Simple and accurate method for parallel web pages detection

Alibi Jangeldin and Zhenisbek Assylbekov

School of Science and Technology, Nazarbayev University
{alibi.jangeldin, zhassylbekov}@nu.edu.kz

Abstract. This paper presents language independent method for measuring structural similarity between web pages from bilingual websites. First we extract a new feature from those which are used by the STRAND architecture and combine it with the existing one. Next we analyze properties of this feature and develop an iterative algorithm to infer the parameters of our model. Finally, we propose an unsupervised algorithm for detecting parallel pairs of web pages based on these features and parameters. Our approach appears to benefit the structural similarity measure: in the task of distinguishing parallel web pages from five different bilingual websites the proposed method is competitive with other unsupervised methods.

Keywords: parallel corpora · document alignment · structural similarity · STRAND · unsupervised learning.

1 Introduction

World Wide Web is an important source of parallel corpora (bitexts), since many websites are available in two or more languages. Many approaches have been therefore proposed for trying to exploit the Web as a parallel corpus: STRAND [11], PTMiner [3], BITS [8], WPDE [15], Bitextor [5], ILSP-FC [9], etc. The task of extracting bitexts from the Web typically involves the following four major consecutive steps: (i) data collection (crawling), (ii) document alignment, (iii) sentence splitting and (iv) sentence alignment.

Very often websites are already aligned on document level, i.e. web pages in one language contain links to the translated versions of themselves. However, even in this case one often encounters misaligned pairs, e.g. when a translated web page is missing (dead link) or documents are not exact translations of each other (comparable texts). To address these issues one usually needs to perform additional filteri ng on document level in order to detect and discard non-parallel document pairs. For this task three main strategies can be found in the literature – they exploit: (i) similarities in URLs; (ii) structural similarity of HTML files; (iii) content-similarity of texts. In this paper we address the second strategy, i.e. measuring structural similarity between HTML documents. We *do not* involve content-similarity metrics here as our goal is to have a language-agnostic tool.

In this paper we develop a simple yet accurate language-independent technique for measuring structural similarity between HTML pages, which uses the same amount of information as previous approaches to distinguish parallelism of web pages and can be applied in unsupervised manner.

2 Related work

Measuring structural similarity between HTML files was first introduced in [10], where a linearized HTML structure of candidate pairs was used to confirm parallelism of texts. Later approaches combined structural similarity metrics with other measures. E.g. Shi et al. [13] additionally used a file length ratio and a sentence alignment score. Zhang et al. [15] used file length ratio and content translation to train k-nearest-neighbors classifier for parallel pairs verification. Esplà-Gomis and Forcada [5] used text-language comparison, file size ratio, total text length difference for preliminary filtering and then HTML tag structure and text block length were used for deeper filtering. In [12] the bitext detection module runs three major filters: link follower filter, URL pattern search, and a combination of an HTML structure filter and a content filter. In [9] structural filtering is based on length ratios and edit distances between linearized versions of candidate pairs. Liu et al. [7] proposed a link-based approach in conjuction with content-based similarity and page structural similarity to distinguish parallel web pages from bi-lingual web sites.

All of these works require labeled data for model training or thresholds estimation in a supervised way, but there is usually not enough labeled parallel corpora for every language pair to do it and manual labeling is expensive. Our work is mainly motivated by unsupervised approach of Assylbekov et al. [1], who developed a statistical model for measuring structural similarity between web pages; they were using raw counts for text lengths and misalignment of HTML tags. In this paper we show that it is more reasonable to consider normalized quantities; we perform a detailed analysis of their distributions, propose an iterative algorithm to infer the parameters of our model in unsupervised manner and introduce two tuning parameters for it. Due to its simplicity the proposed approach can be used both for structural filtering and as a competitive baseline.

3 Methodology

3.1 Features extraction

Let us assume that candidate pairs are linearized as in STRAND and linearized sequences are aligned using a standard dynamic programming technique [6]. For example, consider two HTML-files, in English and in Kazakh[1], that begin as in Figure 1. Then the aligned linearized sequences would be as in Figure 2.

[1] Kazakh is a Turkic language belonging to the Kipchak branch, with approximately 11 million native speakers

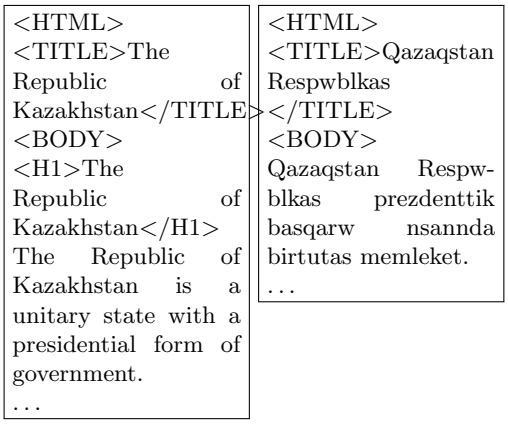

Fig. 1: Raw HTML files

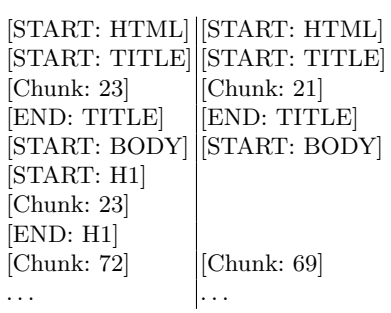

Fig. 2: Aligned and linearized sequences

Let M_1 and M_2 denote the total numbers of alignment tokens in the first and the second documents; let L_1 and L_2 denote the total character lengths of the texts in the first and the second documents respectively. Let W be the total number of alignment tokens that are present in one file, but not the other (alignment cost). In our example, $M_1 = 9$, $M_2 = 6$, $L_1 = 118$, $L_2 = 90$ and $W = 3$.

It is critical to note that we are dealing with the task of unsupervised learning, since initially it is unknown which pairs of web pages are translated in a proper way, and thus meaningful patterns that are inherent to parallel documents should be extracted from the available data. We construct two new features that we believe to be helpful to discriminate between parallel and nonparallel pairs:

$$P_d = \frac{W}{M_1 + M_2} \qquad \text{and} \qquad L_d = \frac{L_1 - L_2}{L_1 + L_2}.$$

Furthermore, we discuss in more details why they are chosen by constructing illustrative examples for each feature. `akorda.kz`[2] and `pm.gc.ca`[3] websites will be used to visualize our work process. Our basic assumption throughout the paper is that *most of the web pages on a bilingual website are translated correctly.*

Let us consider two pairs of web pages with the following features: $M'_1 = M'_2 = 10$ and $W' = 2$, and $M''_1 = M''_2 = 100$ and $W'' = 2$. Comparing their parallelism only in terms of alignment cost gives equality ($W' = W'' = 2$), but with such difference in total number of tokens this conclusion may be misleading. To solve this issue, W is adjusted by the total number of the alignment tokens: $P_d = W/(M_1 + M_2)$. Now $P'_d = 0.1$ and $P''_d = 0.01$ are more trustworthy to differentiate between the two cases. Since alignment cost is a measure of difference between a pair of web pages, the closer P_d of a pair to 0 the more

[2] Official site of the President of the Republic of Kazakhstan
[3] Official site of the Prime-minister of Canada

parallel it should be. This is supported by our basic assumption that most of the web pages are parallel and can be noticed in Figures 3 and 4.

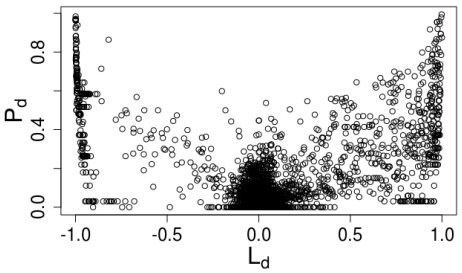

Fig. 3: L_d versus P_d for akorda.kz

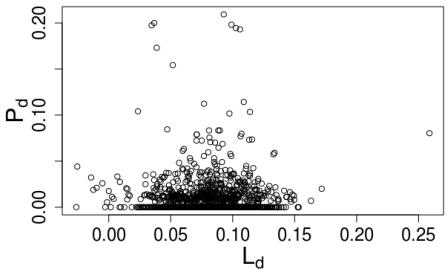

Fig. 4: L_d versus P_d for pm.gc.ca

Furthermore, can we use $|L_d| = \frac{|L_1 - L_2|}{L_1 + L_2}$ as it was done in [5] to measure parallelism in terms of scaled difference in total text length? To answer this question suppose that there are two pairs of web page translations such that their character length features are: $L'_1 = 400$ and $L'_2 = 600$, $L''_1 = 600$ and $L''_2 = 400$. Thus, the equality $|L'_d| = |L''_d| = 0.2$ means that these two pairs should be equally parallel in terms of difference in number of characters. However, in this case we are losing some information on linguistic difference between languages, e.g. if texts in the first language are on average shorter than in the second language, i.e. $E(L_1) < E(L_2)$, then $L_d \to 0^-$ should be a better evidence towards parallelism than $L_d \to 0^+$. Thus, $|L_d|$ may be properly used only for language pairs with $E(L_d)$ closer to 0 as in Figure 5 with Kazakh-English language pair. Thus, we may need more robust estimation of $E(L_d)$. The difference between language pairs was the most prominent when we analyzed `pm.gc.ca` website where total character length in French language was on average 17 percent longer than in English ($E(L_d) = 0.082$), illustrated in Figure 6. Thus, it's important to use L_d instead of $|L_d|$ in order to differentiate between language pairs.

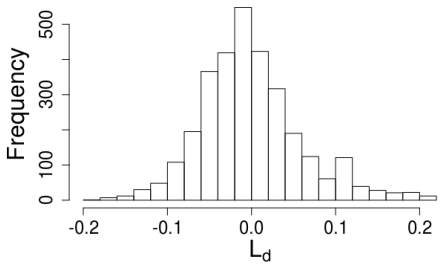

Fig. 5: Histogram of L_d for akorda with (3)

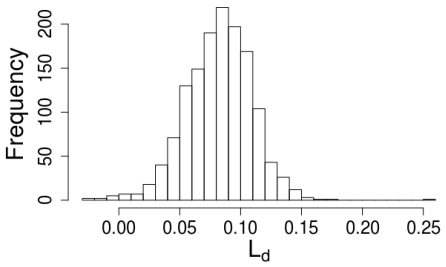

Fig. 6: Histogram of L_d for pm.gc.ca

As mentioned above, initially we do not know which pairs of web pages are parallel: if it was known we could estimate proper thresholds for P_d and L_d from such data. Hence we need another way of finding threshold values for P_d and L_d, that will allow us to discriminate between parallel and non-parallel pairs of web pages.

3.2 Thresholds estimation

Let's formally derive our insight for L_d. For parallel pairs of web pages in a given language pair there is almost linear relationship between L_1 and L_2:

$$L_1 \approx c \cdot L_2,$$

where the constant c depends on the relationship between two languages. Under such assumption, L_d can be approximated as:

$$L_d = \frac{L_1 - L_2}{L_1 + L_2} \approx \frac{c-1}{c+1}. \tag{1}$$

By comparing P_d and L_d one may notice that L_d will have different distributions for different language pairs due to their linguistic differences and thus, will depend on c. Therefore we will need to adapt it to every language pair. On the other hand, P_d has the same interpretation for any language pair since number of alignment tokens is independent of language pairs. Thus, threshold proposed by the author of STRAND method [11] for P_d will be used in our work in accordance with our basic assumption. The rationale for this can be noticed from Figures 3 and 4 in which majority of the points are below 0.2-level on P_d-axis.

One more thing to notice in L_d- P_d scatter plot is that parallel pairs of web pages are grouped around mean value of L_d which, according to (1), should be close to $(c-1)/(c+1)$. We will need this value for our further analysis and if we try to estimate it using the average of L_d across *all* pairs, it may be severely biased by the values of nonparallel pairs (see Figure 3). Hence we need a more robust estimation of $E(L_d)$ for *parallel* pairs. For this purpose we extract a subset of pairs using the following rule

$$D = \{\text{document pairs for which } P_d = 0\}, \tag{2}$$

and we calculate sample mean of L_d on D, let us denote it by μ, since the condition $P_d = 0$ provides strong evidence towards parallelism of pairs supported by our basic assumption. Comparison of mean estimation methods for `akorda.kz` website shows that estimation of the mean for all pairs gives $E(L_d) = 0.033$ and from the Figure 5 it seems that the bias comes from the right tail. Subset of pairs with $P_d = 0$ condition has $E(L_d) = -0.005$ that is much closer to the center of symmetry. For `pm.gc.ca` the new value is almost the same as previous mean with $E(L_d) = 0.083$ because most of the pairs are already parallel there before subsetting. Using the obtained estimate μ we will try to model the distribution of L_d around it.

For the illustrative purposes let us choose 0.2 thresholds for both P_d and $|L_d - \mu|$, i.e.

$$P_d < 0.2 \quad \text{and} \quad |L_d - \mu| < 0.2, \tag{3}$$

and look at the behavior of L_d on pairs which satisfy (3) – its distribution is given in Figure 5.

One can see that it is symmetric around the mean and this property will be used to estimate threshold for L_d in unsupervised way. Essential part of our work is described in Algorithm 1. The main idea is to start with a small threshold for L_d which reliably separates parallel pairs from nonparallel and iteratively increase it while we are confident enough in parallelism of added pairs.

Algorithm 1 L_d threshold estimation

Require: $S \leftarrow$ subset of pairs with $P_d < 0.2$
Ensure: *threshold* – threshold for L_d
  $D \leftarrow \{s \in S : P_d = 0\}$
  $\mu \leftarrow E_D(L_d)$
  $threshold \leftarrow 0.01$
  $step \leftarrow 0.01$
  $N \leftarrow \#\{s \in S : |L_d - \mu| < threshold\}$
  $\Delta \leftarrow 1$
  while $\Delta \geq 0.01$ **do**
    $threshold \leftarrow threshold + step$
    $N_{new} \leftarrow \#\{s \in S : |L_d - \mu| < threshold\}$
    $\Delta \leftarrow N_{new}/N - 1$
    $N = N_{new}$
  end while

After threshold estimation, our suggestion is to predict the following subset of pairs as parallel:

$$P_d < 0.2 \qquad \text{and} \qquad |L_d - \mu| < threshold, \tag{4}$$

The loop described above should be finite because majority of the candidate pairs in all websites are parallel and thus have $P_d < 0.2$ and are located around the mean μ of L_d from which we start iteration. There are two main parameters that may be tuned in the algorithm: Δ threshold and *step* value. The trade-off between precision and recall may be addressed by tuning threshold for Δ. Precision may be maximized by increasing threshold value and thus decreasing threshold for L_d whereas recall can be maximized conversely. On the other hand, *step* value may be tuned to optimize the trade-off between computational efficiency and accuracy of the *threshold* value estimation for L_d. Thus, higher *step* value will lead to faster convergence, whereas lower *step* value will be beneficial for *threshold* estimation. Nevertheless, these two parameters are interdependent and tuning either of them will affect both of the trade-offs described above.

4 Experiments and results

4.1 Data sets and evaluation criteria

To evaluate the performance of our algorithm we used web pages from 5 different sites. To obtain candidate pairs from those websites we used `GNU wget`[4] tool. Since the obtained pairs are *candidate* pairs, there is no guarantee that all of them are parallel. General information about the websites, including URLs, short description, languages and total numbers of candidate pairs, is given in Table 1. From each of the websites we extracted representative samples (sample sizes

Websites:	Description	Languages	Number of pairs	Sample size
`akorda.kz`	President of Kazakhstan	kk-en	4135	352
`egov.kz`	Electronic government of Kazakhstan	kk-en	2400	312
`mfa.kz`	Ministry of Foreign Affairs of Kazakhstan	kk-en	180	180
`presidencia.pt`	President of Portugal	pt-en	960	275
`pm.gc.ca`	Prime-minister of Canada	fr-en	1397	302

Table 1: Information about websites

were calculated using the Cochran's formula [4]) and manually checked them for parallelism. These samples are used as test sets with sizes given in Table 1.

We used precision, recall and F_1-score for performance evaluation:

$$prec = \frac{tp}{tp + fp}, \quad rec = \frac{tp}{tp + fn}, \quad F_1 = \frac{2 \cdot prec \cdot rec}{prec + rec},$$

where tp is the number of true positives (i.e. number of pairs in a sample correctly labeled as parallel) and fn is the number of false negatives (i.e. number of pairs in a sample incorrectly labeled as non-parallel).

4.2 Baseline and other approaches

In the following we describe a baseline and other approaches for parallel web pages detection which will be contrasted to our method.

Baseline. We use the STRAND's default thresholds from [11]:

$$P_d < 0.2 \qquad \text{and} \qquad p\text{-value} < 0.05,$$

where p-value corresponds to the significance of correlations between the lengths of aligned text chunks.

[4] `http://www.gnu.org/software/wget`

Hierarchical clustering (HC). We apply hierarchical clustering [14] with average linkage using the same features P_d and L_d as in Algorithm 1 and different ways of choosing the number of clusters:

HC1. setting two clusters in a belief that document pairs will naturally split into parallel and non-parallel,
HC2. using CH-index [2] to automatically calculate the appropriate number of clusters based on within-cluster and between-cluster variances,

In all of these approaches we assumed the largest cluster to contain parallel pairs and other clusters to contain nonparallel pairs according to our basic assumption.

Statistical model (Algorithm 0). In this approach described in detail in [1] raw features were used as inputs and parameters were estimated in unsupervised way using EM algorithm. This work derives a joint distribution for W, M, N, L_1, and L_2 in a rigorous way throwing in independence assumptions along the derivation. Our approach is more empirical and much more simple.

4.3 Results

We applied all the methods mentioned above, i.e. baseline STRAND, HC1, HC2, HC3, Algorithm 0 and Algorithm 1, to the websites from Table 1, and resulting precision, recall and F_1-scores are provided in Table 2. As we see from Table 2 there is no single method which consistently outperforms all others. Our suggested Algorithm 1 did best on websites from the `.kz` domain which use Kazakh-English language pair, however HC2 was better on `presidencia.pt` (Portuguese-English) where Algorithm 1 showed good results as well. One can see that clustering showed good performance only on the websites with high quality of translation where signal-to-noise ratio is lower, but it still was not stable in those. The main reason why it outperformed proposed approach may be the fact that clustering algorithm is nonlinear, whereas the proposed algorithm has conservative linear boundaries used to maximize precision which may result in losing some of the parallel web pages in the boundaries.

Method	`akorda.kz`			`egov.kz`			`mfa.gov.kz`			`presidencia`			`pm.gc.ca`		
	prec	*rec*	F_1	*prec*	*rec*	F_1	*prec*	*rec*	F_1	*prec*	*rec*	F_1	*prec*	*rec*	F_1
Baseline	92.5	87.5	89.9	100	76.3	86.6	96.0	97.6	96.8	96.6	91.3	93.9	99.0	95.3	97.0
HC1	79.5	100	85.6	75.3	100	86.2	95.8	94.1	95.0	91.8	100	95.7	99.3	100	**99.7**
HC2	87.5	100	93.3	87.2	98.9	92.7	100	31.1	47.5	98.8	99.4	**99.1**	99.4	58.1	73.3
Alg'm 0	94.1	97.1	95.6	91.5	96.9	94.1	94.4	100	**97.1**	99.1	95.0	97.0	99.0	1.00	99.5
Alg'm 1	94.5	98.6	**96.5**	100	90.5	**95.0**	95.5	98.8	**97.1**	97.4	97.7	97.5	99.3	100	**99.7**

Table 2: Results

Percentages of parallel pairs in test sets, estimated threshold values and number of iterations to converge in Table 3 will be used to analyze effectiveness of the proposed method in more details. Process of the while loop convergence in Algorithm 1 for the five data sets is illustrated with corresponding Δ values in Figure 7. One may notice that websites with .kz domain require more iterations to converge than others due to linguistic differences between language pairs, this explains why Δ values of these websites are fluctuating. Nevertheless, the values for all websites are steadily declining at the last three steps before convergence.

Percentages of parallel pairs in the test sets allow us to estimate quality of translations on the considered web sites. Empirical results show that the proposed algorithm is demonstrating better results on websites with higher quality of translation. The main limitations of our algorithm arise from the structural similarity measures. In most of the false positive pairs there are missing sentences in one or both sides and we believe that it would be easier to detect such page pairs with other approaches. On the contrary, some of false negatives have P_d and L_d values that are close to threshold values and therefore can be reached by increasing the thresholds.

Websites:	% of parallel pairs	*threshold*	iterations
`akorda.kz`	0.740	0.15	13
`egov.kz`	0.751	0.24	22
`mfa.kz`	0.944	0.16	14
`pr-cia.pt`	0.915	0.09	7
`pm.gc.ca`	0.990	0.08	6

Table 3: Properties of websites

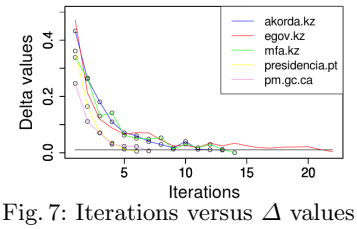

Fig. 7: Iterations versus Δ values

5 Conclusion and future work

In this paper we propose unsupervised method for parallel web pages detection. Feature engineering and analysis of properties of the new feature allowed us to efficiently estimate a threshold for it in unsupervised manner by considering inherent differences between language pairs and tuning parameters of the iterative algorithm. Empirical results show that the proposed approach is competitive with the previous unsupervised approaches and reproducible for further work.

Next step in this work may be to combine the proposed approach with content-similarity of texts as a deeper filtering method. Additionally, it would be good for robustness to make sure that pages in a candidate pair are in different languages, so that we are not measuring similarity of two identical pages. We are planning to use it for compiling large-scale Kazakh-Russian and Kazakh-English parallel corpora. Once it is done, modern approaches in statistical machine translation will be used to build competitive machine translation systems for Kazakh.

Bibliography

[1] Assylbekov, Z., Nurkas, A., and Mouga, I. R. (2015). A statistical model for measuring structural similarity between webpages. In *Recent advances in natural language processing*, pages 24–31.

[2] Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

[3] Chen, J. and Nie, J.-Y. (2000). Automatic construction of parallel english-chinese corpus for cross-language information retrieval. In *Proceedings of the sixth conference on Applied natural language processing*, pages 21–28. Association for Computational Linguistics.

[4] Cochran, W. G. (2007). *Sampling techniques*. John Wiley & Sons.

[5] Esplà-Gomis, M. and Forcada, M. (2010). Combining content-based and url-based heuristics to harvest aligned bitexts from multilingual sites with bitextor. *The Prague Bulletin of Mathematical Linguistics*, 93:77–86.

[6] Hunt, J. W. and MacIlroy, M. (1976). *An algorithm for differential file comparison*. Bell Laboratories.

[7] Liu, L., Hong, Y., Lu, J., Lang, J., Ji, H., and Yao, J. (2014). An iterative link-based method for parallel web page mining. *Proceedings of EMNLP*, pages 1216–1224.

[8] Ma, X. and Liberman, M. (1999). Bits: A method for bilingual text search over the web. In *Machine Translation Summit VII*, pages 538–542.

[9] Papavassiliou, V., Prokopidis, P., and Thurmair, G. (2013). A modular open-source focused crawler for mining monolingual and bilingual corpora from the web. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 43–51.

[10] Resnik, P. (1998). *Parallel strands: A preliminary investigation into mining the web for bilingual text*. Springer.

[11] Resnik, P. and Smith, N. A. (2003). The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

[12] San Vicente, I. and Manterola, I. (2012). Paco2: A fully automated tool for gathering parallel corpora from the web. In *LREC*, pages 1–6.

[13] Shi, L., Niu, C., Zhou, M., and Gao, J. (2006). A dom tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 489–496. Association for Computational Linguistics.

[14] Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

[15] Zhang, Y., Wu, K., Gao, J., and Vines, P. (2006). Automatic acquisition of chinese–english parallel corpus from the web. In *Advances in Information Retrieval*, pages 420–431. Springer.