Yet Another Fast, Robust and Open Source Sentence Aligner. Time to Reconsider Sentence Alignment?

Fethi Lamraoui
RALI / DIRO
Université de Montréal
CP. 6128 Suc. Centre-ville
H3C 3J7 Montréal, Québec, Canada
lamraouf@iro.umontreal.ca

Philippe Langlais
RALI / DIRO
Université de Montréal
CP. 6128 Suc. Centre-ville
H3C 3J7 Montréal, Québec, Canada
felipe@iro.umontreal.ca

Abstract

Although good sentence aligners are freely available, our laboratory regularly receives requests from researchers and industries for aligning parallel data. This motivated us to release yet another open-source sentence aligner we wrote nearly 20 years ago. This aligner is simple but it performs surprisingly well and often better than more elaborated ones, and do so very fast, allowing to align very large corpora. We analyze the robustness of our aligner across different text genres and level of noise. We also revisit the alignment procedure with which the Europarl corpus has been prepared and show that better SMT performance can be obtained by simply using our aligner.

1 Introduction

Sentence alignment has received a lot of attention in the late eighties and early nineties. Based on the excellent results reported (Kay and Röscheisen, 1993; Brown et al., 1993; Gale and Church, 1993; Chen, 1993) the community rapidly considered the sentence alignment problem solved, and the interest for it nowadays remains marginal. This matter of fact is further reinforced by the availability of good ready-to-use open source aligners (Gale and Church, 1993; Varga et al., 2005; Moore, 2002; Li et al., 2010).

While many sentence alignment systems have been proposed, they basically make use of two types of features: length-based (Brown et al., 1991; Gale and Church, 1993) and lexical-based features (Kay and Röscheisen, 1993; Chen, 1993; Melamed, 1997; Wu, 1994; Moore, 2002). Lexical features have been repeatedly reported to improve upon length-based features when noisy corpora are aligned (Chen, 1993; Wu, 1994).

Lexical-based systems differ in the way lexical features are acquired. Some systems require an initial bilingual dictionary (Li et al., 2010; Wu, 1994), while others induce such lexicons online (Kay and Röscheisen, 1993; Chen, 1993; Moore, 2002), often training IBM models (Brown et al., 1993) on the most promising sentence pairs identified by a length-based method. To alleviate the need of a lexicon, Simard et al. (1992) proposed to use cognates, a lightweight lexical feature that is suited for aligning Indo-European language pairs.

Lexical-based systems that refine their lexicon online are typically slower, and somehow dependent on the language pair under consideration, and the quantity of material to align (word-alignment for some language pairs and with small corpora is challenging). On the contrary, length-based systems are mainly language independent and efficient to compute, explaining in great part the popularity of tools such as the G&C aligner (Gale and Church, 1993).

For computation reasons, most aligners we know of (including the one we present) rely at some point on the so-called monotonic hypothesis. This constraint can be very strong as in (Gale and Church, 1993). Also, multi-stage alignment systems often seek for a monotonic alignment first, that seeds subsequent alignment stages, possibly allowing local reordering, as in (Deng et al., 2007).

In this paper we posit that sentence alignment deserves more investigation than it has received recently. We illustrate this by studying a very simple aligner, namely YASA, that we compare

favorably in a number of settings — including noisy parallel texts — to two popular (and much better engineered) open-source aligners, namely BMA (Moore, 2002) and HUNALIGN (Varga et al., 2005). We also show that using our aligner within the procedure with which the EUROPARL bitext, as distributed on `http://www.statmt.org/europarl/`, leads to gains in statistical machine translation (SMT) according to the BLEU metric (Papineni et al., 2002).

In the remainder of this paper, we describe our system in Section 2. We report on the experiments we conducted in Section 3. We discuss related work in Section 4 and conclude in Section 5.

2 Description of the System

YASA operates a two-step process through the parallel data. Cognates are first recognized in order to accomplish a first token-level alignment that (efficiently) delimits a fruitful search space (see Section 2.1). Then, sentence alignment is performed on this reduced search space (see Section 2.2). For efficiency reasons, both stages are accomplished by dynamic programming. As a consequence, YASA, as most aligners we know of, assumes a monotonic alignment.

YASA allows for some parametrization. For instance, the first stage is optional, and similar to (Varga et al., 2005), a bilingual lexicon can be provided which entries are treated as cognates. Also, the function optimized in the second stage is controlled by several meta-parameters that can be changed. Most of those controls are typically not used when we routinely align parallel data in our laboratory. Therefore, we used the system in its default setting in this study. Also, we did not provide any bilingual lexicon to our system.

2.1 Cognate-based Search-Space Reduction

Our space-reduction method is based on the intuition that aligning sentences can be done efficiently using a coarse word-level alignment. This idea has received many incarnations (see Section 4) that are complex to adjust (many heuristics, thresholds or external resources). Our approach is simple, is time-efficient and good at reducing the search space (see Section 3.2).

As suggested by Church (1993), a bilingual corpus can be represented as a dotplot (Gibbs and McIntyre, 1970), that is, a 2-dimensional binary matrix M, where the i_{th} line stands for the i_{th} word of the source text and the column j stands for the j_{th} word of the target text. In our case, a cell $M_{i,j}$ is set to 1 if the i_{th} source word and the j_{th} target word are cognates (or constitute one entry in a specified bilingual lexicon). An example of a dotplot for an excerpt of the novel of Jules Verne *"De la terre à la lune"* is provided in Figure 1. We observe that among the many cognate relations,[1] an alignment emerges, that we track by a simple dynamic programming technique which encourages to stick to the main diagonal, while rewarding cognate correspondences.

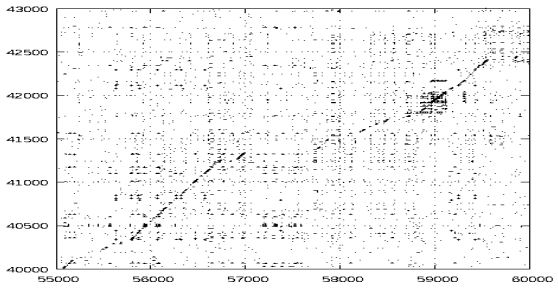

Figure 1: Dotplot of an excerpt of the VERNE bitext of the BAF corpus. Abscissas and ordinates indicate French and English word positions respectively. Each point indicates a cognate relation.

For a bitext of I source words and J target ones that enter into a cognate relation with words in the other language, we seek the alignment which minimizes $S(I, J)$, where $S(i, j)$ is defined as:

$$\begin{cases} 0 & \text{if i = j = 1} \\ \min\limits_{k=i-R}^{i-1} \min\limits_{l|M_{k,l}=1} \begin{pmatrix} S(k,l) + \\ c(k,l,i,j) \end{pmatrix} & \text{otherwise} \end{cases}$$

with a cost function:

$$c(k,l,i,j) = \left| \frac{l-j}{k-i} - a \right| + (k-i-1) \times \mathcal{C}$$

where R is a constant that sets the maximum tolerance for discontinuities in the cognate alignment, $\mathcal{C}$ is a constant that penalizes a discontinuity,[2] and a is the slope of the main diagonal. This cost-function is inherited from the first implementation of our aligner (Langlais, 1997), and definitely has to be revisited, despite its good results.

Once the cognate-based alignment is performed, we simply delimit a sentence beam-search as a

[1] In practice we only mark low-frequency words.

[2] We used the values $R = 50$ and $\mathcal{C} = 5$.

fixed-size number of sentences centered around the sentence pairs involved in alignment found at the cognate level.[3] The search space delimited from the alignment matrix exemplified in Figure 1 is illustrated in Figure 2. We observe that a rather large sequence of source sentences (near sentence 2000) has not been translated in the target language. The search space will force the second stage to discover n-0 alignments that a standard length-based model would simply miss.

Note that when few cognates are found, our alignment procedure backups to setting a fixed-size search space around the main diagonal, which is likely appropriate in most cases.

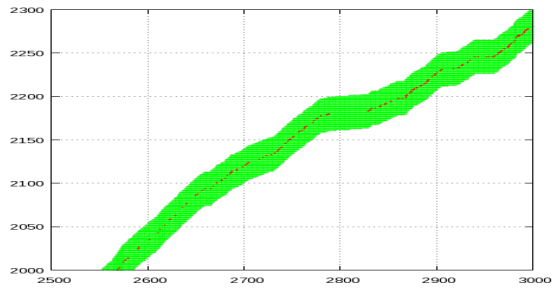

Figure 2: Word-alignment (in red) identified for the cognate-based matrix of Figure 1, and the subsequently delimited sentence-based search space.

2.2 Sentence Alignment

The second stage is very similar to the approach of (Gale and Church, 1993), except that a cognate-based component similar to (Simard et al., 1992) is added to the function minimized while aligning. The score of a match or bisegment is:

$$S_{yasa} = -\log\left[\left(\frac{P_T(c|n)}{P_R(c|n)}\right)^{\lambda_1} \times P(match|\delta)^{\lambda_2}\right]$$

where δ is a quantity directly computed from the length (counted in characters) of the source and target sentences candidate for pairing, and is assumed to follow a normal distribution. The first term is a likelihood ratio that estimates how likely a pairing involving n words on average share c cognates under the assumption that the sentences are in translation (numerator) or not (denominator). Both distributions are modeled with binomials which probability of success p_T and p_R have been setup empirically.[4] Once developed, and introducing a new coefficient for the a priori distribution of bisegments, the score becomes:

$$\begin{aligned} S_{yasa} = \quad & -\lambda_1 \left(\left[c \log \tfrac{p_T}{p_R}\right] - \left[(n-c) \log \tfrac{1-p_T}{1-p_R}\right]\right) \\ & -\lambda_2 \log\ P(\delta|match) \\ & -\lambda_3 \log\ P(match) \end{aligned} \quad (1)$$

We used the non-derivative minimization Simplex technique (Nelder and Mead, 1965) to find the λ-values on a development corpus.[5]

Both cognates (dynamically detected) and the entries of the bilingual lexicon (if provided) are counted for c in equation 1. We used the same definition of a cognate proposed by Simard et al. (1992): two tokens that either do not contain digits and share a prefix of 4 characters (diacritics are being removed before the comparison), or alphanumerical tokens that are identical. Obviously, this definition brings false alarms (*e.g.*, library/librairie (lit. bookshop)), and fails in a number of cases (*e.g.*, night/nuit). It is undoubtedly interesting to use a more sensible definition, such as the one of (Kondrak, 2001).

This scoring function is inherited from the practices in use at the time we developed this system (nearly 2 decades ago) and must be revisited. A natural way to proceed would be to implement each score (and possibly others) as features of a discriminative model that would be adjusted either on a held out corpus (similarly to (Munteanu and Marcu, 2005)) or online after a first accurate pass of sentence alignment, as in (Yu et al., 2012).

3 Experiments

We conducted experiments on two known testbeds, BAF, a small-scaled multi-genre English-French parallel corpus, and EUROPARL, a large-scaled multilingual corpus.

We follow (Langlais et al., 1998) and evaluate the quality of a candidate alignment by precision and recall ratios — that we summarize into F-measure — at different levels of granularity. At the alignment level, a candidate receives a credit for each correctly identified bisegment. Precision and recall are computed accordingly. One problem with this, is that an error at the alignment level

[3] A beam-size of 20 sentences is set as a default.

[4] The default values in YASA are those suggested by (Simard et al., 1992), that is, $p_T = 0.3$ and $p_R = 0.09$.

[5] Legislative data was used, and this yielded the values $\lambda_1 = 0.5$, $\lambda_2 = 0.2$, $\lambda_3 = 1$.

(*e.g.*, 2-1 instead of 1-1) that is partially correct might be of interest in a given application, but not credited. The authors proposed to compute precision and recall at smaller levels of granularity (sentence, word and character). At the sentence level, a n-m bisegment is represented by the cartesian product of its source and target sentences, that is, by $n \times m$ sentence pairs. A candidate bisegment receives a credit proportional to the number of correct sentence pairs it contains. We note F_S and F_A the F-measure computed at the sentence and alignment level respectively.

3.1 BAF

BAF (Simard, 1998) gathers 11 reference bitexts of 4 genres: institutional texts (INST, 4 bitexts), scientific articles (SCIENCE, 5 bitexts), one technical report (TECH) and a novel of Jules Vernes (VERNE). This corpus[6] totalizes 25 049 French sentences and 23 898 English one. It has notably been used in the ARCADE evaluation exercise on sentence alignment (Langlais et al., 1998). Some of those bitexts proved to be more difficult to align, in particular, the VERNE corpus which English version has been drastically abridged compared to the (original) French version.

3.1.1 Performance on BAF

We applied YASA, BMA and HUNALIGN to the different bitexts of the BAF corpus. We also ran the G&C aligner, but its performance was too low because of the lack of anchors required by the approach. The results are reported in Table 1.

	YASA		BMA		HUNALIGN	
bitext	F_A	F_S	F_A	F_S	F_A	F_S
INST	**94.9**	**96.4**	93.3	91.4	91.0	93.4
SCIENCE	**89.4**	**91.7**	88.9	86.4	84.8	86.5
VERNE	69.2	**86.9**	**72.3**	74.6	66.0	74.6
TECH	90.4	**10.9**	**94.2**	10.3	89.6	10.7

Table 1: Alignment and sentence-based F-measures macro-averaged over text genres.

At the alignment level, BMA and YASA behave comparably, slightly outperforming HUNALIGN. BMA is at an advantage on more difficult bitexts (VERNE and TECH). On the other hand, at the sentence-level, YASA outperforms the other two aligners by a large margin for all genres. It must be noted that BMA delivers typically a higher precision than YASA and HUNALIGN, but a lower recall. It is also noteworthy that at the sentence level, the systems did poorly on the TECH corpus. This is because both versions ends with an alphabetically sorted index that is not sentence aligned in the reference, but instead encoded as a 250-250 bisegment. The cartesian product thus contains 62.5k sentence pairs, which drastically impacts recall.

3.1.2 Adding Noise to BAF

In order to measure the robustness of YASA to noisy parallel data, we artificially introduced some noise in the bitexts of the BAF corpus. This was done by randomly removing in the source text 1 to 6 sentence-pairs every 10 ones, thus yielding noise ranging from 10% to 60%.[7] A comparison of F_A obtained by YASA and BMA is provided in Table 2. We concentrated on BMA here because it overall outperformed HUNALIGN in the previous experiment. Also, Yu et al. (2012) observed that BMA is competitive with other aligners on the BAF corpus.

%	INST	SCIENCE	VERNE	TECH
0	94.9 (+1.6)	89.4 (+0.5)	69.2 (-3.1)	90.4 (-3.8)
10	85.9 (+1.1)	78.8 (-1.9)	56.1 (-2.5)	72.2 (-7.1)
20	81.7 (+2.2)	75.2 (+2.9)	54.5 (-3.8)	74.7 (-6.5)
30	77.8 (+7.0)	70.5 (+8.3)	36.7 (-7.6)	69.1 (-3.8)
40	76.0 (+16.6)	69.1 (+30.5)	40.6 (+3.3)	65.0 (+1.8)
50	69.0 (+23.0)	67.2 (+41.8)	44.7 (+28.8)	64.3 (+14.9)
60	69.5 (+44.8)	67.8 (†)	49.9 (+41.9)	65.8 (†)

Table 2: F_A of YASA as a function of noise. Figures in parenthesis are absolute gains of YASA over BMA. (A negative value indicates that BMA outperforms YASA). BMA occasionally crashed, this is marked with a † symbol.

We observe that both YASA and BMA degrade with increasing noise. On the easiest sub-corpus (INST) F_A decreases from 94.9 down to 69.5 for YASA. The loss of BMA is even more drastic (from 93.3 down to 24.7). Similar tendencies are observed for other genres with slight differences though. In particular, for the VERNE sub-corpus, BMA outperforms YASA at low levels of noise, but becomes increasingly worse with the noise ratio.

[6] http://rali.iro.umontreal.ca/rali/?q=en/node/71.

[7] We also implemented the noise procedure described in (Goutte et al., 2012) but found it produces easier-to-align data.

Note that at the sentence level (F_S), YASA always outperforms BMA, and the difference in performance between the two approaches increases with noise (e.g. F_S of 59.2 for YASA, and 15.5 for BMA at 60% noise on VERNE).

3.2 Speed Issue

One noticeable difference between YASA and BMA is the time needed to deliver the alignment. On the BAF corpus, we observed that YASA needs 27s on average to align one of the BAF bitexts, while BMA requires 257s. The difference in speed increases with the length of the parallel texts to align (as well as the level of noise in the data). In order to illustrate this, we ran BMA and YASA on the French-English EUROPARL bitext (see next section) varying its size from 1000 up to 1 million sentence pairs. The speed of both aligners[8] is reported in Figure 3. While for bitexts of up to 100k sentence pairs, YASA is one order of magnitude faster than BMA, the difference in speed increases dramatically with larger datasets. In particular, for the largest bitext we considered, YASA delivered the alignment in less than 24 minutes, while BMA required more than 30 hours. This speed difference is not surprising since BMA is training an IBM model 1 online. Still, this puts YASA at an advantage from a practical point of view. Also, the memory footprint of YASA is lower.[9]

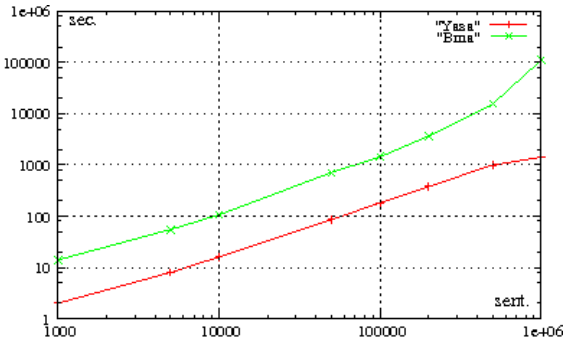

Figure 3: Speed (in seconds) of YASA and BMA as a function of the number of sentence pairs aligned. Note the logarithmic scale on both axes.

3.3 EUROPARL

The EUROPARL corpus (Koehn, 2005) is a large parallel corpus often used for training SMT engines. It comes already organized into a sentence-aligned bitext,[10] thanks to an alignment toolkit that is making use of the markup available in daily files (see Figure 4 for an excerpt) in order to guide a G&C-like aligner.

```
<CHAPTER ID=1>
<SPEAKER ID=1 NAME="President">
Resumption of the session
<P>
I declare resumed the session of the European
Parliament adjourned on Friday 17 December
1999, and I would like once again...
<P>
```

Figure 4: Excerpt of a daily EUROPARL file.

A sketch of the procedure implemented in the script `sentence-align-corpus.perl`, which is referred to as PK in the sequel, goes as follows: chapters (tag CHAPTER) are assumed to be aligned, that is, the i_{th} source chapter will be sentence aligned with the i_{th} target one, regardless of the chapter identifier (ID). Speaker turns (tag SPEAKER) are further assumed synchronized: the i_{th} source speaker turn will be aligned to the i_{th} target one. In case there are more speaker turns in one language, the remaining ones are simply ignored. Finally, the source and target speaker turns are aligned only if their number of paragraphs (tag P) is identical.

While those heuristics certainly contributes to increase the alignment precision, we observed on the French-English EUROPARL bitext that the "number of paragraphs heuristic" concerns 6 484 speaker turns, for a total of 64 441 French and 63 097 English paragraphs. This observation triggered a number of alternative alignment procedures we investigated:

YASA where we replace G&C by YASA in the alignment procedure we just described.

PK++ where we remove the "number of paragraphs" heuristic. Therefore, speaker turns are aligned by the G&C aligner, whatever their number of paragraphs.

YASA++ is similar to PK++ with the exception that YASA is used instead of G&C.

[8] Both aligners were run on a similar computer.

[9] In our experiments, aligning 100k sentences per language requires 700Mb of memory for YASA and 2.3Gb for BMA.

[10] `http://www.statmt.org/europarl/`

DAILY where we do not use explicitly the markup available. The markup is kept, but YASA considers it as ordinary sentences. In the end, bisegments associating sentences to markup are removed.

We trained 5 SMT systems, one on each bitext we obtained thanks to the strategies we described. We tuned all the systems on the same news commentary file NEWS08.[11] The performance of the French-English translation engines we deployed are reported in Table 3. YASA++ and DAILY (which is not making use of corpus specific information) are the two best configurations. Both approaches allow to align all the material (speaker turns that do not have the same number of paragraphs) which seems to be a good strategy. It is noteworthy that using YASA instead of G&C brings as well systematic gains (PK versus YASA, and PK++ versus YASA++). We do not have an explanation for the reason why the systems tuned on news texts did better at translating an excerpt of 2 500 sentence pairs we extracted from Canadian Hansards (HANS).

	NEWS09	NEWS10	NEWS11	HANS
DAILY	**20.02**	20.46	**21.16**	23.64
PK	19.43	20.14	20.77	23.46
YASA	19.81	20.47	20.93	23.60
PK++	19.49	20.27	20.62	23.25
YASA++	19.96	**20.61**	21.05	**23.95**

Table 3: French-English translation results.

We ran a similar experiment with the German-English and Finnish-English language pairs.[12] Results are reported in Table 4. As expected, BLEU scores are much lower than for the French-English translation task. While on the German-English translation task, the difference in BLEU between the different configurations is not significant, we observe that overall, YASA and YASA++ lead to better results.

To analyze more precisely those results, we compared the vocabulary of the IBM model trained by MOSES in the YASA++ and PK configurations,

[11] http://www.statmt.org/wmt12/

[12] For the Finnish-English pair, since news commentary data is not available, we tuned and tested the system on the JRC-Acquis corpus (http://ipsc.jrc.ec.europa.eu/index.php?id=198), with the same number of sentence pairs as used for the other translation directions.

	de-en			fi-en
	NEWS09	NEWS10	NEWS11	ACQUIS
DAILY	15.54	16.12	15.26	9.44
PK	16.44	17.10	16.30	9.56
YASA	16.39	**17.13**	**16.31**	10.42
PK++	16.35	17.08	16.24	10.08
YASA++	**16.45**	17.09	16.30	**10.48**

Table 4: German-English and Finnish-English translation results.

and retained the list of words seen no more than 3 times in PK, but seen at least once more in YASA. The words of this list were searched for in all the sentences of our test material, from which we built 3 subsets: one in which test sentences contain (at least) one word unknown of PK but known of YASA ($f = 0$), one in which they contain (at least) between one and three occurrences in PK and at least one more occurrence in YASA ($f \in [1, 3]$), and the concatenation of both.

The translation results are reported in Table 5. Gains in BLEU are consistently observed for YASA++, although they still are small for the German-English translation direction. Also, gains are more marked for sentences containing words not seen in PK, especially for German-English. For some reasons, those sentences appear to be easier to translate in French-English (BLEU scores are higher than the ones reported in Table 3).

	$f = 0$		$f \in [1, 3]$		$f \in [0, 3]$	
	Y++	PK	Y++	PK	Y++	PK
fr-en	**30.4**	29.5	**21.5**	20.6	**23.7**	22.7
	(118 sent.)		(394 sent.)		(512 sent.)	
de-en	**14.2**	13.8	**16.3**	16.1	**15.9**	15.7
	(100 sent.)		(344 sent.)		(444 sent.)	
fi-en	**7.5**	7.0	**14.6**	14.1	**14.3**	13.9
	(38 sent.)		(227 sent.)		(256 sent.)	

Table 5: Comparison of BLEU scores obtained by YASA++ and PK on sentences containing unfrequent words.

4 Related Work

Different sentence alignmers have been proposed, and many share with YASA a two-stage approach that first determines a plausible search space, then applies a more informed alignment procedure.

Simard and Plamondon (1998) for instance proposed a method for recursively identifying anchor points (basically, isolated cognate pairs) that serve to reduce the search space. Li et al. (2010) developed a bitext splitter in order to reduce the search space, based on an initial length-based alignment, and therefore is prone to errors committed during this step. Those approaches end up making hard decisions, which we found problematic.

Melamed (1997) proposed an "expanding rectangle search strategy" that shares properties with the dynamic programming approach we proposed, while requiring more meta-parameters and heuristics to work. Chen (1993) controls the search space by making use of thresholds, and taking care of large deletions in one language by some heuristics. Our word-level dynamic programming has been shown efficient at identifying large deletions and is conceptually simpler to implement. See also (Chang and Chen, 1997) for an image processing inspired approach.

A few approaches are relaxing the non monotonicity YASA and many other aligners are assuming. The generative model of (Deng et al., 2007) includes a divide clustering phase that accounts for local non monotonicity. Perhaps the most flexible approach we know of is the one of (Bisson and Fluhr, 2000) that proposes to tackle sentence alignment as a cross-lingual information retrieval task. The problem with the approach is that it does not enforce the cohesion a sentence-alignment typically exhibits, thus delivering poor performance when the alignment is indeed monotonic.

Recent alignment methods have been proposed that mainly rely on a length-based alignment step in order to identify reliable sentence pairs that are used either for training a classifier to recognize parallel sentence pairs (Yu et al., 2012), or for training a translation engine used to translate the source text so that the alignment takes place monolingually (Sennrich and Volk, 2011) or for training an IBM model 1 used for grouping together many-to-one and one-to-many clusters of sentences (Braune and Fraser, 2010). Those approaches are time consuming, due to training time.

5 Discussion

We have described a very simple sentence aligner that to our surprise outperforms better engineered ones on different genres, at least for the pair of languages we studied. We stressed our aligner by adding up to 60% of noise in the data to align, and observed that its performance decreases smoothly, at a much lower pace than BMA. We also showed positive impact of our aligner on SMT. Last, YASA is fast, which allows to align very large corpora. The (C++) code of this aligner is free of use and is available at: `http://rali.iro.umontreal.ca/rali/?q=en/yasa`.

Outperforming available aligners was not what drove this work. Indeed, we were surprised of this outcome. The fact that sentence alignment impacts SMT on a well-studied setting was also unexpected, and somehow contrasts with (Goutte et al., 2012) which concludes that SMT can deal with up to 30% sentence alignment error rate. At the very least, we believe this shows that sentence alignment deserves more investigation than it has received recently. We foresee many situations where better sentence alignment should help improving front-end applications. Notably, many parallel texts are nowadays acquired over the web (Uszkoreit et al., 2010) and likely contain noise that challenges sentence aligners.

YASA is that simple, that it likely performs well on a number of language pairs. Still, it might be inaccurate at aligning texts in languages with different scripts, since it relies on cognates for reducing its search space. Transliteration, a technology that has received much attention recently (Zhang et al., 2012) should come at help here. Also, our aligner assumes a monotonic alignment between documents, possibly involving short or long untranslated passages, which might limit its use.

Ultimately, we share the point of view of (Deng et al., 2007) that sentence alignment should be modeled as part of the targeted application (e.g. machine translation), and not as an independent component, so that it can benefit the optimization conducted toward the task.

Acknowledgments

This work has been partially funded by the Natural Sciences and Engineering Research Council of Canada.

References

Bisson, Frédérique and Christian Fluhr. 2000. Sentence alignment in bilingual corpora based on

crosslingual querying. In *6th RIAO*, pages 529–542.

Braune, Fabienne and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *23rd COLING*, pages 81–89.

Brown, Peter F., Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *29th ACL*, pages 169–176.

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

Chang, J.S. and M.H. Chen. 1997. An alignment method for noisy parallel corpora based on imag processing techniques. In *35th ACL*, pages 297–304.

Chen, Stanley F. 1993. Aligning sentences in bilingual corpora using lexical information. In *31st ACL*, pages 9–16.

Church, Kenneth Ward. 1993. Char_align: A program for aligning parallel texts at the character level. In *31st ACL*, pages 1–8.

Deng, Yonggang, Shankar Kumar, and William Byrne. 2007. Segmentation and alignment of parallel text for statistical machine translation. *Natural Language Engineering*, 13:235–260, 8.

Gale, William A. and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Comp. Ling.*, 19(1):75–102, March.

Gibbs, Adrian J. and George A. McIntyre. 1970. The diagram, a method for comparing sequences. its use with amino acid and nucleotide sequences. *Eur. J. Biochem*, 16:1–11.

Goutte, Cyril, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *10th AMTA*.

Kay, Martin and Martin Röscheisen. 1993. Text-translation alignment. *Comp. Ling.*, 19(1):121–142.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *10th Machine Translation Summit*, pages 79–86.

Kondrak, Grzegorz. 2001. Identifying cognates by phonetic and semantic similarity. In *2nd NAACL*, pages 1–8.

Langlais, Philippe, Michel Simard, and Jean Véronis. 1998. Methods and practical issues in evaluating alignment techniques. In *36th ACL and 17th COLING*, pages 711–717.

Langlais, Philippe. 1997. A System to Align Complex Bilingual Corpora. Technical report, CTT, KTH, Stockholm, Sweden. TMH-QPSR 4/1997.

Li, Peng, Maosong Sun, and Ping Xue. 2010. Fast-champollion: a fast and robust sentence alignment algorithm. In *23rd COLING: Posters*, pages 710–718.

Melamed, I. Dan. 1997. A portable algorithm for mapping bitext correspondence. In *35th ACL and 8th EACL*, pages 305–312.

Moore, Robert C. 2002. Fast and accurate sentence alignment of bilingual corpora. In *5th AMTA*, pages 135–144.

Munteanu, Dragos Stefan and Daniel Marcu. 2005. Improving Machine Translation Performance by Exploiting non-parallel Corpora. *Comp. Ling.*, 31:477–504.

Nelder, J. A. and R. Mead. 1965. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th ACL*, pages 311–318.

Sennrich, R. and M. Volk. 2011. Iterative, mt-based sentence alignment of parallel texts. In *18th Nordic Conference of Computational Linguistics*.

Simard, Michel and Pierre Plamondon. 1998. Bilingual sentence alignment: Balancing robustness and accuracy. *Machine Translation*, 13(1):59–80.

Simard, Michel, George Foster, and Pierre Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *4th TMI*, pages 67–81.

Simard, Michel. 1998. The BAF: A Corpus of English-French Bitext. In *1st LREC*, volume 1, pages 489–494.

Uszkoreit, Jakob, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *23rd COLING*, pages 1101–1109.

Varga, D., L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy. 2005. Parallel corpora for medium density languages. In *3rd RANLP*, pages 590–596.

Wu, Dekai. 1994. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *32nd ACL*, pages 80–87.

Yu, Qian, Aurélien Max, and François Yvon. 2012. Revisiting sentence alignment algorithms for alignment visualization and evaluation. In *5th workshop BUCC*, pages 10–16.

Zhang, Min, Haizhou Li, A. Kumaran, and Ming Liu. 2012. Report of news 2012 machine transliteration shared task. In *Proceedings of the 4th Named Entity Workshop*, pages 10–20.