First Steps Towards Coverage-Based Document Alignment

Luís Gomes[1,2] **Gabriel Pereira Lopes**[1,2]

[1]NOVA LINCS, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
[2]ISTRION BOX, Translation and Revision, Lda, Portugal
{luis.gomes,gabriel.lopes}@istrionbox.com

Abstract

In this paper we describe a method for selecting pairs of parallel documents (documents that are a translation of each other) from a large collection of documents obtained from the web. Our approach is based on a *coverage* score that reflects the number of distinct bilingual phrase pairs found in each pair of documents, normalized by the total number of unique phrases found in them. Since parallel documents tend to share more bilingual phrase pairs than non-parallel documents, our alignment algorithm selects pairs of documents with the maximum coverage score from all possible pairings involving either one of the two documents.

1 Introduction

In this paper we describe our algorithm for bilingual document alignment, which is based on a coverage scoring function that reflects the ratio of unique bilingual phrase pairs from a Moses phrase table (Koehn et al., 2007) that are found in each bilingual pair of documents[1].

Basically, we exploit the fact that (parallel) phrase pairs are more likely to co-occur in parallel documents than in non-parallel ones. This insight came to our mind when we learned about the MT-based approach proposed by Sennrich and Volk (2010) to the closely related sentence alignment problem, which is to align parallel sentences within a pair of parallel documents. The MT-based approach to sentence alignment uses the BLEU score between sentences of one document and machine translated sentences of the other, as an indicator of parallelism between sentences. By using a phrase table directly we circumvent the decoding process which inevitably makes translation choices (and sometimes errors) that differ from the ones made by the human translators.

One may argue that using a decoder would have the advantage of avoiding "noisy" phrase pairs from the phrase table. However, we observed that most of the "noisy" phrase pairs in the phrase table *are not* completely unrelated. Instead, they sometimes miss a word or two on one of the sides, but are otherwise parallel to some extent. Nevertheless, since we employ uniform weighting for all phrase pairs (we treat them as binary features; either they are present in a document or not), the effect of noisy entries becomes diluted in a large number of features. For the most sceptical amongst us, please consider that even if the phrase table was created by a *random aligner*, the mere fact that the phrase pairs were sampled from parallel sentences, would cause parallel documents to statistically share more of such phrase pairs than non-parallel documents.

Our earlier successful application of coverage-based scores to the problem of sentence alignment (Gomes and Lopes, 2016) prompted us to develop a similar solution to the document alignment problem. The main characteristics of our approach are:

- it takes advantage of existing knowledge encoded in PBSMT phrase tables (we consider this to be our main characteristic, as it was our foremost goal to reuse existing knowledge);
- it identifies pairs of documents with various degrees of document parallelism ranging from barely comparable to parallel;
- it is language and domain[2] independent, as long as we can manage to create a phrase table for the pair of languages at hand from a relatively general-domain parallel corpus;

[1]hereafter we will avoid repeating the word *bilingual* whenever we mention pairs of documents or phrases

[2]here *domain* refers to text domain

Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, pages 697–702,
Berlin, Germany, August 11-12, 2016.

- it is purely content based (although this is not an advantage for the present shared task and other scenarios where metadata is available);
- it is agnostic with respect to document format (again, this is not an advantage in the present task, because all documents are HTML pages and some tag-structure features could be helpful)

2 Alignment Method Description

Our alignment method has three major steps: a *preparation* step, which constructs a phrase→document indexing data structure, a *candidate generation* step, which generates bilingual pairs of putative parallel documents, and finally, a *candidate selection* step which selects the pairs with maximum coverage score among all *competing candidates* from the generated set (we will define precisely what are *competing candidates*).

Each of these steps is described ahead, in dedicated sub-sections, but first we will define the *coverage score* which is the key concept of the whole method.

2.1 Coverage Score

We define the *coverage score* of a bilingual pair of documents as the geometric mean between two *coverage ratios*, one for each document. The coverage ratio of an English[3] document E when paired with a candidate parallel French document F is given by equation 1:

$$C(E,F) = \frac{|E \cap F|}{|E|} \tag{1}$$

Conversely, to compute the coverage ratio of a French document F when paired with a candidate English document E we simply swap E with F in the equation above.

More formally, the capital letters E and F represent the set of unique phrases present in each document (i.e. in this algorithm a document is represented by the set of unique phrases occurring in it). To compute the cross-lingual intersection of E and F we resort to the Moses phrase table which allows us to match phrases of both languages. Please note that some phrases are common to English and French, such as proper nouns, numbers, URLs, postal addresses, etc. We also consider such phrases as belonging to the cross-lingual intersection of E and F when computing the coverage score, even if they are not in the phrase table.

The coverage score of a candidate pair of documents, is given by a non-parametric combination of the two coverage ratios ($C(E,F)$ and $C(F,E)$). We chose the geometric mean (equation 2b) instead of the arithmetic (equation 2a) or harmonic (equation 2c) means, because it sits in the middle ground between the other two in terms of response to unbalanced inputs (see equation 2d). In fact, the equalities between the three means (equation 2d) only hold if the inputs a and b have the same value.

$$A(a,b) = \frac{a+b}{2} \tag{2a}$$

$$G(a,b) = \sqrt{ab} \tag{2b}$$

$$H(a,b) = \frac{2ab}{a+b} \tag{2c}$$

$$H(a,b) \leq G(a,b) \leq A(a,b) \tag{2d}$$

To better understand our choice of the geometric mean, let us consider for example three pairs of coverage ratios for three hypothetical pairings of documents: (0.9, 0.1), (0.65, 0.35) and (0.5, 0.5). The arithmetic mean of each of these pairs is 0.5 (the same for all pairs) while the geometric mean is 0.3 for the first, 0.48 for the second and 0.5 for the third, which is the most balanced pair. Therefore, if we use the arithmetic mean, then we will not differentiate among these three cases, although the pair with more balanced coverage ratios is more likely to be parallel. From observation we learned that extremely unbalanced coverage ratios typically indicate that one of the documents is much longer than the other. Since longer documents tend to have more unique phrases than shorter ones, whenever we compute the coverage ratios for such a pairing, the shorter document will have a greater coverage ratio than the the longer document. More precisely, the numerator of equation 1 is the same for both paired documents, but the denominator will be larger for the document with more unique phrases. The harmonic mean is slightly more sensitive to unbalanced input values than the geometric mean, and for the three pairings

[3] although we refer to English and French documents throughout the document, the algorithm is nonetheless language independent

in the previous example we would get 0.18, 0.46 and 0.5 (which are not too far from the respective geometric means). In future work we may experiment replacing the geometric with the harmonic mean, but we do not expect dramatic changes in the performance.

Replacing a and b in equation 2b by the equation 1 for both documents, we get the following equation for the coverage score:

$$S(E,F) = \left(\frac{|E \cap F|^2}{|E|\,|F|} \right)^{\frac{1}{2}} \quad (3)$$

For reasons explained in § 2.4, we will not simplify this equation.

2.2 Preparation Step

The preparation step is responsible for creating two phrase→document indices, one for each language, which are used later in the *candidate generation* step. In our prototype, these indices are implemented as hash tables mapping phrases (strings) into lists of document Ids (integers). The time needed for creation of these indices is proportional to the size of the document set, while the memory required is proportional to the number of unique phrases (hash table keys) times the average document-frequency of phrases (each phrase is associated with a list of unique document Ids where it occurs at least once). The creation of the indices is as simple as follows:

- for each document of a given web domain:
  - extract all unique phrases up to 5 tokens (the same maximum phrase length as the phrase table)
  - insert each phrase in the hash table of the respective language (if not there already) and append the document Id to the list of document Ids associated with each phrase

One important implementation detail is that the tokenization algorithm employed for processing the documents must be exactly the same as the one used to process the corpus from where the phrase table was extracted. In our prototype we used the tokenizer from the Moses toolkit (Koehn et al., 2007), and a pre-computed English-French phrase table extracted from the Europarl corpus (Koehn, 2005). Both the tokenizer and the pre-computed

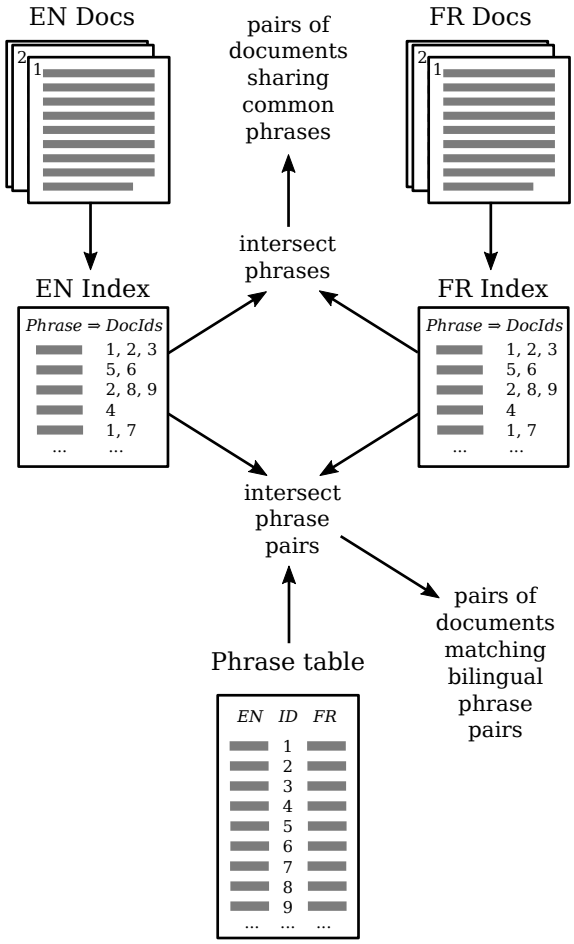

Figure 1: Overview of the candidate generation algorithm

phrase table were downloaded from the Moses website[4].

2.3 Candidate Generation Algorithm

The candidate generation algorithm is responsible for balancing the computation time required with the precision and recall of the aligner. If it generates too many candidates, then the aligner will take a long time to evaluate all generated candidates. If it generates too few candidates then there is an increased chance that some true parallel pairs are not among the generated candidates, and thus absent from the final aligner output.

For the smaller web domains, we may generate all possible pairings, thus ensuring that all true parallel pairs are passed into the selection step. However, in the general case, we need to prune the hypothesis space and generate only a subset of all possible pairs.

[4] http://www.statmt.org/moses/RELEASE-3.0/

Our heuristic for candidate generation is to define an average minimum number of candidates k to be generated for each document (we used k=100 in our experiments). Then, the global minimum number of candidates to be generated is computed by multiplying k by the average number of documents in both languages. For example, if there are 400 English documents and 600 French documents, and we set k=100, then the global minimum number of candidates to be generated is $k\frac{400+600}{2}$=50,000 which is much lower than the number of all possible pairings (400×600=240,000).

The algorithm generates candidate pairs incrementally, considering one bilingual pair of phrases at a time. It starts from the least frequent phrase pairs, which are also the most discriminant[5], and progresses towards the more frequent phrase pairs. For each bilingual phrase pair considered we generate all pairs of documents from the Cartesian product of the document Ids associated with each of the two phrases. Figure 1 shows an overview of the candidate generation algorithm and its interaction with the phrase indices and the phrase table.

As an example of this Cartesian product-based generation, if the English phrase "thank you" occurred in English documents 3 and 7 and the French phrase "merci" occurred in documents 2, 5 and 9, we would generate the following 6 candidate pairs: (3,2), (3,5), (3,9), (7,2), (7,5) and (7,9).

The candidate generation terminates when the required global minimum number of candidates has been generated.

2.4 Candidate Selection Algorithm

The candidate selection algorithm is responsible for selecting, among each group of *competing candidate pairs* (alternative hypotheses), the one with maximum coverage score.

We define *competing candidate pairs* of documents as pairs that share either of the two documents. For example, the pairs (E_1, F_1) and (E_1, F_2) are *competing pairs* since they share the English document E_1, but the pairs (E_1, F_1) and (E_2, F_2) are not. We assume that only one pair of all competing candidate pairs is indeed parallel, i.e. there is at most one parallel French document for each English document (and vice versa).

More formally, the selection algorithm selects pairs of documents (E, F) that verify the following two inequalities, which warrant a maximum coverage among all competing pairs:

$$S(E,F) > S(E,\hat{F}) \quad \forall \hat{F} \neq F \qquad (4a)$$

$$S(E,F) > S(\hat{E},F) \quad \forall \hat{E} \neq E \qquad (4b)$$

We call the selected (E, F) pairs as *maximal pairs*.

Please recall the coverage score equation $S(E, F)$ (equation 3), and its wrapping square root which we did not simplify away. Since the square root is a monotonically increasing function, and given that we are comparing coverage scores of competing pairings instead of looking at their absolute values, we may drop the square root from the equation and the comparisons across competing candidates will hold the same as before. Thus, we save a few computer clocks per candidate pair analysed.

3 Evaluation

The evaluation in this shared task is based on recall, i.e. the ratio of URL pairs from the testset that are correctly identified by the aligner. A one-to-one rule is enforced, which allows each English URL to be aligned with at most one French URL and vice versa.

Despite the merits of a pure content-based approach, which is applicable in scenarios where URLs and other metadata are not available, we acknowledge that for the present task we may obtain better results if we take advantage of all information available (page URL and HTML structure) besides the plain text content.

Therefore, besides evaluating our content-based method on its own, we submitted two additional *extended* sets of results obtained by trivial combinations of our content-based method with the metadata-based (URL-based) baseline method.

The first extended set, called *coverage/url*, gives priority to predictions of the coverage-based method, adding only URL-predicted pairs for URLs that were not aligned by the coverage-based method. Conversely, the second extended set, called *url/coverage*, gives priority to the predictions of the URL-based aligner.

The results obtained with our coverage-based aligner and the two trivial combinations with the baseline aligner for the development and test sets are summarized in Tables 1 and 2, respectively.

[5] A phrase pair that occurs in almost every document (such as the pair "the"↔"la") has very little discriminative power.

Method	Recall	# Predicted Pairs
baseline	67.92%	119979
coverage	72.78%	63207
coverage/url	89.53%	147857
url/coverage	90.52%	148278

Table 1: Evaluation results on the development set.

Method	Recall	# Predicted Pairs
baseline	53.03%	136086
coverage	85.76%	207022
coverage/url	88.63%	235763
url/coverage	94.96%	235812

Table 2: Evaluation results on the final test set.

The coverage-based aligner, alone, improves 5% over the baseline on the development set and 33% on the test set. But when combined with the baseline aligner, the recall is boosted up to 23% above the baseline on the development set and up to 42% on the test set. A possible explanation for the boosted recall is that since the methods rely on completely different feature sets, their predictions are to some degree complementary.

We would like to point out that the coverage-based aligner made substantially fewer predictions than the baseline (52.7%) in the development set, and still yielded higher recall (+4.86%). This allows us to speculate that the precision of the coverage-based alignment is likely to be higher than the precision of the baseline.

4 Future Work

This was our first look into the document alignment problem and a number of ideas for improving the current algorithm sprung up during the experiments. Next, we will briefly discuss ideas which we intend to explore in future work.

4.1 Improve Candidate Generation Strategy

The candidate generation algorithm presented in §2.3 is perhaps the weakest point of the whole method. We arrived at this conclusion when we noticed that many URL pairs from the development set were not being generated due to a too low frequency threshold, particularly for the largest domains. When we tried to counter this effect by increasing the threshold, then the algorithm started to exhibit square asymptotic complexity, taking too long to align the larger domains. In the meantime, we discovered a better candidate generation strategy, but unfortunately, it was not possible to implement it on time for this shared task. The main difference is that instead of a global frequency threshold, we fix a minimum number of *competing candidates* to be compared with each document.

4.2 Better Integration With Metadata-based Features

As described earlier, we submitted two extra datasets resulting from trivial combinations of our aligner and baseline outputs. Due to lack of time, we didn't try more sophisticated forms of combining our content-based features with other kinds of feature, such as URL matching and HTML document structure as proposed in the Bitextor paper (Esplà-Gomis et al., 2010).

Since the trivial combinations achieved the best performance in the development set, we expect to improve the performance further still, if we combine content-, structure- and metadata-based features in a more principled manner.

One possibility for making use of URLs would be to consider the path component of URLs as a slash-delimited sentence, and match phrases from this sentence in the same way that we do for phrases in the text. Therefore, even if the URLs are not exactly identical (after stripping language-specific markers such as "lang=en"), they could still match partially.

4.3 Using Document Structure Information

Following the idea introduced by Bitextor (Esplà-Gomis et al., 2010), we could also compute document similarity based on HTML tag structure, given that many parallel webpages also have a parallel HTML structure. They propose a distance measure, based on edit-distance of a sequence of tags intermixed with text chunks (represented by their length). The computation of the distance measure takes $O(NM)$ time to compute, for a pair of documents with N and M tags respectively. This may be computationally expensive, particularly for large web domains, but we might resort to this measure only for documents with a very low coverage score and/or a very small distance to the second choices in the selection algorithm described in section 2.4.

5 Conclusion

The bilingual document alignment algorithm presented in this paper has several interesting properties, in our view: it is language and domain independent, it is able to align documents with varying degrees of parallelism (ranging from barely comparable documents to fully parallel ones), and it is purely content-based, which makes it applicable in a wider range of scenarios.

On its own, the performance of our aligner is above the baseline, but is not outstanding: 73% recall on the development set and 86% on the test set. But when combined with the URL-based predictions of the baseline aligner, we achieve 90% recall on the development set and 95% on the test-set. The performance boost of the combined systems may be explained by the complementary nature of the features employed by each method.

Finally, we believe that the *phrase-table-coverage* approach still has room for improvement, because this was our first look into the problem and we have several untried ideas for improving it.

Acknowledgements

This work was supported by ISTRION BOX, Translation and Revision, Lda, and the Portuguese Foundation for Science and Technology (FCT) through individual PhD grant (ref. SFRH/BD/65059/2009), research project ISTRION (ref. PTDC/EIA-EIA/114521/2009), and NOVA LINCS (ref. UID/CEC/04516/2013).

References

Miquel Esplà-Gomis, Felipe Sánchez-Martínez, and Mikel L. Forcada. 2010. Combining content-based and url-based heuristics to harvest aligned bitexts from multilingual sites with Bitextor. *The Prague Bulletin of Mathemathical Lingustics*, 93:77–86.

Luís Gomes and Gabriel Pereira Lopes. 2016. First steps towards coverage-based sentence alignment. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia. European Language Resources Association (ELRA).

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5.

Rico Sennrich and Martin Volk. 2010. Mt-based sentence alignment for ocr-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010), Denver, Colorado*.