which means "surround with a rampart or other fortification" **circumvallate**

Hello, my name is Clemens Binz and my Bachelor thesis topic is “Exploiting Hierarchical Structure in Time Series Data”

Since 2009 Google has continuously scanned Books and published this data on the Google Books Ngram Corpus. The Dataset is based on Books from around the 1500s up to 2019 and consists of Ngrams that appear at least 40 times across the corpus. This Data is stored as plain text in the following format. The ngram Word combination followed by the Year, in this case 1978, the 335 occurrences of this ngram in that year overall and lastly the 91 distinct books in which the ngram was found.

The resulting time series based of this data can be of use for historians or for linguistic purposes. This Plot of the Relative frequency of the 1-gram Revolution in relation to famous historic events may be a concrete use case for the Google Books Ngram Corpus Data.

If we take a closer look into the Dataset as a whole, then it becomes clear that larger n-grams dominate the space requirements for every language that is a part of the Corpus provided by Google. Englisch works were focused on and are clearly overrepresented but German and French 5-grams have also reached Terabytes of storage needs.

The Corpus has unique properties such as space requirements growing in relation to n and data from natural language. Therefore, with existing dependencies between different ngram time series based on grammatical rules which can be exploited by our Approach for time series Compresssion.

Time Series compression is a well-researched topic and the most popular approaches can be categorized into 4 Sections as well as Lossy or Lossless compression. Piecewise or Dictionary-based approximation is based on the principle that time series share some common segments. These segments can be extracted into atoms, such that a time series segment can be represented with a sequence of these atoms. Some state-of-the-art techniques in this Section are CORAD or any SAX variation. The main idea behind function approximation is that a time series can be represented by a simpler function similar to interpolation. Here DFT, DWT and Piecewise Polynomial Approximation are prime examples. An autoencoder like RNNA or LFZip are particular neural networks that are trained to give as output the same values passed as input but compressed and optimized. And lastly Sequential Algorithms combine different compression techniques after each other to achieve higher compression. Popular choices include Huffman, Delta or Run-length encoding after each other.

In contrast to these techniques, we intend to use our existing domain knowledge as leverage for a new compression Approach.

we established that our data is based on natural Language and has inherent dependencies between similar nGram time series. If we take a look at the two functions for the 3gram "United States of” and the 4 Gram “United States of America”, then these dependencies become clearly visible to the naked eye. There probably exists some kind of scaling to only store one of these functions for this and other similar examples in the Google Books Ngram Corpus.

The first Observation we made is that the first (or last) n − 1 tokens of a ngram severely restrict the set of possible last (or first) tokens. This information gain increases with higher ngrams because correct and sensible sentence’s structure restricts possible new tokens.

This is where our Hierarchical Compression Algorithm was formed. And it goes as follows: for a given set of n-gram frequency series T = {(g0, τ0), . . . ,(gn, τn)} and a given n-gram gi , find gl (gr ) by removing the last (first) token from gi , and minimize ε = λ(τi , αl · τl + αr · τr) for a given loss function λ. If ε < δ for a given error bound δ, store only αl and αr instead of τi. Therefore, ngram times series of higher order can be stored as linear combinations of n-1 gram time series.

Here u can see our previous 4gram time series “United states of America” next to an example scaling of the two 3grams “United States of” with scaling factor 0.25 and “Sates of America” with a factor of 0.75. Which merge into nearly identical functions after 1975.

Our evaluation will be structured as followed. Experiments will be based on the two metrics; Compression rate and Root mean square Error as distance function between our approximated function and the original. These were chosen as the preferred metrics in many lossy states of the art time series compression techniques and therefore also fit for our purpose. Our plan is to compare our Compression Rate to different allowed error rates previously set as hyper parameter similar to this diagram. The experiments will be concluded on the Google Books Ngram Corpus.

We are going to compare our technique to multiple previously shown state of the art lossy compression techniques like LFzip, CORAD, SZ3 and DWT.

Together with the DFT algorithm, the DWT is commonly used for signal compression. DWT uses wavelet functions to approximate time series data and was chosen as a less complicated textbook baseline comparison.

LFZip is an Autoencoder based on the prediction-quantization-entropy coder framework.

The SZ compressors are sequential algorithms mixing linear regression, linear scale quantization, Huffman encoding and lossless compression. SZ3 improves previous SZ versions by replacing linear regression with cubic spline interpolation.

And finally, Corad is a Dictionary based technique. It learns the correlation between times series measured with the Pearson coefficient and stores only uncorrelated data in a dictionary and correlated data in a Correlation Matrix.

We are going to find out how these techniques compare to our approach.

These are my sources and thank you for your time.