A Method for Fine-Grained Document Alignment Using Structural Information

Naoki Tsujio, Toshiyuki Shimizu, and Masatoshi Yoshikawa

Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
tsujio@db.soc.i.kyoto-u.ac.jp, {tshimizu,yoshikawa}@i.kyoto-u.ac.jp

Abstract. It is useful to understand the corresponding relationships between each part of related documents, such as a conference paper and its modified version published as a journal paper, or documents in different versions. However, it is hard to associate corresponding parts which have been heavily modified only using similarity in their content. We propose a method of aligning documents considering not only content information but also structural information in documents. Our method consists of three steps; baseline alignment considering document order, merging, and swapping. We used papers which have been presented at a domestic conference and an international conference, then obtained their alignments by using several methods in our evaluation experiments. The results revealed the effectiveness of the use of document structures.

Keywords: document alignment, structured document, cross-lingual alignment.

1 Introduction

We can obtain the correspondence between parts of given documents which are in different forms for the same topic or in different versions. For example, a paragraph from one document corresponds to a paragraph or paragraphs in another document. Such documents include pairs of a conference paper presented at an international conference and a journal paper, which was an improved version of the conference paper, or two distinct versions of Wikipedia articles.

The information on corresponding relationships is very useful to understand documents. For example, if one feels that part of a document is hard to understand, the corresponding parts in another document can help one to better understand the given part. In addition, if one wants to find differences between a new document and a document which one has read in the past, corresponding information can provide them. Furthermore, they can be used to complement parts which have poor content by using corresponding parts which have more content. Other important applications include detection of plagiarism.

However, it is difficult to associate parts which have been heavily modified only using information from their content. Therefore, we considered using not only information from content but also the structural information in documents

L. Chen et al. (Eds.): APWeb 2014, LNCS 8709, pp. 201–211, 2014.

such as sections or paragraphs. We can align and associate parts appropriately by using structural information.

Much important content on the Web can be modeled as structured documents. Our motivation is to associate corresponding parts in structured documents, and we propose a method of using the content and structural information of documents for fine-grained document alignments.

Using document structures would be especially effective for cross-lingual cases. It is a way to use machine translation such as Google Translate[1] in advance in order to take alignments of cross-lingual documents. However, machine translation does not have sufficiently good quality to obtain appropriate alignments by only using information from content. In contrast, as structural information is not changed by machine translation, it would be effective for cross-lingual alignments.

We carried out experiments to confirm the effectiveness of using document structures for alignments, where we used pairs of papers such as a conference paper and its modified version which had been published as a journal paper. The results revealed that we could obtain more appropriate alignments by using document structures and there were especially effective for cross-lingual cases.

This paper is organized as follows. Section 2 discusses related work and its differences from our work. Section 3 explains the method of alignment. Section 4 presents results obtained from experiments and discusses the results. Conclusions and future work are presented in Section 5.

2 Related Work

2.1 Document Alignment

Document alignment has been widely studied. Daumé III and Marcu [1] considered alignments at the phrase level between one document and its abstract, and proposed a method using a hidden Markov model (HMM). Jeong and Titov [2] used a Bayesian model for alignments.

Our motivation was to associate corresponding parts of documents, i.e., to obtain alignments, which was similar to the previous work. However, they considered alignments based on information from the content of documents. In contrast, we introduced structural information into alignments.

Romary et al. [3] proposed a multilevel alignment method for structured documents. However, their target structure is balanced tree and they do not consider swapping.

2.2 Similarity of Documents

The idea of using document structures has been presented in studies on calculating similarities in documents. Zhang and Chow [4] regarded a document as having a two-level tree structure which had a root node representing the entire

[1] http://translate.google.com/

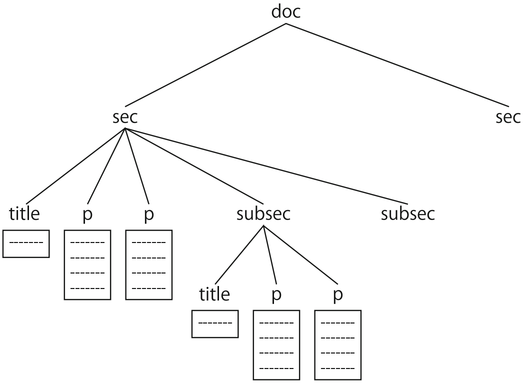

Fig. 1. Example of document tree

document and child nodes representing paragraphs, and they calculated similarities by using the Earth Mover's Distance (EMD). They also applied their method to detect plagiarism [5]. Wan [6] regarded a document as a sequence of its subtopics and he calculated similarities with EMD. Tekli and Chbeir [7] used the Tree Edit Distance to calculate similarities in XML documents.

They considered document structures in their work, but their motivation was to calculate similarities of documents, which was different from our motivation to find alignments.

2.3 Cross-Lingual Alignment

Yahyaei et al. [8] proposed a method using Divergence from Randomness. Yeung et al. [9] regarded document alignment to be a matter where Wikipedia pages had various quantities of descriptions for various languages. They used alignment technique to complete their content by using the same pages written in different languages. Smith et al. [10] used alignment to gather parallel sentences for training of statistical machine translation (SMT). Vu et al. [11] proposed a method to obtain similar news texts written in different languages.

Our method can be applied to cross-lingual alignments by using machine translation. Their work focused on content information of documents. On the other hand, our work introduce the structural information into alignments.

3 Alignment Method

We regard structured documents as tree structures and call them document trees (Figure 1). The input for our method is two document trees and the output is the result from alignment. The granularity of alignment is paragraphs.

If corresponding parts are heavily modified, it is hard to associate them by only using similarities in their content. Therefore, our method takes into considerations the order of paragraphs and the structures of documents.

Our method consists of three steps, where we use the Levenshtein distance algorithm to consider the order of paragraphs in the first step. We not only consider the similarities in paragraphs but also those in sections to associate two paragraphs. We consider merging paragraphs in the second step to associate multiple paragraphs. We consider swapping paragraphs in the third step to deal with changing paragraph locations. Each step is described in subsections 3.1, 3.2, and 3.3.

3.1 Baseline Alignment

We use the algorithm of the Levenshtein distance [12] for the baseline alignment. The Levenshtein distance is calculated as

$$d(s_i, t_j) = \min \begin{pmatrix} d(s_{i-1}, t_j) + cost_{del}(s[i]), \\ d(s_i, t_{j-1}) + cost_{ins}(t[j]), \\ d(s_{i-1}, t_{j-1}) + cost_{ren}(s[i], t[j]) \end{pmatrix} \tag{1}$$

s_i is the substring of string s which has i characters from the begining of s, and $s[i]$ is the i th character of s. The same thing can be said for t_j, $t[j]$, and t.

We regard documents as paragraph sequences and obtain alignment by applying this algorithm to them. When the bottom expression in Eq. (1) is minimum, i.e., when a rename operation is applied, we associate $s[i]$ and $t[j]$. Thus, we can obtain alignment which takes into account the order of paragraphs.

We set the rename cost of similar paragraphs to a lower value to associate similar paragraphs. The cost functions are defined as

$$cost_{del}(p) = cost_{ins}(p) = \alpha \qquad (0 \leq \alpha \leq 1) \tag{2}$$

$$cost_{ren}(p, q) = 1 - similarity(p, q) \tag{3}$$

where p and q are paragraphs. As delete and insert operations are symmetric, we set their costs to be equal. The *similarity* calculates the similarities between p and q in 0 to 1.

In order to associate two paragraphs which are hard to associate only using their similarity, we use sim_{global} which is the similarity between the sections which contain the paragraphs to be associated and sim_{local} which is the similarity between the paragraphs. sim_{global} and sim_{local} are calculated as the cosine similarity between the term vectors by tf-idf. The *similarity* is defined as follows, where C is a parameter which is in 0 to 1.

$$similarity(p, q) = C sim_{global}(p, q) + (1 - C) sim_{local}(p, q) \tag{4}$$

Thus, the similarity between sections will help in associating paragraphs.

This is the association using the Levenshtein distance algorithm. Figure 2 outlines an example of alignment in step 1, where associated paragraphs are

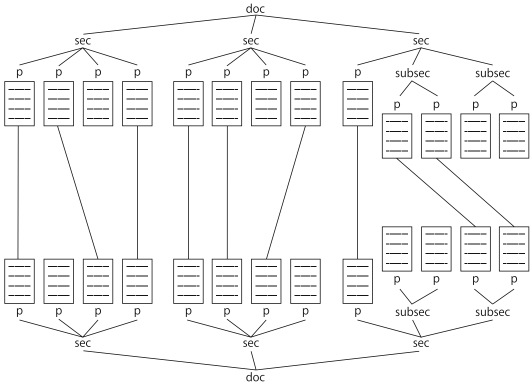

Fig. 2. Example of alignment in step 1

linked in the lines. Paragraphs which are not associated are regarded as deleted or inserted in this step. The alignment process is finished when all paragraphs or no paragraphs are associated in this step.

3.2 Merging

Paragraphs can be split into multiple parts, but the alignments in the previous step are in 1-to-1. We then need to consider alignments in m to n. we can do this by merging paragraphs.

The merging targets are paragraphs which have not been associated yet in the previous step (Section 3.1). We try to merge each paragraph into adjoining paragraphs which have been already associated. That means, we assume that parts of paragraphs which are split are associated in the previous step.

We determine whether paragraphs should be merged by checking for the similarities, i.e., if similarities increased by merging, the paragraphs are merged. If similarity decreased, the paragraphs remain not associated.

We considered using document structures for merging. The merge process was used to deal with split paragraphs, and paragraphs would not split into different sections. Therefore, as we assumed that merging between paragraphs which were in different sections will not occur, we prevented such merging by decreasing similarities. We introduced a parameter, *penalty*, in our method, which was multiplied to the similarity of merged paragraphs of different sections. We can determine whether the merging is done over different sections by checking whether the parents of the paragraphs to be merged are the same. We could reduce false merging by doing this.

We applied the merging process described in this step to the results of the previous step. Figure 3 has an example of alignments in this step, where paragraphs which are associated with the same paragraph have been merged.

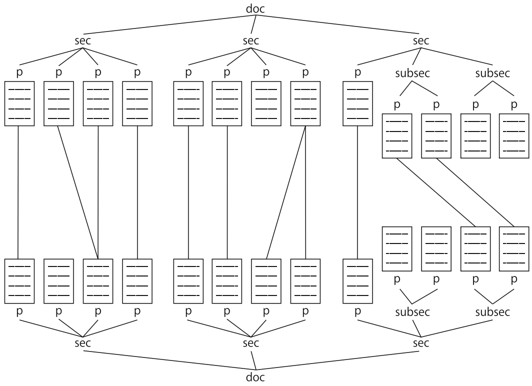

Fig. 3. Example of alignment in step 2

There are also paragraphs which have not been associated yet in this step. They were not merged since similarities decreased if they merged. If all paragraphs are associated in this step, the alignment process is finished.

3.3 Swapping

Alignments which took into consideration associations in multiple paragraphs were achieved in the previous steps. However, there are cases where the locations of paragraphs have changed where we need to consider cross associations. We took cross associations into account in this step, i.e., where paragraphs were swapped.

Like the merging process, the targets for the swapping process are paragraphs which have not been associated yet in the merging step (Section 3.2). Swapped paragraphs can remain not being associated in the previous steps. The targets in the swapping process are paragraphs in a document which are not associated and we try to find corresponding paragraphs in another document which are also not associated.

We need to check for similarities to find swapped corresponding paragraphs. For example, if similarities are greater than a threshold, then paragraphs become associated. However, it is hard to find corresponding paragraphs which are heavily modified only using the information from content as in the previous steps. Furthermore, as the swapping process does not take into consideration the order of paragraphs unlike in the previous steps, false associations can be increased. We need to set the threshold for associations in the swapping process to a greater value. Therefore, there need to be greater similarity of swapped paragraphs to be associated appropriately.

We then introduce the use of document structures into the swapping process. Before finding corresponding paragraphs, we enumerate subtrees which are composed only of the paragraphs that are not associated. We then try to swap the subtrees. Figure 4 shows an example of targets of swapping, where the subtrees enclosed by the dotted lines are the targets. We check for similarities for each

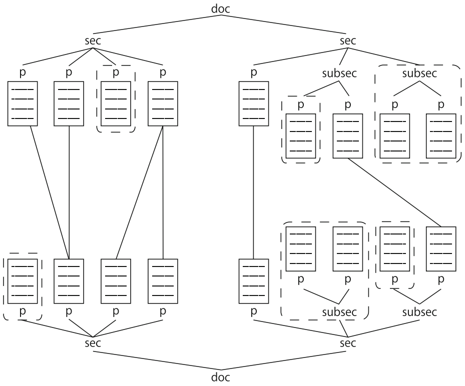

Fig. 4. Example of targets of swapping

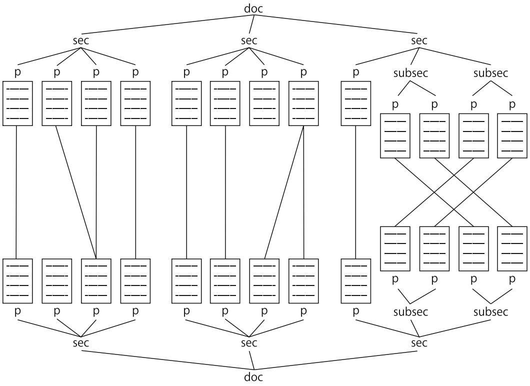

Fig. 5. Example of alignment in step 3

subtree in one document with subtrees in another document and if the similarities are greater than the threshold, then we regard the subtree as corresponding. Thus, we can find corresponding paragraphs by using the structural information in subtrees even when it is hard to swap paragraphs by content information only.

We use the Tree Edit Distance (TED) [13] to calculate the similarities between subtrees. We can find appropriate corresponding subtrees by using the structural information in subtrees even if there are few similarities by using the content of paragraphs. We choose a subtree which has the largest similarity value greater than the threshold as the corresponding subtree.

We focus on the two subtrees and apply our method from step 1 again if a corresponding subtree is found. The alignment process is recursively performed and stops when any of the following three states is achieved

1. All paragraphs are associated, or no paragraphs are associated in step 1.
2. All paragraphs are associated in step 2.
3. No paragraphs are associated in step 3.

Figure 5 shows an example of alignments in this step, where the crossed associations are swapped paragraphs. Paragraphs which are not associated could not find corresponding subtrees or paragraphs.

4 Experiments

4.1 Experimental Setup

We evaluated the proposed method using 8 pairs of papers which had been presented or published in different forms. The papers used in the experiment were stored in our laboratory as PDF files. We transformed the PDF files to XML files, which represented document trees as shown in Figure 1. Among the 8 pairs, 5 pairs are cross-lingual (papers in Japanese and English) and Japanese papers were translated into English by Google Translate as the preprocessing.

We manually judged proper alignments as the ground truth and calculated the precision and recall of the associations. We preliminarily examined optimal values for the parameters and set α in Eq. (2) to 0.425, C in Eq. (4) to 0.2, and *penalty* to 0.8.

We used two other approaches to evaluate our method.

- simple matching method
  It associates paragraphs and does not take their order into consideration. Our method consists of 3 steps, i.e., step 1: alignments using the Levenshtein distance, step 2: merging, and step 3: swapping. However, the merging and swapping steps are required since alignments using the Levenshtein distance are (1) associations in 1 to 1 and (2) these are not able to deal with changes in the locations of paragraphs. We use the Levenshtein distance in step 1 since we take into consideration the order of paragraphs and assume the alignments to be more appropriate.

  The simple matching method works as follows to confirm the effectiveness of considering the order of paragraphs. Each paragraph in one document is associated with a paragraph in another document which is the most similar to and its similarity is greater than threshold $1 - 2\alpha$. Thus, it identifies alignments and does not take into account the order of paragraphs.
- non-structural method
  The main idea of this research was that alignments would be improved by using document structures. We arranged this method, which is based on the proposed approach, and modified it in three ways to confirm this.
  1. It does not use sim_{global} in step 1 (alignments using the Levenshtein distance), i.e., C in Eq. (4) is set to 0.
  2. It does not impose a penalty on merging between different sections in step 2 (merge), i.e., the *penalty* is set to 1.
  3. It swaps using paragraphs in step 3 (swap), not subtrees.

  Thus, it makes alignments using only the content and the order information.

Table 1. Precision and recall for each method

	proposed method		non-structural method		simple matching method	
Pair ID	precision	recall	precision	recall	precision	recall
p1	0.95 (36/38)	0.95 (36/38)	0.91 (32/35)	0.84 (32/38)	0.62 (31/50)	0.82 (31/38)
p2	0.94 (34/36)	0.89 (34/38)	0.91 (30/34)	0.79 (30/38)	0.65 (24/37)	0.63 (24/38)
p3	0.85 (34/40)	0.76 (34/45)	0.80 (33/41)	0.73 (33/45)	0.68 (34/50)	0.76 (34/45)
p4	0.86 (44/51)	0.92 (44/48)	0.81 (38/47)	0.79 (38/48)	0.44 (35/80)	0.73 (35/48)
p5	0.95 (37/39)	0.90 (37/41)	0.93 (37/40)	0.91 (37/41)	0.69 (31/45)	0.76 (31/41)
p6	0.98 (51/52)	0.98 (51/52)	0.98 (51/52)	0.98 (51/52)	0.98 (51/52)	0.98 (51/52)
p7	1.0 (51/51)	0.93 (51/55)	0.98 (51/52)	0.93 (51/55)	0.88 (45/51)	0.82 (45/55)
p8	0.97 (37/38)	1.0 (37/37)	0.95 (37/39)	1.0 (37/37)	0.97 (31/32)	0.84 (31/37)

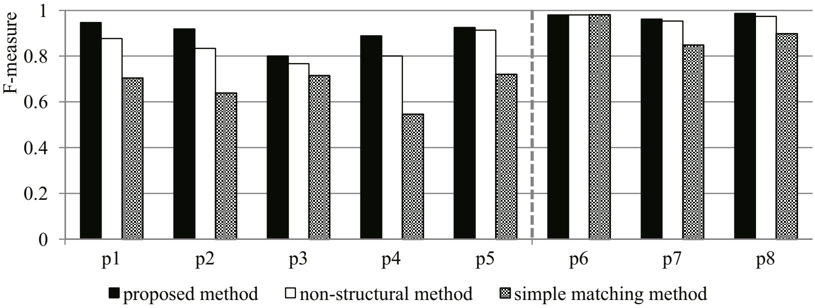

Fig. 6. F-measures for each method

4.2 Results

Table 1 lists the results of alignments for the methods. Precision and recall are calculated as follows.

- Number of correct associations in result / Number of associations in result
- Number of correct associations in result / Number of correct associations

Figure 6 shows the F-measures for the methods. The proposed method performed the best for each pair according to Figure 6.

There were many false alignments in the results for the simple matching method, which ignored the order of paragraphs. In contrast, the proposed method and non-structural method, which took into consideration the order of paragraphs, obtained natural alignments. These results confirm that taking into consideration the order of paragraphs is effective for alignments.

In contrast to the results of the non-structural method, the effectiveness of using document structures can be seen in the results of the proposed method. The proposed approach associated paragraphs which had not been associated by the non-structural method by using structural information. These results confirmed that taking document structures into consideration is effective for alignments.

Furthermore, the proposed method outperformed the other methods especially in the cross-lingual cases (p1, p2, p3, p4, and p5) on the left part of Figure 6. They were translated by machine translation in preprocessing. This means that using structures is especially effective for finding cross-lingual alignments.

5 Conclusion

We proposed a method of document alignment, in which we applied the algorithm of the Levenshtein distance and took into consideration document structures. The method also took into account the merging and swapping of paragraphs. We confirmed its effectiveness of using the structures of documents for alignments in our experiments. We will apply our method to more documents and improve it in future work.

References

1. Daumé III, H., Marcu, D.: A phrase-based HMM approach to document/abstract alignment. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, pp. 119–126 (July 2004)
2. Jeong, M., Titov, I.: Multi-document topic segmentation. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, pp. 1119–1128 (October 2010)
3. Romary, L., Bonhomme, P.: Parallel alignment of structured documents. In: Véronis, J. (ed.) Parallel Text Processing, pp. 233–253. Kluwer Academic Publishers (2000)
4. Zhang, H., Chow, T.W.S.: A multi-level matching method with hybrid similarity for document retrieval. Expert Systems with Applications 39(3), 2710–2719 (2012)
5. Zhang, H., Chow, T.W.S.: A coarse-to-fine framework to efficiently thwart plagiarism. Pattern Recognition 44(2), 471–487 (2011)
6. Wan, X.: A novel document similarity measure based on earth mover's distance. Information Sciences 177(18), 3718–3730 (2007)
7. Tekli, J., Chbeir, R.: A novel XML document structure comparison framework based-on sub-tree commonalities and label semantics. Journal of Web Semantics 11, 14–40 (2012)
8. Yahyaei, S., Bonzanini, M., Roelleke, T.: Cross-lingual text fragment alignment using divergence from randomness. In: Grossi, R., Sebastiani, F., Silvestri, F. (eds.) SPIRE 2011. LNCS, vol. 7024, pp. 14–25. Springer, Heidelberg (2011)
9. Au Yeung, C., Duh, K., Nagata, M.: Providing cross-lingual editing assistance to wikipedia editors. In: Gelbukh, A. (ed.) CICLing 2011, Part II. LNCS, vol. 6609, pp. 377–389. Springer, Heidelberg (2011)
10. Smith, J.R., Quirk, C., Toutanova, K.: Extracting parallel sentences from comparable corpora using document level alignment. In: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Los Angeles, USA, pp. 403–411 (June 2010)

11. Vu, T., Aw, A., Zhang, M.: Feature-based method for document alignment in comparable news corpora. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Athens, Greece, pp. 843–851 (2009)
12. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
13. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM Journal on Computing 18(6), 1245–1262 (1989)