Generalized Biwords for Bitext Compression and Translation Spotting: Extended Abstract*

Felipe Sánchez-Martínez,[†] **Rafael C. Carrasco,**[†] **Miguel A. Martínez-Prieto**[‡] and **Joaquín Adiego**[‡]
[†] Dep. de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, E-03071, Spain
{*fsanchez,carrasco*}@*dlsi.ua.es*
[‡] Dep. de Informática, Universidad de Valladolid, E-47011, Spain
{*migumar2,jadiego*}@*infor.uva.es*

1 Introduction

The increasing availability of large collections of *bilingual parallel corpora* has fostered the development of natural-language processing applications that address bilingual tasks, such as corpus-based machine translation, the automatic extraction of bilingual lexicons, and translation spotting [Simard, 2003]. A bilingual parallel corpus, or *bitext*, is a textual collection that contains pairs of documents which are translations of one another. In the words of Melamed [2001, p. 1], "bitexts are 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".

Large bitexts are usually available in a compressed form in order to reduce storage requirements, to improve access times [Ziviani *et al.*, 2000], and to increase the efficiency of transmissions. However, the independent compression of the two texts of a bitext is clearly far from efficient because the information contained in both texts is redundant. Previous work [Nevill-Manning and Bell, 1992; Conley and Klein, 2008; Martínez-Prieto *et al.*, 2009; Adiego *et al.*, 2009; 2010] has shown that bitexts can be more efficiently compressed if the fact that the two texts are mutual translations is exploited.

Martínez-Prieto *et al.* [2009], and Adiego and his colleagues [2009; 2010] propose the use of *biwords* —pairs of words, each one from a different text, with a high probability of co-occurrence— as input units for the compression of bitexts. This means that a biword-based intermediate representation of the bitext is obtained by exploiting alignments, and unaligned words are encoded as pairs in which one component is the empty string. Significant spatial savings are achieved with this technique [Martínez-Prieto *et al.*, 2009], although the compression of biword sequences requires larger dictionaries than the traditional text compression methods.

The biword-based compression approach works as a simple processing pipeline consisting of two stages (see Figure 1). After a text alignment has been obtained without pre-existing linguistic resources, the first stage transforms the bitext into a biword sequence. The second stage then compresses this sequence. Decompression works in reverse order: the biword sequence representing the bitext is first generated from the compressed file, and the original texts are then restored from this sequence.

The biword sequences obtained with the former biword-based compression methods contain a large fraction (between 10% and 60%, depending on the language pair) of *unpaired words*, that is, biwords of which one of the words in the pair is the empty word ϵ. The unpaired words are generated in three different cases:

- The aligner is unable to connect a word with any of the words in the parallel text because, for example, infrequent idiomatic expressions or free translations have been found.
- The aligner generates a one-to-many alignment because a word has been translated into a multiword expression. For instance, if the Spanish word *volver* is translated into English as *to go back*, the biword extractor has to select one of the links, build a pair of words from that link, and leave the other words unpaired.
- The aligner generates some crossing alignments as a result of word reordering in the translation. For instance, in Figure 2, either the pair $(\mathrm{verde}, \mathrm{green})$ or the pair $(\mathrm{casa}, \mathrm{house})$ must be ignored by the biword extractor, thus leaving two unpaired words; otherwise, the information provided by the sequence will not be sufficient to retrieve both texts in the original order.

The last two sources of unpaired words are responsible for the different spatial savings reported by Martínez-Prieto *et al.* [2009] for bitexts consisting of closely-related languages (e.g., Spanish and Portuguese) and for those involving divergent language pairs (e.g., French and English), in which word reorderings and multiword translations are frequent.

2 Overview

We describe and evaluate the simple biword extraction approach, and compare it with other schemes used to generate generalized biword sequences that maintain all or part of the structural information provided by the aligner. A biword essentially becomes a word from one text of the bitext (left word) connected with a variable number of words from the other text of the bitext (right words) plus additional information concerning the relative position of each right word with regard to the preceding one. The fraction of unpaired words is thus reduced, and better compression ratios can be obtained.

[*] This paper is an extended abstract of the JAIR publication [Sánchez-Martínez *et al.*, 2012].

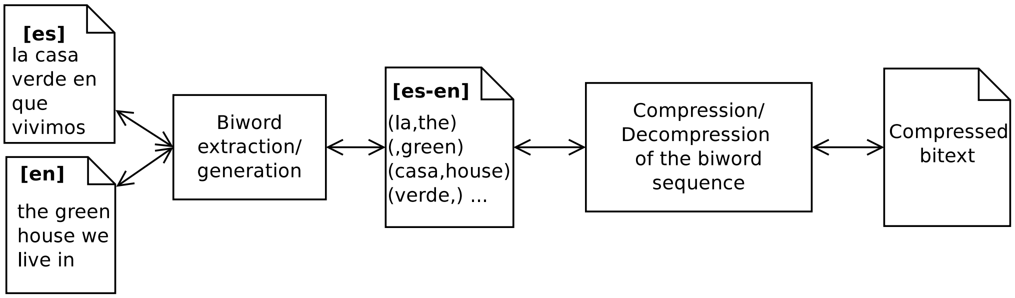

Figure 1: Processing pipeline of a biword-based bitext compression approach.

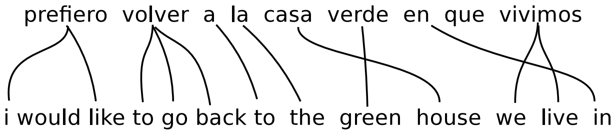
prefiero volver a la casa verde en que vivimos
i would like to go back to the green house we live in

Figure 2: Example of a Spanish–English pair of sentences with one-to-many word alignments.

We also show that this generalization of biwords allows for the implementation of an efficient *translation spotting* [Simard, 2003] algorithm on the compressed bitext; a task that consists of identifying the words (or text segments) in the other text that are the translation of the words in the query. Indeed, generalized biword sequences contain all the information needed in order to retrieve connected passages.

Generalized biwords can also be used as an ingredient in the bilingual language model employed in some statistical machine translation systems [Koehn, 2010]. For instance, Mariño *et al.* [2006] use bilingual n-grams and consider the translation as a bilingual decoding process. Casacuberta and Vidal [2004] also exploit bilingual n-grams but apply stochastic finite-state transducers to this task. In both cases, the local reordering of words is addressed by considering multiword segments of source and target words as the fundamental translation units. Some alternative approaches [Niehues *et al.*, 2011; Hasan *et al.*, 2008] integrate bilingual language models as an additional feature in the decoding function that drives the statistical translation process. However, none of the approaches mentioned includes the structural information provided by the aligners as part of the bilingual language model.

In addition, the improvement in compression performance obtained when taking advantage of the fact that the two texts in a bitext are mutual translations may be regarded as an indication of the quality of word alignments [Och and Ney, 2003]. This indicator, which bounds the mutual information [Cover and Thomas, 1991] in the two texts of a bitext, does not require a manually-annotated corpus to evaluate the automatic alignment.

3 Extraction of Biword Sequences

Before extracting the sequence of biwords representing a bitext, the alignments between the words in the left text $L = l_1 l_2 \cdots l_M$ and the words in the right text $R = r_1 r_2 \cdots r_N$ must be established. For this purpose we have used the open-source GIZA++ toolkit[1] [Och and Ney, 2003].

The result of *word alignment* is a bigraph $G = \{L, R, A\}$ in which an edge $\{l_i, r_j\} \in A$ between word $l_i \in L$ and word $r_j \in R$ signifies that they are mutual translations according to the translation model used by the aligner. These complex structures are processed by splitting the bigraph into connected components: each connected component is either an unpaired (right or left) word, or a left word σ aligned with a sequence ρ of (one or more) right words. A connected component including the structural information needed to place all the words in their original positions in the bitext is what we term as a *generalized biword*.

In order to build a sequence $\mathcal{B}$ of generalized biwords, biwords are sorted primarily according to their left component σ and, secondarily, by the head of their right component ρ.

Every *generalized biword* $\beta = (\sigma, \rho, \omega)$ in the sequence $\mathcal{B}$ consists of:

- a string σ in Σ_L (the set of different words in the left text L enhanced with the empty word ϵ),
- an array of strings ρ in Σ_R (the set of subsequences in the right text R plus the empty subsequence), and
- an integer array ω, with one offset per string in ρ, that stores the structural information needed to place each word in ρ in its original position.

Figure 3 shows the sequence of generalized biwords generated from the word-aligned sentence shown in Figure 2. The offset in the biword `(casa, (house),(1))` signifies that there is a one-word gap between `house` and `the` which is occupied by the word `green` with offset 0 in `(verde,(green),(0))`. The offsets in `(vivimos, (we,live), (0,0))` indicate that `we` comes directly after the word `house` and `live` comes immediately after `we`.

Henceforth, we shall call *biwords with shifts* those biwords with at least one non-null offset (*biwords without shifts*, otherwise). We shall further differentiate between *biwords with simple shifts*, where only the first offset is non-null, and *biwords with complex shifts*, with non-consecutive words in R.

4 Compression of Biword Sequences

We evaluate two different methods, namely TRE and 2LCAB, to compress the intermediate representation introduced in the

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

```
(prefiero,(i,like),(0,1))
(ϵ,(would),(0))
(volver,(to,go,back),(0,0,0))
(a,(to),(0))
(la,(the),(0))
(casa,(house),(1))
(verde, (green),(0))
(en,(in),(2))
(que,(),())
(vivimos,(we,live),(0,0))
```

Figure 3: Generalized biword sequence for the word-aligned sentence shown in Figure 2.

previous section.

The *Translation Relationship-based Encoder* (TRE) assigns codewords to the left word and to the sequences of right words in the biword through the use of two independent methods. The left text is encoded using word-based Huffman coding [Moffat, 1989], whereas the right text is encoded by using the left text as its context. To do this, TRE uses three dictionaries: one, Σ_L, with the left words, a second one, Σ_R, with the sequences of right words, and a third one, $\tau_{\mathcal{B}}$, which maps each word $\sigma \in \Sigma_L$ onto the subset of entries in Σ_R with which it has been aligned in the corpus.

In contrast to TRE, the 2-Level Compressor for Aligned Bitexts (2LCAB; [Adiego *et al.*, 2009]) encodes every biword with a single codeword based on a two-level dictionary. The first level consists of two dictionaries, Σ_L and Σ_R, containing the left words and the sequences of right words, respectively, that appear in the biword sequence $\mathcal{B}$. The second level dictionary Σ_B stores the different biwords in $\mathcal{B}$ as an integer sequence of alternating references to the entries in Σ_L and Σ_R. The text in the sequence $\mathcal{B}$ can then be mapped onto a sequence of references to entries in Σ_B.

Both methods use *prediction by partial matching* (PPM; [Cleary and Witten, 1984]) to encode the dictionaries Σ_L and Σ_R, and encode the offsets as two streams of integer values: one with the relative positions $\mathcal{P}$ of the biwords with shifts in the sequence $\mathcal{B}$, and another with the offset values $\mathcal{O}$ for the biwords with shifts. Both streams are therefore encoded by using two independent sets of Huffman codewords.

5 Compression Results

We evaluate the performance of the bitext compressors based on generalized biwords with nine different language pairs (Spanish–Catalan, `es-ca`; Welsh–English, `cy-en`; German–English, `de-en`; Spanish–English, `es-en`; Spanish–French, `es-fr`; Spanish–Italian, `es-it`; Spanish–Portuguese, `es-pt`; French–English, `fr-en`; and Finnish–English, `fi-en`) when they are used in combination with four different methods to extract a sequence of biwords:

- `1:N Complex`: the one-to-many word alignments generated by GIZA++ are used to generate a sequence of generalized biwords.
- `1:N Simple`: the biwords with complex shifts generated by the one-to-many alignments provided by GIZA++ are split into biwords with simple shifts plus unpaired words; the result is a sequence of biwords with simple shifts or without shifts.
- `1:1 Non-monotonic`: one-to-one word alignments are obtained by computing the intersection of the alignments produced by GIZA++ when the left and the right text are exchanged; the result is a sequence of biwords whose right component contains, at most, one word (and these biwords cannot, therefore, have complex shifts).
- `1:1 Monotonic`: the 1:1 non-monotonic sequence is transformed into a sequence of biwords without shifts by splitting biwords with shifts into unpaired words.

The last method, `1:1 Monotonic`, does not use the enhancement provided by the generalization of biwords (i.e., the structural information), and is therefore equivalent to the basic procedures described earlier [Martínez-Prieto *et al.*, 2009; Adiego *et al.*, 2009; 2010].

Both TRE and 2LCAB outperform general-purpose compressors in all cases but that of the `en-fi` pair. This suggests that TRE and 2LCAB take advantage of the fact that the texts contain the same information but "encoded" in different languages, particularly in the case of highly parallel bitexts (`en-cy`) and languages with a high syntactic correlation (`es-ca`). The low performance for `en-fi` is the consequence of the larger translation dictionaries ($\tau_{\mathcal{B}}$) used by TRE, and the larger bilingual dictionary (Σ_B) used by 2LCAB, in comparison to the other language pairs. Furthermore, the percentage of unpaired words is also higher than that of the other language pairs.

The best results are obtained in most of the cases when one-to-one alignments are used with both techniques. This is due to the fact that the use of one-to-many alignments causes the size of the dictionary to grow considerably, particularly in the case of the biword dictionary used by 2LCAB. This side effect can be alleviated by discarding very infrequent biwords by splitting them into smaller, more frequent, biwords.

Discarding the most infrequent biwords (about two thirds of them) usually leads to an improvement in the compression ratios, except in the case of very similar languages, such as Catalan and Spanish, in which the translation is highly parallel. This effect is more important in the case of 2LCAB because the pruning leads to a large reduction in the size of the biword dictionary and this compensates the small increment in the total number of biwords needed to represent the bitext (between 5% and 10% of increment depending on the method used for its generation). With this filtering, 2LCAB and TRE obtain the best results when extracting the biword sequence with method `1:N Simple`.

6 Translation Spotting with Compressed Bitexts

The exploitation of bitexts by computer-aided translation tools has evolved from simple *bilingual concordancers* [Bowker and Barlow, 2004] that can only provide a whole sentence as the result of a translation query, to advanced *translation search engines* [Callison-Burch *et al.*, 2005a;

Bourdaillet *et al.*, 2010] with translation spotting capabilities, i.e. they can retrieve parallel text segments in bitexts.

It would seem that existing translation search engines [Callison-Burch *et al.*, 2005a; Bourdaillet *et al.*, 2010] do not access bitexts in their compressed forms because storing the correspondences between the translated segments requires additional data structures such as word indexes or suffix arrays [Lopez, 2007; Callison-Burch *et al.*, 2005b]; suffix arrays typically require four times the size of the text [Manber and Myers, 1993]. In contrast, the generalized biwords require much less space, they integrate the alignment information into the compressed bitext, and this information can be exploited to retrieve translation examples.

We apply the 2LCAB compression technique to the translation spotting task after replacing the use of Huffman codewords and PPM compression by the use of *End-Tagged Dense Code* (ETDC; [Brisaboa *et al.*, 2007]), *succinct data structures* [Navarro and Mäkinen, 2007, Sec. 6], and *directly addressable variable-length codes* (DAC, [Brisaboa *et al.*, 2009]). These changes need to be introduced to allow both direct searching and random access to the compressed text.

6.1 Translation Spotting

The searchable 2LCAB summarized above is complemented with a search algorithm which, given a single word w in the left text, proceeds as follows:

1. The word w is looked for in Σ_L and its ETDC codeword c is obtained.
2. An exact pattern-matching algorithm identifies all the occurrences of the codeword c in the biword dictionary Σ_B, and the codeword of the biwords where c happens to refer to the left word being looked for are added to the search set Z.
3. The multi-pattern matching algorithm Set-Horspool [Horspool, 1980; Navarro and Raffinot, 2002] locates all the codewords in the sequence of biwords $\mathcal{B}$ that match one of those contained in Z, and the matching positions are added to a new set M.
4. For every match $m \in M$, the adjacent right component is read from $\mathcal{B}$ and the offsets, if any, are recovered from $\mathcal{O}$ and used to place the right words in the original order.

In case the query consists of a sequence of words, the Set-Horspool algorithm is executed only for the word in the sequence generating the smallest set of codewords to locate Z, and the remaining words are then used to filter the results once the biword context has been retrieved.

Table 1 shows an actual example of the output obtained for a multiple word query and a compressed biword sequence obtained with the `1:N Complex` method. Note that the third match shows a non-contiguous translation, a case which cannot be retrieved with the original 2LCAB implementation [Adiego *et al.*, 2009].

Left text:	[. . .] all the information they need **in order to perform** their civic duties in society .
Right text:	[. . .] acceder a la información necesaria **para actuar** como ciudadanos en sus sociedades .
Left text:	[. . .] the information and intelligence they need **in order to perform** their tasks .
Right text:	[. . .] la información y los datos que necesiten **para poder realizar** su trabajo .
Left text:	the co-decision procedure must be used **in order to perform** this legislative work [. . .]
Right text:	**para que** ese trabajo legislativo se **realice** en condiciones [. . .]
Left text:	[. . .] what are the procedures , that we need **in order to perform** them ?
Right text:	[. . .] los procedimientos que necesitamos **para ejecutarlas** ?

Table 1: Output obtained after the query "*in order to perform*" on the bitext compressed with the `1:N Complex` method. The query terms and their translations are spotted in boldface.

6.2 Experimental Evaluation

The compression ratios obtained with the searchable 2LCAB method are worse tan those obtained with the original 2LCAB method summarized in Section 4: it achieves compression ratios which are slightly worse than those obtained with general purpose and word-based compressors. However, these compressed files include the information concerning the alignments between the words, information that is not included in the files compressed with the standard compressors but is necessary to perform translation spotting.

We have studied how the time needed to process a query depends on the language pair and also on the number and frequencies of the words in the query. The results show that the time required to process a query depends on the quality of the alignments and on the degree of parallelism of the bitexts. Poor quality alignments makes words participate in a large number of different biwords, and bitexts with a highly parallel structure make words participate only in a small number of biwords. The larger the number of biwords in which a word participates, the lower the performance of the Set-Horspool algorithm.

Acknowledgments

This work has been supported by the Spanish Government through projects TIN2009-14009-C02-01 and TIN2009-14009-C02-02.

References

[Adiego *et al.*, 2009] 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 String Processing and Information Retrieval Symposium*, pages 114–121, Saariselkä, Finland, 2009.

[Adiego *et al.*, 2010] J. Adiego, M.A. Martínez-Prieto, J.E. Hoyos-Torio, and F. Sánchez-Martínez. Modelling parallel texts for boosting compression. In *Proceedings of the 2010 Data Compression Conference*, page 517, Snowbird, USA, 2010.

[Bourdaillet *et al.*, 2010] J. Bourdaillet, S. Huet, P. Langlais, and G. Lapalme. TransSearch: from a bilingual concordancer to a translation finder. *Machine Translation*, 23(3–4):241–271, 2010. Published in 2011.

[Bowker and Barlow, 2004] L. Bowker and M. Barlow. Bilingual concordancers and translation memories: a comparative evaluation. In *Proceedings of the Second International Workshop on Language Resources for Translation Work, Research and Training at Coling 2004*, pages 70–79, Geneva, Switzerland, 2004.

[Brisaboa *et al.*, 2007] N.R. Brisaboa, A. Fariña, G. Navarro, and J.R. Paramá. Lightweight natural language text compression. *Information Retrieval*, 10(1):1–33, 2007.

[Brisaboa *et al.*, 2009] N. Brisaboa, S. Ladra, and G. Navarro. Directly addressable variable-length codes. In *Proceedings of the 16th String Processing and Information Retrieval Symposium*, pages 122–130, Saariselkä, Finland, 2009.

[Callison-Burch *et al.*, 2005a] C. Callison-Burch, C. Bannard, and J. Schroeder. A compact data structure for searchable translation memories. In *Proceedings of the 10th European Association for Machine Translation Conference*, pages 59–65, Budapest, Hungary, 2005.

[Callison-Burch *et al.*, 2005b] C. Callison-Burch, C. Bannard, and J. Schroeder. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 255–262, Ann Arbor, USA, 2005.

[Casacuberta and Vidal, 2004] F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

[Cleary and Witten, 1984] J.G. Cleary and I.H. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32(4):396–402, 1984.

[Conley and Klein, 2008] 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.

[Cover and Thomas, 1991] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[Hasan *et al.*, 2008] S. Hasan, J. Ganitkevitch, H. Ney, and J. Andrés-Ferrer. Triplet lexicon models for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods on Natural Language Processing*, pages 372–381, Honolulu, USA, 2008.

[Horspool, 1980] R. N. Horspool. Practical fast searching in strings. *Software: Practice and Experience*, 10(6):501–506, 1980.

[Koehn, 2010] P. Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.

[Lopez, 2007] A. Lopez. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 976–985, Prague, Czech Republic, 2007.

[Manber and Myers, 1993] U. Manber and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[Mariño *et al.*, 2006] J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, and M.R. Costa-Jussà. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549, 2006.

[Martínez-Prieto *et al.*, 2009] 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 *Proceedings of the 2009 Data Compression Conference*, page 459, Snowbird, USA, 2009.

[Melamed, 2001] I.D. Melamed. *Emplirical methods for exploting parallel texts*. MIT Press, 2001.

[Moffat, 1989] A. Moffat. Word-based text compression. *Software: Practice and Experience*, 19(2):185–198, 1989.

[Navarro and Mäkinen, 2007] G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Computing Surveys*, 39(1), 2007. Article 2.

[Navarro and Raffinot, 2002] G. Navarro and M. Raffinot. *Flexible Pattern Matching in String: Practical on-line search algorithms for texts and biological sequences*. Cambridge University Press, 2002.

[Nevill-Manning and Bell, 1992] C.G. Nevill-Manning and T.C. Bell. Compression of parallel texts. *Information Processing & Management*, 28(6):781–794, 1992.

[Niehues *et al.*, 2011] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel. Wider context by using bilingual language models in machine translation. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 198–206, Edinburgh, UK, 2011.

[Och and Ney, 2003] F.J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[Sánchez-Martínez *et al.*, 2012] F. Sánchez-Martínez, R.C. Carrasco, M.A. Martínez-Prieto, and J. Adiego. Generalized biwords for bitext compression and translation spotting. *Journal of Artificial Intelligence Research*, 43:389–418, 2012.

[Simard, 2003] M. Simard. Translation spotting for translation memories. In *Proceedings of NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 65–72, Edmonton, AB, Canada, 2003.

[Ziviani *et al.*, 2000] 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.