory and is the latest version available for download, WordNet 2.1 is recommended as it has already sufficient features for word searching and sense relating. WordNet 2.1 unix package is required as the document analyzer tool is built on Linux platform. Since the WordNet browser makes use of the open source Tcl and Tk packages, both the Tcl and Tk packages are required prior WordNet. A C compiler is needed before Tcl/Tk or WordNet installation. GNU gcc compiler is recommended since WordNet is built and tested with the compiler.

3) QTAG Setup for POS Tagging

QTAG is an acronym for probabilistic parts-of-speech tagger developed by the University Birmingham. It functions by parsing passages of text and returning each word token with the designated part-of-speech. It is a fairly robust, easy to use and lightweight tool with minimal memory consumption that employ the statistical methods to tag texts with good accuracy. QTAG can be run either as a stand-alone program, in which the texts are prepared and loaded into QTAG or integrated as a module via API.

In order to run QTAG successfully, the Java run-time kit need to installed and two sets of resource files are required on the system platform. The resource files can be self-created for a different language to enhance the tagging procedure but a pre-tagged training corpus is required. The size of the training data is significantly important for the accuracy of the tagging procedure, and thus the more data available the better the POS tagging will be. The jar-file is directly executable and in the above example qtag-eng is the basename of the resource files for English and myfile is the textfile.

IV. Application Features

As an overview description of the utilities or the different functions that are available, the application features are divided into the different parts of document organization approach. Figure 2 shows the main interface of the document analyzer tool.

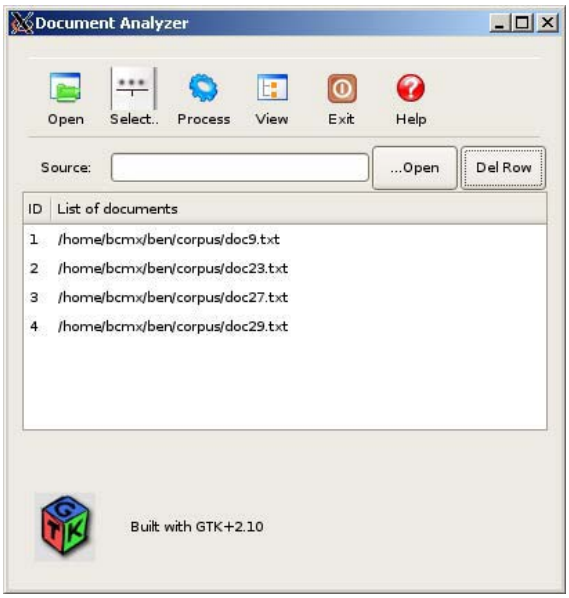

Figure 2. Document Analyzer Main Interface

A. Data Comparison

The *tf.idf* method produces an output based on the cosine similarity between a source document and a set of documents to be compared. The *tf.idf* weighting takes into account of the weight as a statistical measure used to evaluate the degree of importance a term is to a document in a set of documents. Similarly for the *tf.idf* module, it will help to gauge document similarities by assigning scores to each pair of documents, where a higher score indicates a greater level of similarity to the source of document. Therefore by applying this module, the user can easily identify the plagiarized document from the original text.

The conventional *tf.idf* method uses all type of words. However in this module, we only take into account the extracted noun words by applying the part-of-speech (POS) method, which is further mentioned in [7].

Results from our experiments in [7] have confirmed that by using noun words will produce better performance. From the several tests carried out, we can see that there is not much of a difference in terms of the similarity scores when using all POS types when compared to only noun words. At occasional times, using nouns would produce better score results. Moreover, using the rest of other POS words will produce a larger document size, which will cause slower processing time.

B. Data Visualization

One of the added-value features of the software tool highlights the visualization of similarity scores for a collection of documents. Extended Kohonen's self-organizing map (SOM) is used to order a set of high-dimensional vectors in clarifying relations in a complex set of data. It would also allow *tf.idf* similarity scores to be easily viewed in an intuitive way for the different kinds of textual relationship between documents as illustrated in Table \ref{fig:textrel}.

DOCUMENT TEXTUAL RELATIONSHIPS TEST CASE

Doc	Textual Relationship	Score
D1	Identical document	1.00
D2	Modified document with small edits	0.92
D3	Modified document with small edits	0.84
D4	Reorganized document	0.97
D5	Reorganized document	0.94
D6	Revised document	0.87
D7	Condensed document	0.41
D8	Expanded document	0.75
D9	Document that include portions of other docs	0.82
D10	Document that include portions of other docs	0.71

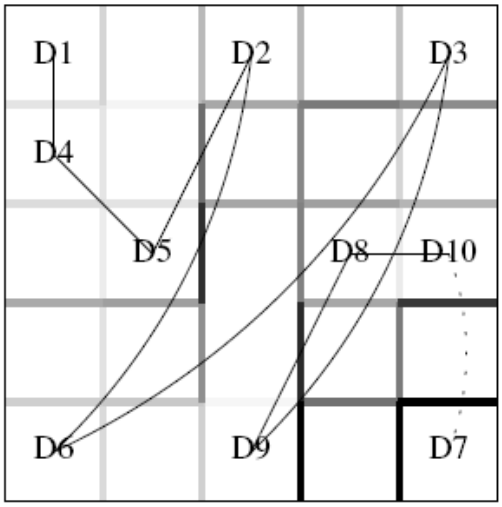

Figure 3. Extended Kohonen Map with Minimal Spanning Tree

In Figure 3, "D1" to "D10" refers to text documents over a range of sizes. As can be seen from the minimal spanning tree, it can be concluded that the source document is almost similarly related to documents that are reorganizations of other documents from the line linking from D1 to D5 and subsequently closely matched with documents that are the result of small edits or modifications to other documents like D2 and D3 in the map. From Figure 2, the source document is not similar to D7, which is a document that is a condensed version of other documents viewed by open lines indicate strong difference from D10 linking to D7.

Distance between points is proportional to the level of difference between the corresponding documents, but nearness does not necessarily imply resemblance. For instance, when making comparisons between D8 with D10 and D9 with D7, as shown in Figure 2; although the distance in between each pair shown is near, this does not necessarily imply resemblance. An additional type of information is made visible here. Solid connected lines indicate strong resemblance depicted by line spanning from D1 to D4, whereas open lines indicate strong difference depicted by line spanning from D10 to D7. This is why even if the distance between D9 and D7 is near these documents are not necessarily related.

To summarize, from this available utility the users are able to extract useful insight from the document similarity scores when for an example, students' assignments are submitted and analyzed with the document analyzer.

C. Data Classification

While the extended Kohonen Map is being incorporated, it only represents the preliminary stage of data comparison described in section IV-A. Referring to the following figures, they represent sample datasets (condensed for the sake of brevity) to test the efficacy of the document analyzer tool for document classification.

```
Biology also referred to as the biological
sciences, is the study of living organisms
utilizing the scientific method. Biology
examines the structure, function, growth,
origin, evolution, and distribution of
living things.
```

Figure 4. Biology-related document

```
The Wind biology began, and blew with
all his might and main a blast, cold
and fierce as a Thracian storm; but the
stronger he blew,the closer the traveler
wrapped his cloak around him, biology and
the tighter he biology grasped it with his
hands.
```

Figure 5. False biology document

Figure 3 illustrates a real case of a document which is related to the biology category. On the other hand, Figure 4 indicates where the ``biology'' term is randomly inserted to generate a false case scenario of a biology document. The proposed CSTW used in the software tool is able to distinguish category information from these documents and based on the computed weights, the documents can be differentiated whether they belong to a specific class.

To demonstrate the usefulness of the utility, the document analyzer tool can function in the following two perspectives:

1) *Weighting Score Analysis:* From Figure 6, there are total of 10 documents in the corpus. Each of the documents are weighted by CSTW accordingly by the category specified by the user. For an example, if the user selects the biology category, these documents are analyzed by the software tool and will be assigned a category weight score. From these scores as illustrated in Figure 6, the user is able to distinguish which documents are related to the selected category.

2) *Extended Kohonen Map Analysis:* Referring to Figure 7, the documents are separated into different regions of the map. Category weight scores computed with CSTW are projected onto the extended Kohonen Map to visualize any existing trends or ``patterns'' of emerging organization. Similar documents are clustered together within the same region where similarity is some function on a document. As can be seen, we observe that extended Kohonen Map incorporated into our software tool provides an added-value feature to provide a clearer representation view of a collection of documents for explorative analysis of large information spaces.

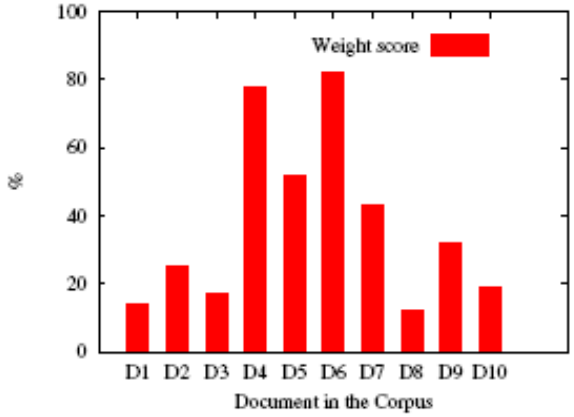

Figure 6. Category analysis by CSTW scores

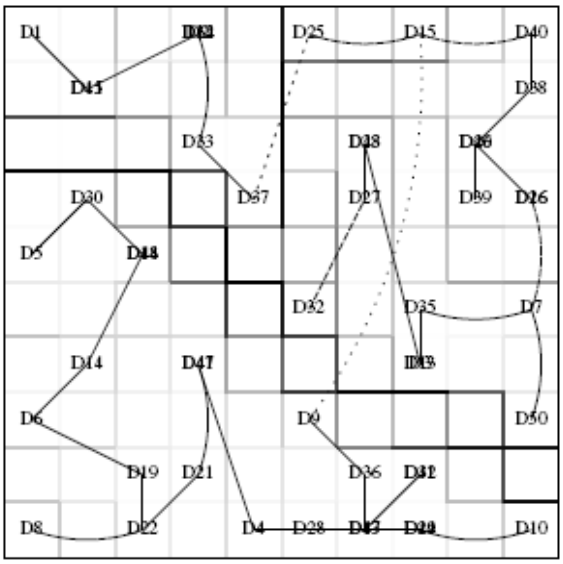

Figure 7. Category analysis by the extended Kohonen Map

D. Miscellaneous Functions: Auto-Generation Document Module

The aim of this function is to create a sample set of documents and from the many set of documents generated, the module can provide sufficient data for the benchmarking test used in an earlier work [7].

The auto-generation document function creates random documents by capturing randomly a spool of words from vocabulary files. This action will generate

randomized words, which are then inserted into a document without considering any grammar rules and sentence structure. The reason for this because we would want to analyze the existence of words, where we assign higher weights to words that are present in a document.

Options are available for the users to specify the number of documents to be generated and select different document sizes. This set of randomized documents would be useful to create false scenarios to test whether a weighting scheme is able to distinguish documents that are actually related to the different categories.

V. Analysis and Conclusions

Document structure analysis and information representation are innovative approaches to the exploration of document comparison and visualization in information retrieval. Document structures can be exploited to access relevant information to a given query to gauge similarities or detect plagiarism. For many users, document matching remains a problem because comparing documents manually is a laborious and error-prone task, particularly when dealing with a large document collection. While the current implementation of the document analyzer tool is at the preliminary stage, but thus far the developed tool is capable to perform the following tasks successfully:

- Extracting terms
- Identifying concept words
- Computing term weights
- Learning to weigh terms using category information
- Visualizing document similarities in high dimensional input spaces
- Generate overall document summaries and similarity scores
- Classify documents by category

Moreover, with the developed software several existing term weighting schemes are tested against our proposed method for document classification.

References

[1] J. Allan, "Building hypertext using information retrieval," in *Information Processing and Management*, vol. 33, 1997, p. 145159.

[2] D. Dubin, "Document analysis for visualization," in *In Proceedings of ACM SIGIR*, 1995, pp. 199–204.

[3] A. Leuski, "Evaluating document clustering for interactive information retrieval," in *CIKM*, 2001, pp. 33–40. [Online]. Available: citeseer.ist.psu.edu/leuski01evaluating.html

[4] J. Johansson, M. Jern, R. Treloar, and M. Jansson, "Visual analysis based on algorithmic classification," *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pp. 86–93, 2003.

[5] P. Kleiweg, "Neurale netwerken: Een inleidende cursus met practica voor de studie Alfa-Informatica," Master's thesis, Rijksuniversiteit Groningen, 1996, software available at http://www.let.rug.nl/ nerbonne/teach/neuro/kleiweg/nn.html.

[6] G. Rakic, *GTK+ Reference Manual*, 2007.

[7] C. M. X. Benjamin, W. L. Woon, and K. S. D. Wong, "A customized term weighting scheme for document classification