

# Compression of Multilingual Aligned Texts

*Ehud S. Conley* and *Shmuel T. Klein*  
Department of Computer Science  
Bar-Ilan University, Ramat-Gan 52900, Israel  
{konli,tomi}@cs.biu.ac.il

In countries like Canada, Belgium and Switzerland, where speakers of two or more languages live side-by-side, all official texts have to be published in multilingual form. Similarly, all official texts of the European Union are translated into the languages of all member states. As a result, there is a growing corpus of important texts, large parts of which are highly redundant, since they do not have any information content of their own. Rather, they are just transformed copies of some other parts of the text collection.

We wish to exploit this redundancy to improve compression efficiency in such situations, and introduce the notion of *Multilingual-Text Compression*: one is given two or more texts, which are supposed to be translations of each other and are referred to as *parallel* texts. One of the texts will be stored on its own (possibly using a standard compression scheme), whereas the other texts can be compressed by referring to the first text, using appropriate dictionaries.

The basis for enabling multilingual-text compression is first the ability to match the corresponding parts of related texts by identifying semantic correspondences across the various sub-texts, a task generally referred to as *text alignment*. Some methods for detailed alignment use an existing multilingual glossary, but all of them generate their own probabilistic glossary, which corresponds to the processed text.

The idea of the current work is to save storage space by replacing words and phrases with pointers to their translations, determined by any alignment algorithm. Unaligned words are compressed on their own using HuffWord encoding. The offset field in each pointer indicates the distance of the referred translation from a rough *linear alignment*, simply computed using the ratio between the lengths of the given parallel sections. As opposed to other pointer-based methods, these values are always very small (up to a few dozens, but usually less than 10).

Another pointer field is the index of the translation of the source sequence into the target language within the bilingual glossary. The pointers also store the number of words to be read from the source text as well as some lemmatization/inflection information, enabling the exact restoration of the original text through lemmata/variant monolingual dictionaries. As all pointer field values are expected to be very small, they can be encoded by a space-efficient variable-length code (Gamma, Huffman or the like).

The suggested method was tested on an English-French corpus of the European Union. The French part (ca. 7.5MB) was compressed using pointers towards the English part. The obtained compression rate (22.0%) is similar to the performances of Bzip and HuffWord and better than that of Gzip. However, Bzip and Gzip's performances degrade when small sub-sections are processed separately, which makes them inappropriate for systems which often decode only small pieces.