Measuring Text Similarity With Dynamic Time Warping

Michael Matuschek
michael.matuschek@
uni-duesseldorf.de

Tim Schlüter
schlueter@
cs.uni-duesseldorf.de

Stefan Conrad
conrad@
cs.uni-duesseldorf.de

Heinrich-Heine-
Universität
Düsseldorf
Universitätsstraße 1
40225 Düsseldorf, Germany

ABSTRACT

In this work, we describe an approach which aims to make typed texts comparable with temporal data mining methods. This proposal was made in earlier work [11], but to our knowledge no significant research on this subject has been done yet. The basic idea is to derive artificial time series from texts by counting the occurrences of relevant keywords in a sliding window applied to them, and these time series can be compared with techniques of time series analysis. In this particular case the Dynamic Time Warping distance [3] was used. By extensive testing adequate parameters for time series calculation were derived, and we show that this approach might aid in the recognition of similar texts since the observed distances between similar documents are significantly lower than those between unrelated texts. Our idea might also be especially suitable for comparison in different languages since only the keyword translations must be known.

Categories and Subject Descriptors: H.2.8 [Database Applications]: Data mining; I.2.7 [Natural Language Processing]: Text analysis; G.3 [PROBABILITY AND STATISTICS]: Time series analysis

Additional Key Words and Phrases: Data Mining, Pseudo Time Series, Dynamic Time Warping, Text Comparison, Text Databases and Digital Libraries, Databases and Information Retrieval

1. INTRODUCTION

The recent development of the internet allows to easily find and copy documents covering almost every topic. Thus, the detection of plagiarism is an eminent problem, e.g. in academics and journalism. It is vital to decide whether a given text is original work or was published before with high confidence as the theft of creative work might be bad style in the best case but can also imply serious consequences such as the revocation of an academic degree. The identification of documents with similar content is also important in information retrieval where "query by example" is a common approach to find texts covering a certain topic. Moreover, in the course of linguistic analysis it might be interesting to identify documents with similar characteristics, e.g. regarding the occurences of terms within them.

In this paper, we discuss an approach which allows to compare texts by considering the frequency of terms as well as their position, but with a certain tolerance regarding syntactic variation or translation of sentences [10]. This is achieved by representing texts as time series, in conjunction with an appropriate distance measure. These artificial time series were first proposed in [11], and (to the best of our knowledge) this approach has not yet been further explored. The basic idea for obtaining the time series is to define one or more keywords and count how often these terms occur within a predefined sliding window which is applied to the document. We present an implementation of this approach which calculates time series from texts and the distances between them. The purpose of this application is to aid the user in the identification of texts which are similar according to their term distribution and therefore might be interesting for closer examination. The similarity measure in our case is the Dynamic Time Warping distance [3], which we deem most appropriate for this task.

The rest of this paper is structured as follows: In section 2 we review some related work, in section 3 a short overview of the DTW distance is given, in section 4 the calculation of time series from texts in our implementation is described, in section 5 we present a short overview of our implementation, in section 6 some results from our tests are given, and finally, in section 7, some concluding remarks and directions for future work are given.

2. RELATED WORK

A common approach from the area of information retrieval to identify texts with similar content is the *vector space model* [14] which aims to make texts comparable by representing them as a vector of terms. In the simplest case these terms are the distinct words contained in a document and the vector describes how often each word occurs. The document vectors can then be compared, e.g. using the cosine distance [2]. A more complex approach is to calculate a graph representing a text where nodes and edges represent the terms and and the associations between them [16].

IDEAS08 2008, September 10-12, Coimbra [Portugal]
Editor: Bipin C. DESAI

However, although these techniques might reliably identify texts with the same topic, they might not be sufficient for the special case of plagiarism search since other document properties have to be considered as well.

Ideas in this field include measuring the edit distance between text segments [19], calculating the occurrences of letter and word combinations [8] and the syntactic analysis of individual sentences [9], [17]. These "low-level" approaches aim at identifying sentences or passages which have been copied verbatim or with only slight variations, and it was also proposed to combine them with content-based comparison to improve the retrieval performance [5]. Nevertheless, these methods might fail if enough variation is introduced to cover plagiarism, and they most certainly fail if translated texts are considered since syntax and other "technical" properties of documents may vary greatly in different languages.

As we will see later on, these issues may to some extent be resolved by representing texts as artifical time series. Similar proposals to represent non-temporal data as time series to gain additional insight have been made in fields such a shape recognition in images [18] and the analysis of handwritings [12].

3. DYNAMIC TIME WARPING

In the context of data mining, it is vital to meaningfully define similarity between data objects. A well-discussed approach to do so is the Dynamic Time Warping distance which has been used in many contexts ([12], [13], [6]). The basic idea is to find a non-linear matching of the points of two time series which is intuitively equivalent to stretching or compressing the time series in the x-axis. The result is a distance measure which captures the human perception of similarity better than the classic Euclidean distance. An example of such a warping can be found in figure 1.

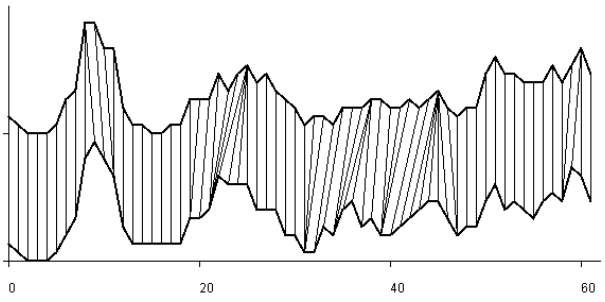

Figure 1: Example of an optimal warping between two time series.

The formal definition of this technique is as follows [7]: Suppose we investigate two time series, $A = a_1, a_2, ..., a_n$ and $B = b_1, b_2, ..., b_m$, with lenghts n and m, respectively. To align these sequences we construct a $n \times m$-matrix where element (i,j) of the matrix contains the distance $d(a_i, b_j)$ between the points a_i and b_j. In this context the Euclidean distance is typically used. Each matrix element (i, j) now corresponds to an alignment between the points a_i and b_j, and a *warping path* W is a contiguous set of matrix elements that defines a mapping between the whole sequences A and B. Formally, $W = w_1, w_2, ..., w_K$ with $w_k = (i, j)_k$ and $max(m, n) \leq K < m + n - 1$.

In order to avoid "degenerated" warping paths some constraints are usually applied:

- $w_1 = (1, 1)$ and $w_K = (m, n)$. This expresses that the warping path should start in the lower left corner of the matrix and end in the upper right corner.
- Given $w_k = (i, j)$, then $w_{k-1} = (i', j')$ with $0 \leq i - i' \leq 1$ and $0 \leq j - j' \leq 1$. Informally, this restricts the warping path in a way that it only moves towards the upper right corner.

Of course, there are many possible paths which fulfill these simple restrictions, but the goal is to find the optimal warping path i.e. the one which minimizes the warping cost $DTW(A, B)$ between A and B:

$$DTW(A, B) = min\left(\frac{1}{K}\sqrt{\sum_{k=1}^{K} w_k}\right) \quad (1)$$

A straightforward method for finding this path is dynamic programming, but a major disadvantage of the naive approach is the runtime of $O(mn)$. Extensive research has led to some interesting proposals for optimization ([13], [15], [4]), including the restriction of the possible warping path and the dimensionality reduction of the underlying time series using some kind of approximation. However, these improvements will not be discussed here in detail since they are beyond the scope of this paper.

4. CALCULATION OF ARTIFICIAL TIME SERIES

Besides the Dymanic Time Warping distance as similarity measure, the second cornerstone of our approach are the time series derived from texts. In this section we will explain how these are defined and calculated.

First of all, one or more keywords need to be selected, i.e. the words that will be counted during time series computation. Semantically important terms with high frequency are a good choice here. The second parameter is the width of the window which will be moved along the text, i.e. the amount of words considered at each step. The natural maximum for this value is the word count of the whole text. The third (and last) parameter determines how far the window is moved forward in each computation step. The value for this step size is limited by the window size since for a larger value not every word in the text would be considered. An illustration of all parameters is given in figure 2 where the whole time series creation process is explained with a small example. As figure 2 also shows, an intermediate step in calculating the time series in our implementation is creating a binary representation of the text, where the keywords selected by the user are shown as 1 and the other words as 0. This simplifies the calculation of the individual points constituting a time series as this can be achieved with basic arithmetics.

The choice of parameters allows to directly adjust the length of the time series and, as a consequence, the precision and computation time. The relationship between the length of a time series l, the window size w and the step size s for a text of length N is expressed by the following equation:

$$l = \left\lceil \frac{N-w}{s} \right\rceil + 1 \quad (2)$$

The intuition behind this formula is that $N - w$ is the amount of words that remains after the first window (i.e. the first point) has been calculated. Since the window is moved forward s words in each step the correct time series length is derived by dividing this remainder by s; the "+ 1" makes sure the first point is also included. Note that the last window might be smaller than w if s is not a divisor of $N - w$.

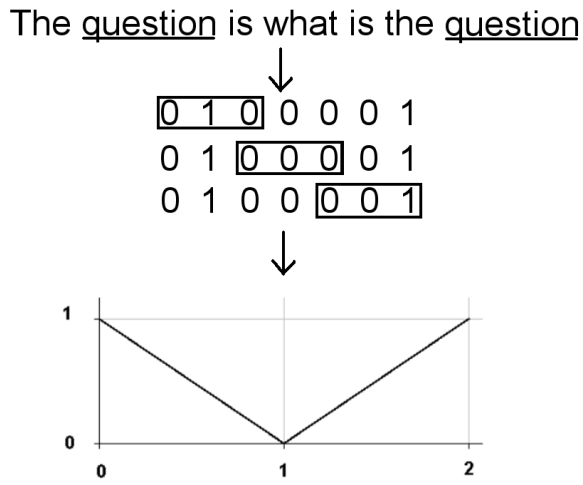

Figure 2: Illustration of the time series creation process. The keyword of interest is "question", hence occurrences of this term are shown as 1 in the binary representation and all other terms as 0. The window size is 3, the step size is 2, and these settings yield a time series of length 3 with the values (1, 0, 1) derived by counting the set bits in each window.

5. IMPLEMENTATION

In this section, we briefly describe our implemenation which allows to compare typed texts using their time series representation. We will start off by explaining how texts are imported and preprocessed, then we explain how the time series are calculated and compared.

5.1 Text Import

First of all, the user can choose whether he wishes to import textual documents in German or English. As an option, the user can choose to strip the texts of stopwords (i.e. semantically unimportant words), and it is also possible to apply stemming at this point. This means that grammatically different uses of words are resolved to allow the identification of terms in different contexts. These two techniques are established in the field of text comparison and often improve the quality of the results, so we decided to incorporate them into our program since this was possible without too much effort. See [2] for a detailed description of various methods for text preprocessing.

After a text is imported, the unique terms are identified and it is measured how relevant each of them is, i.e. we count how often each term occurs. This relevance value will be helpful to select the appropriate keywords for time series calculation later on. The whole import process is also repeated for every paragraph of the original document to allow for the comparison of individual text sections.

5.2 Time Series Calculation

Once the documents of interest are imported the next step towards comparison is the calculation of the artificial (or "pseudo") time series which are the foundation of our work. This basically happens as described in section 4, but some features have been added for the sake of user convenience. First of all, in the list of selectable keywords, the relevance is displayed next to each word to aid the user in his decision, and it also possible to hide the stopwords so that only semantically important words are shown. Regarding window and step size, the values can easily be adjusted to certain ratios (e.g. "the window size should be $\frac{1}{4}$ of the text length"), and the user may also adopt parameter values which have been selected for previous time series calculation.

5.3 Time Series Comparison

After the time series have been calculated the final (and crucial) step is to compare them, i.e. calculate the distances between them. In our implementation, the user has several options considering which time series should be compared, ranging from the comparison of two single time series to the comparison of all time series of all texts in the database. However, we assume that the typical scenario is that a query text (or paragraph) is compared to a set of documents to check whether a similar document can be obtained.

In the actual comparison process, the Dynamic Time Warping distance is used and it is calculated exactly as described in section 3. The reason why the DTW distance (instead of the Euclidean distance) was chosen as the main distance measure in this implementation is that the Euclidean distance is not suitable if minor variations and shifts in the curves are observed; however, this is exactly the kind of behavior expected when a text is translated into another language or slightly altered to disguise plagiarism: The overall word count and the exact counts and positions of the keywords may change, but the general distribution of the keywords remains similar. In this scenario the DTW distance is appropriate since it allows comparison with a certain degree of tolerance regarding these alterations.

Nevertheless, the different extensions of the DTW algorithm were not implemented. The aim of these is (for most parts) speeding up the DTW calculation, but as the possibility of dimensionality reduction is intrinsically included in our time series calculation approach implementing modifications of the DTW algorithm was not a priority, although this could easily be done in the future.

6. EVALUATION

After describing the basic concepts of our system, we present some empirical results from tests with a variety of documents and parameter settings which hint towards the usefulness of the approach in practical applications.

In short, our results show that for appropriate parameters (semantically important words as keywords, window size not too large/small) it is possible to identify similar documents with high confidence as characteristic curves for texts can be obtained which remain quite stable even if a text is altered.

This is especially true for translated texts, where only the proper translations of the keywords are needed for meaningful cross-linguistic comparisons. Moreover, it turned out that the approach is sensitive enough for document retrieval from a considerably large database ("query by example"). In the following subsections a detailed breakdown of our results will be provided.

6.1 Parameter Influence

As mentioned earlier, the first (and probably most important) parameter for time series calculation is the choice of keywords. A natural assumption, which is also supported by the tests, is that stopwords, i.e. semantically unimportant words, are not well suited for this purpose. The curves yielded by terms like "the" or "and" were comparable to random walk data, showing almost no remarkable peaks due to their even distribution in the text. Another apparent problem in this context is the comparison of texts from different languages since the position and amount of certain stopwords may vary wildly because of different grammatical constructs and ambiguous translations. Therefore, the focus of the tests was on non-stopwords with high relevance as keywords, assuming that their distribution best captures the profile of a given text.

When considering the value for the window size w it can intuitively be claimed that for a bigger w the count observed for each individual window also becomes larger and the influence of a single keyword occurrence is small, resulting in a rather imprecise representation of a text. In the extreme case (window size equals text length) all occurrences of a keyword are concentrated in a single point and distance calculation between two texts degenerates to a comparison of the keyword occurrence count. On the other hand, for a window size of 1 we obtain a single peak for each keyword occurence. In this case it is plausible to assume that even rather dissimilar time series have small distances due to the nature of the DTW algorithm; the small peaks are simply "warped together" then, ignoring their exact positions on the x-axis (i.e. ignoring the positions of the keywords in the text). These examinations suggest that for meaningful results a balance between too small and too large window sizes has to be found.

When adjusting the value for the step size s while sticking to a constant window size w we observe that this parameter (unlike w) does hardly influence the overall shape of the curve. Instead, it determines the curve smoothness. For the extreme case of the step size being 1 each keyword occurrence shows on the curve, resulting in a "nervous" curve which displays even small variations. On the other hand, when the step size is great, minor variations are more or less "swallowed" in the calculation process which implies less precise comparison results. However, since the length l of the resulting time series grows linearily with a larger s (compare equation 2) and the run time of the DTW algorithm is $O(l^2)$, a trade-off between precision and computational effort has to be made at this point.

6.2 Retrieval Testing

As a basis to test the validity of our approach and to determine useful parameters, time series were calculated for a variety of English texts where the non-stopwords occurring in each text with highest relevance were selected as keywords. Every text was compared to a slightly altered version of itself (using the same keyword), a translation into German (using the translated keyword) and several dissimilar texts of varying lengths. In the latter case, keywords with the same relevance as the original keyword were chosen.

The central observation we made is that it is indeed possible to identify similar texts from the observed distances. For appropriate window sizes (we recommend a minimum of 500 words for meaningful results) the measured DTW distances between the unrelated texts are in most cases significantly (two to more than ten times) higher than those between the similar texts. Thus, the time series approach to text comparison apparently has the potential to discriminate similar documents from dissimilar ones. Moreover, the distances between the texts and their altered versions are very close to the distances between the translated versions. This implies that the approach is language-independent in the sense that it allows to meaningfully compare texts in different languages with almost no additional effort; the only information needed in this case is the translation of the keyword. This is quite different for techniques like the vector space model where basically all terms have to be translated for useful results.

Apart from the comparison of manually selected documents to determine useful parameters, our approach was also tested in a "query by example" setting where texts were matched against a whole collection of documents to see if the correct answer could be retrieved using the time series representation. The results mirrored the first observation that the implemented idea might be well suited for the recognition of similar texts. Although over 1000 comparisons were made for each single document, the correct answer was always in the top ten of the most similar documents returned. This implies that the technique might be sensitive enough for retrieval from even larger document databases since it only produces few false hits (less than 1% in this case).

7. CONCLUSIONS AND FUTURE WORK

In this work we presented an implementation which allows to compare texts using temporal data mining techniques. For that purpose the problem of text similarity was reformulated as a problem of time series similarity (as originally proposed in [11]), and the test results show that this approach might be useful for the desired task if suitable parameters are selected for calculating artifical time series from the input documents. Nevertheless, there are several enhancements and extensions which might be considered in future work.

First of all, an obvious extension is text comparison for other languages than English and German, especially since the technique proved to be a potential aid for language-independent comparison. If the user does not wish to apply any preprocessing steps the current implementation is already sufficient, otherwise stemmers and stopword lists for the languages involved would need to be provided. These are for most parts freely available and could be added quite easily.

Moreover, the integration of other established solutions from the field of information retrieval should also be possible. So far, only stopword reduction and stemming have been considered, but further ideas include the resolution of homonyms and synonyms and the detection of composite terms, i.e. terms which consist of more than one word. With these additional preprocessing steps a more accurate repre-

sentation of a document could be obtained, especially since the test documents imply that composite terms are not uncommon. However, the use of sufficiently large dictionaries might be necessary to achieve good results.

Of course, dictionaries could also be useful to support the user by automatically suggesting appropriate keywords for time series calculation. This might be a simple task for texts in the same language (as the same keyword should most likely be used for all texts involved), but it might be less obvious when translations are considered. A more elaborate implementation could also make recommendations for the other parameters (window and step size) based on the text length and the keyword relevance. In an additional step, the automatic labeling of documents as "similar" should also be possible, but more research on the relationship between the parameters and the observed distances would be needed for reliable results.

The use of multivariate time series (i.e. the composition of time series from multiple keywords) is already possible in our implementation, but in our testing scenario we could not achieve better results than with univariate time series. However, it might be possible to gain additional insight from using more than one time series per text by modifying the input parameters. Ideas include the attachment of weights to the dimensions according to the keyword relevances [2] and the adaptation of the window and step sizes for less relevant keywords. Nevertheless, further testing would be necessary to determine useful settings.

A profound addition to the implementation would be the realization of partial matching. By now it is only possible to compare documents or paragraphs as a whole, but of course it is also possible that, for example, a copied passage is seamlessly embedded into another document. This would probably go undetected when using only the DTW distance, thus the notion of similarity would need to be adapted. One idea is to partition the time series into windows of variable lengths and then identify matching windows instead of considering the whole text [1]. This would correspond to slicing a document into "artificial" paragraphs and then comparing these. An idea which relies on the characteristics of DTW is to calculate the amount of warping observed between different regions of the texts, i.e. the discrepancy between a linear assignment and the actual warping. A proposal on how to define this was made in [7], and similar passages could then be identified since the warping between them should be small.

To conclude this discussion of future work, a final question to be raised is if and how the presented approach can be combined with established techniques such as the vector space model to improve the overall performance in practical applications.

8. REFERENCES

[1] R. Agrawal, K. Lin, H. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Twenty-First International Conference on Very Large Data Bases*, 1995.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Pearson / Addison Wesley, 1999.

[3] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, pages 359–370, 1994.

[4] S. Chu, E. Keogh, D. Hart, and M. Pazzani. Iterative deepening dynamic time warping for time series. In *In Proc 2 na SIAM International Conference on Data Mining*, 2002.

[5] P. Iyer and A. Singh. Document similarity analysis for a plagiarism detection system. In *IICAI*, pages 2534–2544, 2005.

[6] E. Keogh and M. Pazzani. Scaling up dynamic time warping for datamining applications. In *Knowledge Discovery and Data Mining*, 2000.

[7] E. Keogh and M. Pazzani. Derivative dynamic time warping. In *First SIAM International Conference on Data Mining, (Chicago, IL)*., 2001.

[8] T. Lancaster and F. Culwin. Towards an error free plagarism detection process. *SIGCSE Bull.*, 33(3):57–60, 2001.

[9] C. Leung and Y. Chan. A natural language processing approach to automatic plagiarism detection. In *SIGITE '07: Proceedings of the 8th ACM SIGITE conference on Information technology education*, pages 213–218, New York, NY, USA, 2007. ACM.

[10] M. Matuschek. *Temporal Aspects in Data Mining (Master Thesis)*. Heinrich-Heine-Universität, Düsseldorf, 2008.

[11] C. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, 2004.

[12] T. Rath and R. Manmatha. Word image matching using dynamic time warping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, 2003.

[13] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. In *Readings in speech recognition*, 1990.

[14] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[15] S. Salvador and P. Chan. FastDTW: toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5), 2007.

[16] J. Tomita, H. Nakawatase, and M. Ishii. Calculating similarity between texts using graph-based text representation model. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 248–249, New York, NY, USA, 2004. ACM.

[17] D. White and M. Joy. Sentence-based natural language plagiarism detection. *J. Educ. Resour. Comput.*, 4(4):2, 2004.

[18] X. Xi, E. Keogh, L. Wei, and A. Mafra-Neto. Finding motifs in a database of shapes. In *SDM*, 2007.

[19] M. Zini, M. Fabbri, M. Moneglia, and A. Panunzi. Plagiarism detection through multilevel text comparison. In *AXMEDIS*, pages 181–185. IEEE Computer Society, 2006.