Using the Universal Similarity Metric to Map Correspondences between Witnesses

Holmes, Martin
mholmes@uvic.ca
University of Victoria

Thomas Sonnet de Courval's satirical work, the *Satyre Menippée du mariage*, was initially published in 1608. In 1609, an expanded version appeared, with the addition of a second satire, the *Timethélie, ou Censure des Femmes*. In 1621, the first satire appeared in a new edition titled *Satyre sur les Traverses du Mariage*, then in the following year, Sonnet de Courval published his *Œvres Satyriques*. The *Œvres* includes 12 satires, with the final six consisting of fragmented, re-organized and re-edited versions of the *Satyre Menippée* and the *Timethélie*. A new edition of the 1609 text was published in 1623, edited by the publisher and probably without Courval's consent. In 1627, two more editions of the Œvres *Satyriques* appeared. (Coste 340-341).

The *Mariage sous L'ancien Régime* project has already digitized the 1609 text and most of the 1621, and will be working on other editions in the future. Our objective is to produce a genetic edition of those parts of Courval's work which bear on marriage; in particular, we would like to map the process by which the original two satires (*Menippée* and *Timethélie*) were re-constituted as six satires in the *Œvres Satyriques*. Since the texts are lengthy (the 1609 text runs to nearly 3,000 lines), we have begun to investigate ways to automate this mapping to some degree, and in particular, methods of measuring similarity between two pieces of text. In particular, we needed to find a way to detect corresponding lines between two witnesses, even when those lines might have been both relocated and altered.

The Universal Similarity Metric is a method of measuring the similarity of two sets of data based on Kolmogorov complexity. It is described in Vitanyi (2005) as "so general that it works in every domain: music, text, literature, programs, genomes, executables, natural language determination, equally and simultaneously" (1). A practical implementation of this metric can be achieved using data compression, according to the following formula, where x and y are the two pieces of data being compared:

$$NCD(x,y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))}$$

(Vitanyi 2)

NCD is "Normalized Compression Distance", an expression of the similarity of x and y; C(x) and C(y) are the respective lengths of the two compressed inputs; and C(xy) is the length of the compressed concatenation of x and y. The resulting NCD is a value between 0 and 1, where proximity to zero indicates greater similarity. This metric has been widely used in the sciences; for instance, Krasnogor and Pelta (2004) describes its use to measure the similarity of protein structures, and Li et al. (2004) apply it to evolutionary trees and to building language trees based on text corpora. Its universality and simplicity suggested that it might be the ideal tool to discover correspondences between lines and line-groups at different points in two of Courval's texts. To test it, I have created a prototype application using Borland Delphi. These are some example values generated with the prototype (Figure 1):

Text 1	Text 2	NCD Score
These two lines are absolutely identical.	These two lines are absolutely identical.	0,0000000
The vndiscouered country, at whose sight	The vndiscouered Countrey, from whose Borne	0,3673469
To be, or not to be, I there's the point,	To be, or not to be, that is the Question:	0,4545455
To Die, to sleepe, is that all? I all:	Whether 'tis Nobler in the minde to suffer	0,7234043
これは日本語です。	To sleepe, perchance to Dreame; I, there's the rub,	0,8392857

Figure 1: Example comparisons showing NCD raw score using Borland's Pascal ZLib library (zlib 1.2.3). (Shakespearean lines taken from Quarto 1 and Folio 1, *Internet Shakespeare Editions* transcriptions.)

Scores below 0.5 appear to be strongly indicative of similarity, while those over 0.6 usually signify disparity; it is actually quite difficult to generate any score above 0.84, as shown by the final example, in which there are no points of similarity at all.

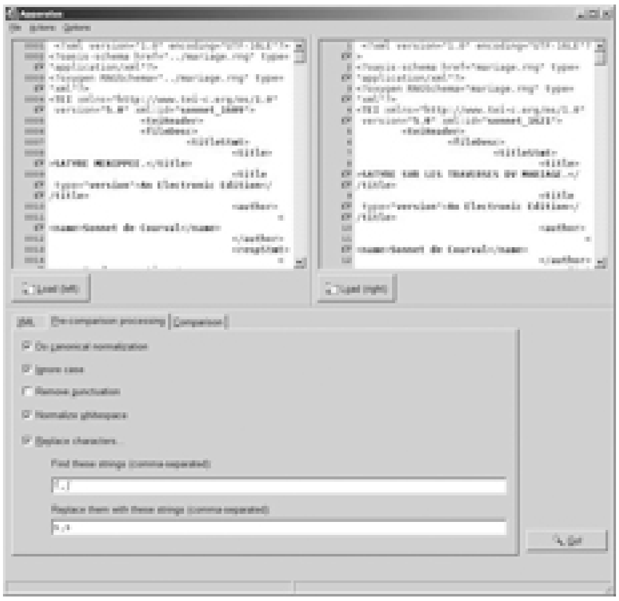
The prototype application takes two XML files as input, and performs the following steps:

1. Identifies the target elements to be compared (this is currently hard-coded, but would ideally be based on user-specified XPath).
2. Adds @id attributes to any of those elements which don't yet have them.
3. Extracts the text of all the target elements, and normalizes it in a variety of ways (specified by the user).
4. Compares each single text item with each other text item, and generates a comparison score for it.
5. (Optional) Runs an additional contextualizing algorithm which modifies the original scores based on the scores of surrounding elements in the document (see below).
6. Sorts the matches in ascending order of similarity score (best matches first).
7. Presents each of these matches to the user for categorization as one of:
   - Corresponding (equivalent) items
   - Not corresponding, but an interesting relationship
   - No relationship at all
8. Saves an XML file containing the scores for all matches, along with any categorization values chosen by the user.
9. Saves copies of the input files with the added @id attributes, and also a CSV version of the score data for use in a spreadsheet program.

This correspondence data can be used to provide a component of a critical apparatus attached to a witness, linking it to corresponding lines in other witnesses.

Step 5 is an attempt to detect correspondences between lines in situations where a line has changed significantly, or perhaps even been replaced completely, but the lines around it still match closely. Normally, where lines differ, a high score will be generated; if the user chooses only to examine the lower scores in the search for correspondences, the link between these two lines may not be detected. The contextualizing algorithm massages the score of each match such that it is affected by the score of the lines around it; if the preceding and following lines have low scores, the score of the line is lowered so that it too may be detected more easily as "corresponding", even though it has been substantially changed. The contextualizing algorithm can be run many times if required, massaging the scores each time. We are still investigating the outcomes and value of this process.

These screenshots (Figures 2 and 3) show the prototype in use.



☑ Do canonical normalization

☑ Ignore case

☐ Remove punctuation

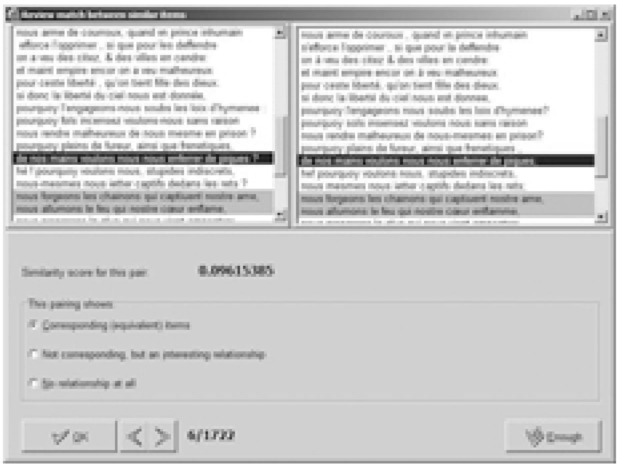
☑ Normalize whitespace

☑ Replace characters...

Find these strings (comma-separated)

Replace them with these strings (comma-separated)

Figure 2: *The main window of the prototype, showing the two input texts, and the Pre-comparison processing settings.*



Similarity score for this pair: 0.09615385

This pairing shows:

● Corresponding (equivalent) items

○ Not corresponding, but an interesting relationship

○ No relationship at all

Figure 3: *Reviewing matches between lines based on score.*

This application is in many ways similar to the TEI-Comparator project created as part of the Holinshed Project. This is described in some detail in Cummings (2009). When writing our *Mariage* prototype at the beginning of 2009, I was unaware of the TEI-Comparator project; in addressing similar problems, we have arrived at remarkably similar solutions, especially in terms of process. The comparison algorithm used by TEI-Comparator, which is called Shingle Cloud, was developed by Arno Mittelbach, and uses a completely different process of comparison based on n-grams. Documentation for TEI-Comparator will be available soon; when it appears, I am looking forward to running tests to compare results between Shingle Cloud and the Universal Similarity Metric, and I will report the results in this paper.

The prototype application will probably now be ported to C++, to create a cross-platform application, and I also intend to create a standalone Java library that can be called from the command line or from another Java application; perhaps it might be integrated into TEI-Comparator as an alternative comparison metric.

References

Coste, Joël (2008). 'Un regard médical sur la société française à l'époque d'Henri IV et de Marie de Médicis'. *XVIIe siècle.* **239**: 339-61.

Cummings, James (4 September 2009). *"TEI-Comparator." Blog posting on In my <element/>.* http://blogs.oucs.ox.ac.uk/jamesc/2009/09/04/tei-comparator/.

Krasnogor, N. and D. A. Pelta (2004). 'Measuring the similarity of protein structures by means of the universal similarity metric'. *Bioinformatics.* **20(7)**: 1015-1021. http://bioinformatics.oxfordjournals.org/cgi/reprint/20/7/1015.pdf.

Li, Ming, Xin Chen, Xin Li, Bin Ma, and Paul Vitanyi (2004). 'The Similarity Metric'. *IEEE Transactions on Information Theory.* **50(12)**: 3250-3264.

Vitanyi, Paul (2005). 'Universal Similarity'. *Proc. ITW2005 - IEEE ITSOC Information Theory Workshop 2005 on Coding and Complexity.* Rotorua, New Zealand, 29th Aug. - 1st Sept., 2005. http://www.cwi.nl/~paulv/papers/itw05.pdf.