Boosting Bitext Compression

Joaquín Adiego, Miguel A. Martínez-Prieto, Javier E. Hoyos-Torío, and Felipe Sánchez-Martínez

Abstract Bilingual parallel corpora, also know as bitexts, convey the same information in two different languages. This implies that when modelling bitexts one can take advantage of the fact that there exists a relation between both texts; the text alignment task allow to establish such relationship. In this paper we propose different approaches that use words and *biwords* (pairs made of two words, each one from a different text) as representation symbolic units. The properties of these approaches are analysed from a statistical point of view and tested as a preprocessing step to general purpose compressors. The results obtained suggest interesting conclusions concerning the use of both words and biwords. When encoded models are used as compression boosters we achieve compression ratios improving state-of-the-art compressors up to 6.5 percentage points, being up to 40% faster.

Joaquín Adiego
Dep. de Informática, Universidad de Valladolid, Spain. e-mail: jadiego@infor.uva.es

Miguel A. Martínez-Prieto
Dep. de Informática, Universidad de Valladolid, Spain; Dep. de Ciencias de la Computación, Universidad de Chile, Chile. e-mail: migumar2@infor.uva.es

Javier E. Hoyos-Torío
Dep. de Informática, Universidad de Valladolid, Spain. e-mail: javierht@gmail.com

Felipe Sánchez-Martínez
Dep. de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, Spain. e-mail: fsanchez@dlsi.ua.es

Work supported by the Spanish Government through projects TIN2009-14009-C02-01 and TIN2009-14009-C02-02 and by the Millennium Institute for Cell Dynamics and Biotechnology (ICDB) (Grant ICM P05-001-F).

1 Introduction

As a consequence of the globalisation and the existence of countries and supra national entities embracing regions with different languages the amount of texts that are stored together with their translation into different languages has dramatically increased. A text placed together with its translation into other languages is referred to as a *multilingual parallel corpus*; a *bilingual parallel corpus*, also know as *bitext*, is a parallel corpus made of two texts. Bitexts are, as stated by Melamed [10], "one of the richest sources of linguistic knowledge because the translation of a text into another language can be viewed as a detailed annotation of what that text means".

The use of bitext by human language technology applications has grown in parallel with the availability of large collections of bitexts. These applications may use compression techniques to solve storage problems and improve access time in storing and processing [13] by trading disk transfer operations for processor operations. In addition, the use of compression techniques reduces transmission times, which increases the efficiency of communications.

Next section explains the main concepts related to the compression of bitexts and gives a brief description of related work. Section 3 covers some modelling policies to represent bitexts, Section 4 then explains a new modelling technique and gives several statistics. Section 5 discusses the results achieved when compressing different bitext corpora. The paper ends with some conclusions.

2 Related Work

A *bitext* consists of two texts that are mutual translations. A considerable part of the information contained in a bitext is highly redundant because the same semantic content is represented in two different ways. Storage and transmission costs are thus unnecessarily increased by storing bilingual versions of the same content. Ideally, this fact could be avoided if perfect machine translation systems were available, since only one version of the text would be needed to produce its translation [11].

A bitext in which the translation relationship between the words in one text (left) and the words in the other text (right) has been established is usually referred to as a *word-aligned bitext*. A way to benefit from the fact that both texts in a bitext (say, L and R) are mutual translations is to know for each segment in L which is its counterpart in R. The *word alignment* task [12] connects words in text L with those in R. The result is a bigraph for the words in L and the words in R with an arc between word $l \in L$ and word $r \in R$ if and only if they are mutual translations. Word alignment is done after *sentence alignment* [7] which, analogously, identifies pairs of sentences that are mutual translations.

Nevill-Manning and Bell [11] explore how traditional text-compression methods can be extended to the compression of bitexts. Their model is based on two types of relations: exact correspondences between two words and synonymy relationships between the words in both texts (as given by a thesaurus).

In [4], *text alignment* is proposed as the basis for bilingual text compression. The alignment-based algorithm requires, in addition to the left (L) and right (R) texts, the existence of word- and phrase-level alignments, the lemmatised forms of L and R, a lemmata dictionary of words in L, a variant dictionary with the lemmata of all words in R and a bilingual glossary. Slight improvements are reported [4] with respect to classical compression algorithms such as BZIP2 or word-oriented Huffman.

3 Modelling Bitexts

To preserve the connection established in the word alignment process which, in turn, allow us to recognise which text segments are mutual translations, a bitext may be represented as a sequence of word pairs (*biwords*) where the first element corresponds to a word in the left text and the second element refers to its counterpart in the right text. An *empty word* (ϵ) is assumed as the counterpart word if a word in one text is not aligned with any word in the other text, or if its alignment was discarded in order to be able to restore the original texts when decompressing. For instance, the biword representation of the Spanish–English bitext *(la casa donde vivimos, the house where we live)* is *(la,the), (casa,house), (donde,where), (ϵ,we) (vivimos,live).*

Using a word-based model enhanced with the empty word symbol may be seen as a reasonable choice to represent a bitext. This implies to perform a pair-to-pair parsing and to identify each word as an independent symbol. These symbols are then encoded in such a way that the codewords assigned to the words in the pair are contiguous in the compressed text. Codeword assignment can be made in two different ways. On one hand, a single vocabulary can be used to represent all the words in the bitext (1V); this is a good choice to compress bitexts made of closely-related languages which share many words. On the other hand, two different vocabularies, one for the words in the left text and another for the words in the right text, may be used. This last method (2V) requires to represent the empty word in both vocabularies. Although these two approaches allow to get compressed bitext representations, they are not effective because the same information is represented twice, enlarging the bitext representation.

To represent the two texts that comprise a bitext on a single model, the concept of *biword* was introduced by Martínez-Prieto et al. [9]. The use of biwords allows to represent with a single symbol two words with high mutual information [5]. The main drawback of using biwords is that larger dictionar-

ies are needed; however, this is not an obstacle to achieve a large spatial saving when a bitext is compressed by using biwords as the symbols to compress.

A biword-based scheme, called 2LCAB, is proposed in [1]. 2LCAB builds a two-level dictionary in which word and biword representations are stored. The aligned bitext representation is pair-to-pair parsed and each word in the pair is then represented in the first-level vocabulary corresponding to its language. Finally, each pair of words (biwords) is represented as a single symbol in the biword vocabulary. The *End Tagged Dense Coding* (ETDC) [3] is used to encode both the words in isolation and the biwords.

In 2LCAB words in the first-level dictionaries are ranked in accordance with the number of different biwords in which they appear. Biwords in the second-level vocabulary are ranked in accordance with their frequency in the bitext. However, no biword strings are used when the biword vocabulary is stored with the compressed bitext, instead the concatenation of the codewords of the two words present in the biword are used. 2LCAB allows the bitext to be processed (searched and retrieved) in its compressed form.

4 First-Order Model on Translations Relationships

Bearing in mind Melamed's [10] affirmation which states that the translation of a text into another language can be viewed as a detailed annotation of what that text means, we propose a new model (TRC_1) for bitexts in agreement with this affirmation. The idea is to represent the words in the right text with respect to the preceding word in the left text, thus, a first-order model based on translation relationships is proposed. In a dictionary we represent all the different words in the left text ranked in accordance with their frequency in the bitext; an independent "translation" dictionary is then associated to each left word in this (left) dictionary. This second dictionary stores all the different words in the right text paired in a biword with the corresponding word in the left dictionary. Words in associated "translation" dictionaries are ranked in accordance with the number of different biwords in which they appear. With the exception of the empty word, each word in a texts is related to a small number of words in the other text and, therefore, one byte should be sufficient to code the right word in each biwords. Figure 1 illustrates the dictionaries used by TRC_1.

While the possible gain of this method is clear when it is used as a preprocessing step to another compressors, as a side effect it requires to encode two symbols (left and right words) instead of just one (biword). It is therefore worth exploring if this permits to achieve better compression ratios since, as a previous step to another compressor, coding biwords with a single symbol results in larger dictionaries and the redundancy loss in the encoded stream may be a handicap for the compressor.

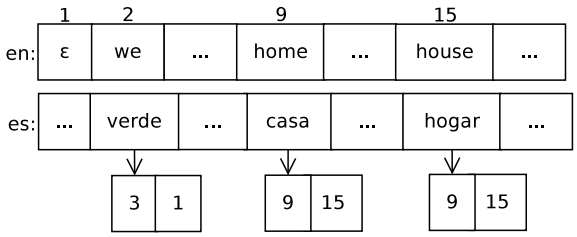

Fig. 1: Dictionaries used by TRC$_1$. The translation dictionaries link the words in the left dictionary (es) with those in the right dictionary (en) by storing the corresponding indexes.

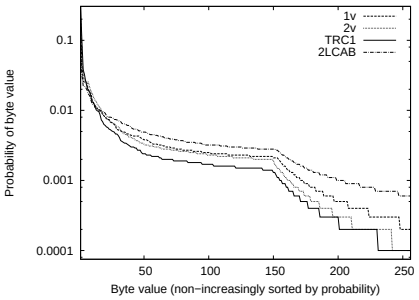

Fig. 2: Relationship between byte values (non-increasingly sorted by probability) from a Spanish–English bitext.

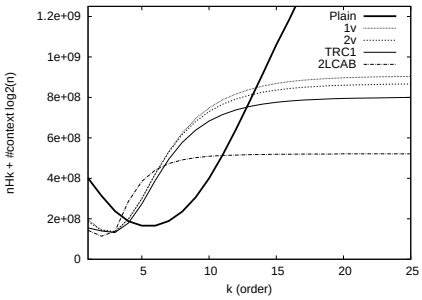

Fig. 3: Values $nH_k + \#\text{contexts} \times \log_2(n)$ for plain and different models codified with ETDC from a Spanish–English bitext.

Figure 2 plots all byte frequencies when Spanish–English bitexts are modelled using the 1V, 2V, 2LCAB and TRC$_1$ aforementioned approaches and their symbols are encoded via ETDC [3]. As can be seen, the byte frequency distribution is very skewed, which suggests compressing this byte stream with a bit-oriented technique. As the order grows, compression improves because the model captures larger correlations between consecutive characters in the text but, unfortunately, the number of different contexts becomes unmanageable. The average length of a word is close to 5 bytes in English texts [2] but the variance is relatively high and is raised if attention is paid to the distance between two consecutive words. A text encoded with ETDC is obviously advantageous because the average length of a word is around 2 bytes with low variance. A k-order modeller can therefore capture the correlation between consecutive words with a much smaller k, or capture longer correlations with a given k [6]. Figure 3 provides a realistic estimation of the size of the compressed bitext achievable by a k-th order compressor. It displays the value nH_k as a function of the number of contexts, where n is the size of the text. Notice that these values are corrected by penalising each new context with $\log(n)$ bits [6].

5 Evaluation

Experiments have been performed in order to test, on one hand, the efficiency of different approaches to compress bitexts and, on the other hand, the effect of the bitext size on the compression ratio. We used the following corpora: (i) a Spanish–Catalan (`es-ca`) bitext from *El Periódico de Catalunya*,[1] a daily newspaper published both in Catalan and Spanish; (ii) a Spanish–Galician (`es-gl`) bitext from *Diario Oficial de Galicia*,[2] the bulletin of the Government of Galicia, published both in Galician and Spanish; and (iii) bitexts for Spanish–English (`es-en`), French–English (`fr-en`), and German–English (`de-en`) from the *European Parliament Proceedings Parallel Corpus* [8].

We created bitexts of different sizes for each language pair in such a way that larger bitexts contain smaller ones. To compute the word alignments we used the largest bitext for each language pair and the GIZA++ statistical aligner [12].[3] An isolated AMD Athlon Dual Core processor at 2 GHz with 2 GB of RAM and running Debian 4 Etch was used. We used a g++-4.1.2 compiler with full optimisation; time results measure CPU user time.

We have tested the two word-based approaches described in Section 3: 1V uses a single vocabulary in which all words in the bitext are represented, whereas 2V considers two vocabularies to store independent representations of the words in each text. Both approaches use the *encode* function defined by `ETDC` [3] for codewords assignment. We do not compare with the approach proposed by Conley and Klein [4] because we have not found any implementation to test it with the bitext collection mentioned above. Anyway, Conley and Klein compare their TRANS approach with GZIP and BZIP2 and they conclude that TRANS is slightly better that BZIP2 (an improvement of 1% is achieved). However, the authors do not consider the size of the auxiliary files that TRANS requires to decompress the bitext.

Table 1 shows the compression ratios achieved when 1V, 2V, 2LCAB and TRC_1 are used as compression boosters; in this case, they are used as a preprocessing step to the well-know GZIP, BZIP2 and PPMDI compressors. Results show that 1V+GZIP, 2V+GZIP, 2LCAB+GZIP and TRC_1+GZIP obtain the worst compression ratios in their category. However, for large bitexts consisting of closely-related languages they improve PPMDI up to 4.5 percentage points (they can be directly compared), and BZIP2 up to 6.3 percentage points. When the bitexts consist of less-related languages the improvement is reduced to 2.1 percentage points for PPMDI and to 3.8 percentage points for BZIP2. 1V+PPMDI, 2V+PPMDI, 2LCAB+PPMDI and TRC_1+PPMDI obtain the best compression ratios, improving PPMDI up to 4.0 percentage points.

Compression ratios obtained when 2LCAB and TRC_1 are used as a preprocessing step are very similar, although TRC_1 approximately encodes twice as

[1] `http://www.elperiodico.com`

[2] `http://www.xunta.es/diario-oficial`

[3] `http://code.google.com/p/giza-pp/`

	MB	1V			2V			2LCAB			TRC1		
		GZIP	BZIP2	PPMDI	GZIP	BZIP2	PPMDI	GZIP	BZIP2	PPMDI	GZIP	BZIP2	PPMDI
es-gl	1	12.43%	9.84%	8.72%	14.16%	11.38%	10.12%	12.88%	11.61%	10.82%	13.27%	10.76%	9.59%
	10	10.77%	7.42%	6.88%	11.61%	8.09%	7.47%	9.16%	7.35%	6.90%	9.96%	7.24%	6.52%
es-ca	1	28.59%	23.86%	21.82%	29.79%	25.08%	23.24%	28.63%	26.23%	25.28%	28.13%	24.24%	22.34%
	10	23.42%	18.21%	17.81%	23.94%	18.69%	18.20%	19.61%	17.46%	16.65%	20.18%	16.98%	15.76%
	100	21.34%	16.52%	16.24%	21.53%	16.65%	16.30%	15.67%	13.87%	13.25%	16.87%	14.07%	13.11%
es-en	1	26.36%	23.43%	21.30%	25.52%	23.19%	21.00%	26.73%	25.43%	24.09%	26.22%	24.33%	22.12%
	10	22.94%	19.53%	18.34%	22.06%	19.29%	17.97%	20.67%	19.20%	18.29%	21.22%	19.31%	17.74%
	100	21.27%	17.97%	16.95%	20.43%	17.70%	16.58%	17.36%	16.11%	15.40%	18.68%	16.91%	15.56%
fr-en	1	25.69%	22.82%	20.71%	25.06%	22.73%	20.59%	26.18%	24.90%	23.59%	25.76%	23.82%	21.66%
	10	22.57%	19.17%	17.99%	21.79%	19.02%	17.71%	20.34%	18.83%	17.96%	20.93%	18.95%	17.43%
	100	21.31%	18.09%	17.03%	20.49%	17.84%	16.67%	17.54%	16.25%	15.56%	18.83%	17.03%	15.69%
de-en	1	26.93%	24.08%	21.88%	25.94%	23.71%	21.49%	27.38%	25.95%	24.61%	26.92%	24.92%	22.60%
	10	23.44%	20.07%	18.88%	22.49%	19.85%	18.45%	21.32%	19.76%	18.84%	21.90%	19.87%	18.22%
	100	21.90%	18.90%	17.76%	21.00%	18.61%	17.34%	18.30%	17.07%	16.35%	19.61%	17.86%	16.42%

Table 1: Compression ratios achieved by 1V, 2V, 2LCAB and TRC1 when they are used as compression boosters.

MB	GZIP	BZIP2	PPMDI	2LCAB			TRC1		
				GZIP	BZIP2	PPMDI	GZIP	BZIP2	PPMDI
es-ca									
1	0.23/0.04	0.77/0.25	0.91/0.98	0.37/0.09	0.55/0.17	1.09/1.03	0.50/0.10	0.65/0.19	1.15/1.04
10	1.94/0.25	4.58/1.94	5.31/5.84	2.70/0.44	3.59/1.34	6.22/5.50	3.30/0.47	4.18/1.58	6.85/5.90
100	9.60/1.88	35.34/12.19	38.83/41.46	10.26/2.53	17.43/6.92	31.36/30.22	15.67/3.61	22.32/8.80	37.14/33.19
es-en									
1	0.23/0.03	0.73/0.25	0.87/0.95	0.33/0.06	0.50/0.16	0.97/0.89	0.43/0.06	0.57/0.18	0.97/0.86
10	1.85/0.18	4.40/1.76	5.43/5.56	2.27/0.41	3.33/1.29	6.06/5.53	3.10/0.48	4.19/1.53	6.02/5.45
100	12.11/1.90	41.68/14.70	46.98/50.22	12.06/2.84	19.94/7.63	37.59/35.67	17.72/3.25	24.95/8.96	42.13/36.41

Table 2: Compression/Decompression times achieved with different bitext corpora. Word alignment times are not taken into account.

symbols as 2LCAB. In fact, TRC1 is better than 2LCAB for small bitexts and for bitexts consisting of closely-related languages.

Table 2 shows compression and decompression times, respectively; word alignment times are note taken into account. Due to lack of space we only show 2LCAB and TRC1 as boosters times, however all the times obtained by using 1V and 2V as boosters are similar to those of TRC1. Results show that the techniques we propose are very fast at compression and, mainly, at decompression. This is explained by the fact that both BZIP2 and PPMDI are very slow and when they compress the transformed text they are actually compressing 30% of the plain text. As a result, not only 2LCAB+BZIP2 and 2LCAB+PPMDI compress more than BZIP2 and PPMDI respectively, but also they are up to 40% faster.

6 Conclusions

We have shown and analysed some modelling proposals that use words (1V, 2V and TRC_1) and biwords (2LCAB) as symbols. When a bitext is modelled with the aforementioned techniques and encoded with a well-know byte-oriented code, as ETDC, acceptable compression ratios are obtained. In addition, they can be seen as a transformation of the bitext that boost general purpose compressors because they transforms a bitext in a shorter byte sequence (19-50% of the original) that can still be compressed. On the other hand, we have shown that the compression ratios reported are coherent with the statistics of the representation model and the previous analytical study.

When encoded models are used as a preprocessing step to general-purpose compressors, the experiments show that they improve the compression ratio as well as their performance in both compression and decompression yielding an attractive space/efficiency trade-off.

References

1. J. Adiego, N. R. Brisaboa, M. A. Martínez-Prieto, and F. Sánchez-Martínez. A two-level structure for compressing aligned bitexts. In *Proceedings of the 16th International Symposium on String Processing and Information Retrieval*, pages 114–121, Saariselkä, Finland, 2009.
2. T. C. Bell, J. G. Cleary, and I. H. Witten. *Text Compression*. Prentice Hall, Englewood Cliffs, N.J., 1990.
3. N. R. Brisaboa, A. Fariña, G. Navarro, and J. R. Paramá. Lightweight natural language text compression. *Information Retrieval*, 10(1):1–33, 2007.
4. E. S. Conley and S. T. Klein. Using alignment for multilingual text compression. *International Journal of Foundations of Computer Science*, 19(1):89–101, 2008.
5. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.
6. A. Fariña, G. Navarro, and J. Paramá. Word-based statistical compressors as natural language compression boosters. In *Proc. 18th Data Compression Conference (DCC)*, pages 162–171, 2008.
7. W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.
8. P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. `http://www.statmt.org/europarl/`.
9. M. A. Martínez-Prieto, J. Adiego, F. Sánchez-Martínez, P. de la Fuente, and R. C. Carrasco. On the use of word alignments to enhance bitext compression. In *Data Compression Conference*, page 459, 2009.
10. I. D. Melamed. *Emplirical methods for exploting parallel texts*. MIT Press, 2001.
11. C. G. Nevill-Manning and T. C. Bell. Compression of parallel texts. *Information Processing & Management*, 28(6):781–794, 1992.
12. F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.
13. N. Ziviani, E. Moura, G. Navarro, and R. Baeza-Yates. Compression: A key for next-generation text retrieval systems. *IEEE Computer*, 33(11):37–44, November 2000.