PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 97 APRIL 2012 43–53

BIA: a Discriminative Phrase Alignment Toolkit

Patrik Lambert[a], Rafael E. Banchs[b]

[a] LIUM, LUNAM Université, University of Le Mans
[b] Institute for Infocomm Research

Abstract

In most statistical machine translation systems, bilingual segments are extracted via word alignment. However, word alignment is performed independently from the requirements of the machine translation task. Furthermore, although phrase-based translation models have replaced word-based translation models nearly ten years ago, word-based models are still widely used for word alignment. In this paper we present the BIA (BIlingual Aligner) toolkit, a suite consisting of a discriminative phrase-based word alignment decoder based on linear alignment models, along with training and tuning tools. In the training phase, relative link probabilities are calculated based on an initial alignment. The tuning of the model weights may be performed directly according to machine translation metrics. We give implementation details and report results of experiments conducted on the Spanish–English Europarl task (with three corpus sizes), on the Chinese–English FBIS task, and on the Chinese–English BTEC task. The BLEU score obtained with BIA alignment is always as good or better than the one obtained with the initial alignment used to train BIA models. In addition, in four out of the five tasks, the BIA toolkit yields the best BLEU score of a collection of ten alignment systems. Finally, usage guidelines are presented.

1. Introduction

Most statistical machine translation (SMT) systems (*e.g.* phrase-based, hierarchical, n-gram-based) build their translation models from word alignment trained in a previous stage. Many papers have shown that intrinsic alignment quality is poorly correlated with MT quality (for example, Vilar et al. (2006)). Accordingly, some research has attempted to tune the alignment directly according to specific MT evaluation metrics (Lambert et al., 2007). Furthermore, although phrase-based transla-

Corresponding author: patrik.lambert@lium.univ-lemans.fr
Cite as: Patrik Lambert, Rafael E. Banchs. BIA: a Discriminative Phrase Alignment Toolkit. The Prague Bulletin of Mathematical Linguistics No. 97, 2012, pp. 43–53. doi: 10.2478/v10108-012-0003-z.

tion models have replaced word-based translation models nearly ten years ago, word-based models are still widely used for word alignment.

In this paper we present the BIA (BIlingual Aligner) toolkit, a suite consisting of a discriminative phrase-based word alignment decoder based on linear alignment models (Moore, 2005; Liu et al., 2005, 2010), along with training and tuning tools. Thus this toolkit allows one to overcome the limitations of most current word alignment systems: the basic alignment unit is not a single word but a phrase (a group of consecutive words),[1] and it provides tools to tune the alignment model parameters directly according to MT metrics. Although currently these tuning tools are implemented to work with the Moses phrase-based decoder (Koehn et al., 2007), it is straightforward to extend them to work with other MT systems (hierarchical, n-gram-based, etc.).

The paper is organised as follows. In Section 2, we present the alignment algorithm and the tuning procedure. In Section 3, we detail how we implemented the different parts of the alignment system presented in Section 2. Then in Section 4, we report results of experiments conducted on the Spanish–English Europarl task (with three corpus sizes), on the Chinese–English FBIS task, and on the Chinese–English BTEC task. In Section 5, we give instruction for training, tuning and decoding with our toolkit. Finally, some conclusions are provided.

2. Phrase-based Discriminative Alignment System

2.1. Alignment Algorithm

This aligner implements a linear combination of feature functions calculated at the sentence pair level. It searches the alignment hypothesis $\hat{a}$ which maximises this linear combination, as expressed in (1):

$$\hat{a} = \arg\max_{a} \sum_{m} \lambda_m h_m(s, t, a), \tag{1}$$

where s, t and a refer respectively to the source sentence, the target sentence and the alignment hypothesis, h stands for the feature functions used and the λs are their corresponding weights. It follows a two-pass strategy, as proposed by Moore (2005). The initial alignment may be computed using BIA with a first set of features (Lambert and Banchs, 2008), or with any other alignment system. In the experiments presented in Section 4, we actually took as initial alignment the combination of the IBM Model 4 source–target and target–source alignments with the "grow-diag-final-and" heuristic (Koehn et al., 2003). This initial alignment was used to calculate the following improved features:

[1] The output of BIA is nevertheless an alignment at the word level, that is, in many SMT systems, the step previous to phrase-pair extraction.

- a phrase association score model with relative link probabilities (Melamed, 2000). These links are between phrases (although in practice most phrases are of length one, *i.e.* single words).
- source and target *word* fertility models giving the probability for a given *word* to have one, two, three or more than three links.

These improved features, together with the following features, were used to align the corpus in a second pass:

- A link bonus model, proportional to the number of links in a.
- Two distortion models, counting respectively the number and amplitude (the difference between target word positions) of crossing links.
- A 'gap penalty' model, proportional to the number of embedded positions between two target words linked to the same source words, or between two source words linked to the same target words.

To find the best hypothesis, we implemented a beam-search algorithm based on dynamic programming (see Section 3).

2.2. Weight Optimisation According to BLEU Score

The alignment weights λ of Equation 1 are optimised so as to maximise the BLEU score calculated on a parallel development corpus (with no alignment annotations), as proposed by Lambert *et. al* (2007).

The optimisation algorithm (presented in Section 2.3) adjusts the weights so as to maximise the BLEU score. At each iteration, the training corpus is aligned as described in Section 2.1. This alignment is used to build an SMT system, including bilingual phrase extraction, translation model(s) estimation and MERT (Och, 2003). Then the development corpus is translated with this SMT system and the BLEU score is computed. Two different development sets can be used for the alignment weight optimisation (Dev) and the MERT process performed at each iteration (MERT Dev).

Note that it would be straightforward to introduce MT metrics other than the BLEU score, and that it would be easy to implement a supervised weight optimisation procedure (for example according to F-score).

2.3. Optimisation Algorithm

The available optimiser is the SPSA algorithm (Spall, 1992). The SPSA (Simultaneous Perturbation Stochastic Approximation) is a stochastic implementation of the conjugate gradient method which requires only two evaluations of the objective function, regardless of the dimension of the optimisation problem. The SPSA procedure is in the general recursive stochastic approximation form, as shown in 2:

$$\hat{\lambda}_{k+1} = \hat{\lambda}_k - \alpha_k \hat{g}_k(\hat{\lambda}_k) \tag{2}$$

where $\hat{g}_k(\hat{\lambda}_k)$ is the estimate of the gradient $g(\lambda) \equiv \partial E/\partial\lambda$ at the iterate $\hat{\lambda}_k$ based on the previous evaluations of the objective function. α_k denotes a positive number that usually decreases as k increases. In the default settings, the gradient is computed with a one-sided approximation which, given $E(\hat{\lambda}_k)$, requires the evaluation of $E(\hat{\lambda}_k +$ perturbation$)$. The original SPSA algorithm has been adapted to achieve convergence after typically 60 to 100 evaluations of the objective function. Note that in general, this algorithm converges to a local minimum.

3. Implementation

The BIA toolkit is implemented in C++ (with the Standard Template Library) and Perl and contains:

- training tools (mostly in C++).
- an alignment decoder (in C++).
- tools to tune the alignment model parameters directly according to MT metrics (in Perl).
- Perl scripts which pilot the training, tuning and decoding tasks.
- a sample (bash) shell script to run the whole pipeline (the same as the one used to produce the results of Section 4, but with sample data).

Although the BIA toolkit uses the Moses toolkit (Koehn et al., 2007) by default for two tasks, it is straightforward to use other tools instead. First, the initial alignment used to train BIA models (see Section 2.1) is by default the "grow-diag-final-and" alignment computed by Moses. However, any other initial alignment may be used instead. Second, in the BIA tuning tools, a function commands the training, tuning and evaluation of an MT system from the output of the alignment decoder and returns an MT score to be optimised. Currently, only a function performing these steps with the Moses phrase-based SMT system is implemented, in which the symmetrised Giza++ alignment is substituted by the BIA alignment. However, the only task required to extend the toolkit to another MT system is to write another function performing the same steps for that MT system.

The toolkit has been only tested in linux, but should be portable to any system compatible with `cmake`. No multi-threading is implemented. However, a parameter for the number of threads available allows the user to divide tasks by forking or submitting jobs to a cluster (via the `qsub` command).

3.1. Training

The main training task is the estimation of the phrase association model and the source and target fertility models (the "improved features" described in Section 2.1). This task consists of counting the number of links and co-occurrences found in the initial alignment for each co-occurring phrase pair (to calculate relative probabilities), as well as the number of links for each source and target word. It is performed using `hash maps` with a custom hash function.

3.2. Decoding

Before aligning each sentence pair, models are loaded in memory (into `hash maps`). Then, for each sentence pair, a set of links to be considered in search is selected. This set is formed by the n best links for each source and for each target phrase (typically $n = 3$). For each link selected, relevant information (source and target positions, costs, etc.) is stored in a specific data structure. The set of considered links is then arranged in stacks corresponding to each source (or target) word.

Decoding consists in extending alignment hypotheses (that is, sets of links), also called states, by including each link of these link stacks. Note that we use an hypothesis stack for each number of source+target words covered. Decoding is based on a beam-search algorithm as follows:

```
insert initial state (empty alignment) in hypothesis stack
for each stack of links considered in search
* for each state in each hypothesis stack
    for each link in link stack
      - expand current state by adding this link
      - place new state in corresponding hypothesis stack
* perform histogram and threshold pruning of hypothesis stacks
```

Note that having one link stack for each source (or target) word ensures a fair comparison between hypotheses in which this word is covered. Furthermore, multiple hypothesis stacks ensure a fair comparison between hypotheses having the same number of covered words.

3.3. Tuning

At each iteration of the alignment weight tuning procedure (see Section 2.2), an SMT system is build, with which the development set is translated. The feature weights of this SMT system may be kept constant during alignment tuning, or they may be tuned with MERT at each iteration. In the latter case, we restrict the number of MERT runs to 12 iterations (not limited by default) and 10 restarts[2] (20 by default), to limit its maximum processing time. We also increase the minimum required change in weight variable from 0.00001 to 0.0001. Internal experiments on the tasks presented in Section 4 showed consistently that it is better to perform MERT at each iteration than to use a constant set of SMT feature weights during alignment tuning.

[2] According to internal experiments on two tasks, the average and standard deviation after 10 MERT runs is not affected by using 10 restarts instead of 20. In contrast, limiting the number of iterations to 12 may of course affect the results.

3.4. Issues

We had to face some issues during the implementation of the algorithm. First, the alignment result depends on the order of introduction of the links in the alignment hypotheses. Several solutions were envisaged: (i) a future cost; however, it should include the cost of crossing links, which we found no effective way to estimate. (ii) introduce the most confident or less ambiguous links first (iii) start from a non-empty initial alignment (for example, decode along the source side, then along the target side, and finally re-decode taking the intersection as initial alignment). In this configuration, we can expand a state by deleting or substituting a link. (iv) use multiple hypothesis stacks, which help decoding being more stable.

Second, the tuning process is not very stable (the optimisation algorithm can fall into a poor local maximum).

4. Experimental Evaluation

4.1. Data Sets

The experiments were conducted for the following tasks:

- the TC-STAR OpenLab[3] Spanish–English EPPS parallel corpus, which contains proceedings of the European Parliament. The BIA alignment model weights were tuned on two subsets extracted by randomly selecting 100,000 and 20,000 sentence pairs (these subsets will be referred to as 'ran100k' and 'ran20k' respectively). We built SMT systems from the optimum alignment obtained on each of these subsets. We also aligned the whole corpus (referred to as 'full') with the optimum weights obtained by tuning on the ran100k corpus, and built an SMT system from this alignment.
- the Chinese–English FBIS corpus, a collection (LDC2003E14) of texts in the news domain and released by the Linguistic Data Consortium (LDC[4]). We selected 100k sentence pairs as training data. Since the data was released in 2003, we used the test sets of NIST 2001 (nist01), NIST 2002 (nist02) and NIST 2003 (nist03) as development and test data.
- the Chinese–English data provided within the IWSLT 2007 evaluation campaign, extracted from the Basic Travel Expression Corpus (BTEC) (Takezawa et al., 2002). This speech corpus contains sentences similar to those that are usually found in phrase books for tourists going abroad. Training data consisted of the default training set, to which we added the sets devset1, devset2 and devset3.

The characteristics of the training, development and test sets used in each task are indicated in Tables 1, 2 and 3. More specifically, the statistics shown are the number

[3] http://www.tcstar.org/openlab2006

[4] http://www.ldc.upenn.edu

Set	Language	Sentences	Words	Vocabulary	Lmean	Ref.
Train	Spanish	1.27 M	36.2 M	152 k	28.4	1
(full)	English	1.27 M	34.6 M	106 k	27.2	1
Train	Spanish	100 k	2.8 M	55 k	28.4	1
(ran100k)	English	100 k	2.7 M	38 k	27.2	1
Train	Spanish	20 k	0.57 M	27 k	28.6	1
(ran20k)	English	20 k	0.55 M	20 k	27.3	1
MERT Dev.	Spanish	892	28.6 k	4.8 k	32.0	2
1st ref.	English	892	28.9 k	3.9 k	32.4	
Dev.	Spanish	1008	25.8 k	3.9 k	25.6	2
1st ref.	English	1008	26.3 k	3.1 k	26.1	
Test	Spanish	840	22.7 k	4.1 k	27.1	2
1st ref.	English	840	22.8 k	3.3 k	27.1	

Table 1. Statistics for the training, development and test data sets for EPPS data (M and k stand for millions and thousands, respectively, Lmean refers to the average sentence length in number of words, and Ref. to the number of available translation references).

of sentences, the number of words, the vocabulary size (or number of distinct words), the average sentence length in number of words and the number of available translation references. As mentioned previously, Dev refers to the development set used to calculate the BLEU score at each alignment optimisation iteration (with the SPSA algorithm), MERT Dev refers to the development corpus used within the internal SMT MERT procedure at each SPSA iteration, and Test refers to the test set used to realise an extrinsic evaluation of the optimal alignment system.

4.2. Results

In this section we compare the BLEU score obtained by SMT systems built from alignments computed with the BIA toolkit and with a number of state-of-the-art alignment systems. The second-pass BIA models were trained on a high-quality alignment computed by combining the IBM Model 4 source–target (s2t) and target–source (t2s) alignments with the "grow-diag-final-and" heuristic (Koehn et al., 2003). We also computed the source–target and target–source alignments of IBM Model 4 (Brown et al., 1993) as implemented by Giza++, and 4 different combinations of these alignments (intersection (I), union (U), grow-diag-final (GDF) and grow-diag-final-and (GDFA) heuristics (Koehn et al., 2003)). In addition, we used an HMM-based joint training model with posterior decoding (Liang et al., 2006) and an HMM-based model which explicitly takes into account the target language constituent structure (DeNero

Set	Language	Sentences	Words	Vocabulary	Lmean	Ref.
Train	Chinese	100 k	3.1 M	30.4 k	30.7	1
	English	100 k	3.7 M	56.2 k	37.3	1
MERT Dev.	Chinese	935	27.9 k	4.6 k	29.9	3
1st ref.	English	935	28.9 k	4.9 k	30.9	
Dev.	Chinese	993	26.7 k	4.7 k	26.9	8
1st ref.	English	993	29.1 k	4.9 k	29.3	
Test	Chinese	878	25.4 k	4.3 k	28.9	5
1st ref.	English	878	28.2 k	4.8 k	32.1	

Table 2. Basic statistics for the training, development and test data sets for FBIS data.

Set	Language	Sentences	Words	Vocabulary	Lmean	Ref.
Train	Chinese	41.5 k	362 k	11.4 k	8.7	1
	English	41.5 k	389 k	9.7 k	9.4	1
MERT Dev.	Chinese	500	6.1 k	1.3 k	12.1	7
1st ref.	English	500	7.3 k	1.2 k	14.7	
Dev.	Chinese	489	5.7 k	1.1 k	11.7	7
1st ref.	English	489	6.4 k	1.0 k	13.4	
Test	Chinese	489	3.2 k	0.9 k	6.5	6
1st ref.	English	489	3.7 k	0.8 k	7.6	

Table 3. Basic statistics for the training, development and test data sets for BTEC data.

and Klein, 2007), both implemented in the Berkeley word alignment package,[5] and referred to as "bk" and "syn-bk", respectively. Finally, we computed alignments with the Posterior Constrained Alignment Toolkit (PostCAT[6] (Graça et al., 2010)).

Table 4 shows, for each of the five considered tasks, the BLEU score of the SMT systems built from four types of alignments: (i) BIA alignment, (ii) the best alignment(s) (named in parenthesis) among the nine other alignment systems, (iii) the Moses default alignment (GDF), and (iv) the initial alignment used to train BIA models (GDFA). These BLEU scores are an average obtained over four MERT runs with different random seeds. We can make a number of interesting observations from these results. First, in all cases, the score achieved via BIA alignment was at least as good as the score achieved via the initial alignment used to train BIA models. Second, with respect to the Moses default alignment scheme, using BIA yielded a lost of 0.1 BLEU point in one task, and gains of 0.5, 0.4, 1.3 and 1.2 BLEU points in the other tasks. Fi-

[5] http://nlp.cs.berkeley.edu/pages/WordAligner.html

[6] http://www.seas.upenn.edu/~strctlrn/CAT/CAT.html

	EPPS				
Alignment	Full	Ran100k	Ran20k	FBIS	BTEC
BIA	56.2	**51.7**	**46.6**	**23.0**	**35.2**
Best other	**56.7** (U)	51.4 (t2s)	46.2 (U,GDF,GDFA)	**23.0** (GDFA)	34.8 (bk,syn-bk)
Moses default (GDF)	56.3	51.2	46.2	21.7	34.0
Initial (GDFA)	56.2	51.1	46.2	23.0	33.9

Table 4. Extrinsic evaluation (in terms of BLEU score) of the BIA alignment system, compared to the other alignment systems considered.

nally, BIA always yielded the best alignment (in terms of BLEU score) of the set of ten alignment systems when its model parameters had been tuned on the whole corpus. This was the case for the EPPS ran100k and ran20k tasks, and for the FBIS and BTEC tasks. For the EPPS “full” task, the parameters had been tuned on the ran100k task, and the whole corpus had then been aligned with the optimal parameters found.

This last result is problematic for the alignment of large corpora given a limitation of the current version: a full SMT system must be built at each iteration of the alignment parameter optimisation, which would be very costly on a large corpus. Thus the tuning cannot be performed on the whole training corpus, unless it is reasonably small.

5. Usage Instructions

Training, tuning and decoding instructions are available on the BIA aligner project website,[7] from where the source code can also be freely downloaded. The sample shell script also gives usage examples. Several wrapper scripts were implemented to make training, tuning and decoding easier. To train the alignment models, use:

```
training/train-models.pl
```

To tune the alignment feature weights, use:

```
tuning/tune-model-weights.pl
```

and finally to run the alignment decoder, use the following binary:

```
bia
```

6. Conclusions and Further Work

We presented the BIA toolkit, a suite consisting of a discriminative phrase-based word alignment decoder based on linear alignment models, along with training and

[7] http://code.google.com/p/bia-aligner/

tuning tools. The tuning of the model weights may be performed directly according to MT metrics.

We reported results of experiments conducted on the Spanish–English Europarl task (with three corpus sizes), on the Chinese–English FBIS task, and on the Chinese–English BTEC task. The BLEU score obtained with BIA alignment was always as good or better than the one obtained with the initial alignment used to train BIA models. In addition, BIA always yielded the best alignment (in terms of BLEU score) of a set of ten alignment systems when its model parameters had been tuned on the whole corpus. In one task, the corpus was too large to perform the tuning on all the data and thus tuning was performed on a subset of it (less than 10% of its size).

In the future we want to develop a new tuning procedure whose required computing time would be independent from the size of the training corpus.

Acknowledgements

This research is supported by the European Union under the EuroMatrix Plus project (http://www.euromatrixplus.net, IST-2007.2.2-FP7-231720) and by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University.

Bibliography

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

DeNero, John and Dan Klein. Tailoring word alignments to syntactic machine translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Prague, Czech Republic, June 2007.

Graça, João V., Kuzman Ganchev, and Ben Taskar. Learning tractable word alignment models with complex constraints. *Computational Linguistics*, 36(3):481–504, 2010.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proc. of the Human Language Technology Conference of the NAACL*, pages 48–54, Edmonton, Canada, 2003.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (Demo and Poster Sessions)*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P07/P07-2045`.

Lambert, Patrik and Rafael E. Banchs. Word association models and search strategies for discriminative word alignment. In *Proc. of the Conference of the European Association for Machine Translation*, pages 97–103, Hamburg, Germany, 2008.

Lambert, Patrik, Rafael E. Banchs, and Josep M. Crego. Discriminative alignment training without annotated data for machine translation. In *Proc. of the Human Language Technology Conference of the NAACL (Short Papers)*, pages 85–88, Rochester, NY, USA, 2007.

Liang, Percy, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proc. of the Human Language Technology Conference of the NAACL*, pages 104–111, New York City, USA, June 2006.

Liu, Yang, Qun Liu, and Shouxun Lin. Log-linear models for word alignment. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 459–466, Ann Arbor, Michigan, June 2005.

Liu, Yang, Qun Liu, and Shouxun Lin. Discriminative word alignment by linear modeling. *Computational Linguistics*, 36(3):303–339, 2010.

Melamed, I. Dan. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, 2000.

Moore, Robert C. A discriminative framework for bilingual word alignment. In *Proc. of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Vancouver, Canada, October 2005.

Och, Franz J. Minimum error rate training in statistical machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics*, pages 160–167, 2003.

Spall, James C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Automat. Control*, 37:332–341, 1992.

Takezawa, Toshiyuki, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proc. of Third International Conference on Language Resources and Evaluation 2002*, pages 147–152, Las Palmas, Canary Islands, Spain, 2002.

Vilar, David, Maja Popovic, and Hermann Ney. AER: Do we need to ”improve” our alignments? In *Proc. of the Int. Workshop on Spoken Language Translation, IWSLT'06*, pages 205–212, Kyoto, Japan, 2006.

Address for correspondence:
Patrik Lambert
patrik.lambert@lium.univ-lemans.fr
LIUM, University of Le Mans
Avenue Laënnec, 72085 Le Mans Cedex 9, France