

# Multi-Task Text Segmentation and Alignment Based on Weighted Mutual Information

Bingjun Sun\*, Ding Zhou\*, Hongyuan Zha\*, John Yen†

\*Department of Computer Science and Engineering,† College of Information Sciences and Technology  
The Pennsylvania State University, University Park, PA 16802

{bsun,dzhou,zha}@cse.psu.edu, jyen@ist.psu.edu

## ABSTRACT

Text segmentation is important for text analysis, while text alignment is to determine shared sub-topics among similar documents. Multi-task text segmentation and alignment is the extension of single-task segmentation to utilize information of multi-source documents. In this paper we introduce a novel domain-independent unsupervised method for multi-task segmentation and alignment based on the idea that the optimal segmentation and alignment maximizes weighted mutual information, mutual information with term weights. The experiment results show that our approach works well.

**Categories and Subject Descriptors:** H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval-*Clustering*; I.2.7 [Artificial Intelligence]: Natural Language Processing-*Text analysis*;

**General Terms:** Algorithms, Design, Experimentation.

**Keywords:** Multi-task, text segmentation, text alignment, weighted mutual information.

## 1. INTRODUCTION

Text segmentation tasks are to determine the boundaries of sentence sequences to capture the latent structures. Some previous approaches consider sentence dependence, such as *HMM* or *CRF*, while others are based on sentence similarity[3, 7, 8], which suffer the effect of stop words. Text classification and clustering is a related area which categorizes documents into groups, such as *LSA*[4], *PLSA*[6], and approaches based on mutual information (*MI*)[1, 5]. Traditional text segmentation approaches usually focused on single tasks. Multi-task learning[2] is an potential direction, but most of previous multi-task approaches focus on supervised or semi-supervised learning, instead of on clustering or segmentation. In this paper, we extend research from single-task to multi-task. We view the text segmentation issue as an optimization issue in information theory to find the optimal boundaries given the number of segments which minimize the loss of *MI* after segmentation. Text alignment of multi-source documents can be achieved by clustering sentences about the same sub-topic into the same segment. Term weights based on entropy learned from multi-source documents and weighted *MI* (*WMI*) is used to increase the contribution of cue words and decrease the effect of common stop words, noisy word, and document-

dependent stop words, which are removed before segmentation in methods based on sentence similarity.

## 2. PROBLEM FORMULATION

Let  $T$  be the term set  $\{t_1, t_2, \dots, t_l\}$ , appearing in the document set  $D$ ,  $\{d_1, d_2, \dots, d_m\}$ . Let  $S_d$ ,  $\{s_1, s_2, \dots, s_{n_d}\}$  be the sentence set for  $d \in D$ . The probability distribution  $P(D, S_d, T)$  is estimated as  $p(t, d, s) = T(t, d, s)/N_D$ , where  $T(t, d, s)$  is the number of  $t$  in  $d$ 's sentence  $s$  and  $N_D$  is the total term frequency in  $D$ .  $\hat{S}$  represents the segment set  $\{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_p\}$  after segmentation where the segment number  $|\hat{S}| = p$ . The multi-task text segmentation and alignment with term co-clustering is to find the optimal term clustering mapping  $Clu(t) : \{t_1, t_2, \dots, t_l\} \rightarrow \{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_k\}$ , where  $k \leq l$  is the number of clusters, and the optimal segmentation and alignment mapping  $Seg_d(s_i) : \{s_1, s_2, \dots, s_{n_d}\} \rightarrow \{\hat{s}'_1, \hat{s}'_2, \dots, \hat{s}'_p\}$  and  $Ali_d(\hat{s}'_i) : \{\hat{s}'_1, \hat{s}'_2, \dots, \hat{s}'_p\} \rightarrow \{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_p\}$ ,  $\forall d \in D$ , with the constraint that only adjacent sentences can map to the same segment. *MI* can measure the amount of information in several random variables[5]. In the segmentation task, our goal is to find the best solution to maximize  $I(\hat{T}; \hat{S}) = \sum_{\hat{t} \in \hat{T}} \sum_{\hat{s} \in \hat{S}} p(\hat{t}, \hat{s}) \log \frac{p(\hat{t}, \hat{s})}{p(\hat{t})p(\hat{s})}$  after segmentation and alignment. Similar as *tf-idf* weight, we define weights for four types of terms. *Common stop words* are common both along  $D$  and  $\hat{S}$ . *Document-dependent stop words* are common only along  $\hat{S}$  for some  $d$ . *Cue words* which are common along  $D$  only for some  $\hat{s}$ . *Noisy words* are others. To reinforce the contribution of *Cue words*, we introduce term weight  $w_{\hat{t}} = \left(\frac{E_D(\hat{t})}{\max_{\hat{t}' \in \hat{T}}(E_D(\hat{t}'))}\right)^a \left(1 - \frac{E_{\hat{S}}(\hat{t})}{\max_{\hat{t}' \in \hat{T}}(E_{\hat{S}}(\hat{t}'))}\right)^b$ , where  $E_D(\hat{t}) = \sum_{d \in D} p(d|\hat{t}) \log_{|D|} \frac{1}{p(d|\hat{t})}$ , similar for  $E_{\hat{S}}(\hat{t})$ , and usually  $a = b = 1$  to adjust  $p_w(\hat{t}, \hat{s}) = \frac{w_{\hat{t}} p(\hat{t}, \hat{s})}{\sum_{\hat{t}' \in \hat{T}; \hat{s} \in \hat{S}} w_{\hat{t}'} p(\hat{t}', \hat{s})}$ , and  $I_w(\hat{T}; \hat{S}) = \sum_{\hat{t} \in \hat{T}} \sum_{\hat{s} \in \hat{S}} p_w(\hat{t}, \hat{s}) \log \frac{p_w(\hat{t}, \hat{s})}{p_w(\hat{t})p_w(\hat{s})}$ .

## 3. METHODOLOGY

Since term weights depending on text segmentation and alignment are unknown and the problem is NP-hard, an iterative greedy algorithm is proposed to find a local maximum with simultaneous weight estimation. It can find the global optimum for single tasks without term co-clustering. For initialization,  $w_t = \left(\frac{E_D(t)}{\max_{t' \in T}(E_D(t'))}\right) \left(1 - \frac{E_S(t)}{\max_{t' \in T}(E_S(t'))}\right)$ , where  $E_S(t) = \frac{1}{|D_t|} \sum_{d \in D_t} \left(1 - \sum_{s \in S_d} p(s|t) \log_{|S_d|} \frac{1}{p(s|t)}\right)$ , where  $D_t$  is the set of  $d$  which contain  $t$ . Then, for  $Seg_d^{(0)}$ , we can simply segment documents equally, or we can find the optimal segmentation just for each  $d$  so that  $Seg_d^{(0)} =$

**Table 1: Error Rates of Single-task Segmentation**

Range of $n$	3-11	3-5	6-8	9-11
C99[8]	12%	11%	10%	9%
U00[3]	10%	9%	7%	5%
ADDP03[7]	6.0%	6.8%	5.2%	4.3%
$MI_l$	<b>4.68%</b>	<b>5.57%</b>	<b>2.59%</b>	<b>1.59%</b>
$WMI_l$	<b>4.94%</b>	<b>6.33%</b>	<b>2.76%</b>	<b>1.62%</b>

$\operatorname{argmax}_s I_w(T; \hat{S})$ , where  $w_i^{(0)}$  are used. For  $Ali_d^{(0)}$ , we can first assume that the segment order for each  $d$  is the same. For  $Clu_t^{(0)}$ , cluster labels can be set randomly. Then there are three stages, where the first is for single tasks without term clustering, while the other two are both iterative. Stage 2 is for term clustering, while Stage 3 is for term weight estimation. The algorithm is listed below:

**Input:**  $P(D, S_d, T)$ ,  $p \in \{2, \dots, \max(s_d)\}$ ,  $k \in \{2, \dots, l\}$ ,  $w \in \{0, 1\}$ . **Output:**  $Clu, Seg, Ali, w_i$ . (1)  $i = 0$ . Initialize  $Clu_t^{(0)}, Seg_d^{(0)}, Clu_d^{(0)}$ , and  $w_i^{(0)}$ ; (2) If  $|D| = 1$ ,  $k = l$ , and  $w = 0$ , check all segmentations of  $d$  and find the best  $Seg_d = \operatorname{argmax}_s I(\hat{T}; \hat{S})$ , return; (3) If  $k \neq l$ ,  $\forall t$ , find the best  $\hat{t}$  so  $Clu_t^{(i+1)} = \operatorname{argmax}_t I_w(\hat{T}; \hat{S}^{(i)})$  based on  $Seg^{(i)}$  and  $Ali^{(i)}$ ; (4)  $\forall d$ , check all segmentations of  $d$  with mapping  $s_i \rightarrow \hat{s}$ ,  $i \in 1, \dots, n_d$  and find the best  $Seg_d^{(i+1)} \& Ali_d^{(i+1)} = \operatorname{argmax}_s I_w(\hat{T}^{(i+1)}; \hat{S})$  based on  $Clu_t^{(i+1)}$ ; (5) If  $Clu, Seg$ , or  $Ali$  changed,  $i++$ , go to 3; otherwise, if  $w = 2$ , go to 6, else return. (6) Update  $w_i^{(i+1)}$  based on  $Seg^{(i)}, Ali^{(i)}, Clu$ ; (7)  $\forall d$ , check all segmentations of  $d$  with mapping  $s_i \rightarrow \hat{s}$ ,  $i \in 1, \dots, n_d$  and find the best  $Seg_d^{(i+1)} \& Ali_d^{(i+1)} = \operatorname{argmax}_s I_w(\hat{T}^{(i+1)}; \hat{S})$  based on  $Clu$  and  $w_i^{(i)}$ ; (8) If  $I_w(\hat{T}; \hat{S})$  not changed, return; else,  $i++$ , go to 6.

Dynamic programming is used for each step. We only show the steps for Step 7 below:  $\forall d$ : (1) Compute  $p_w(\hat{t})$ , partial  $p_w(\hat{t}, \hat{s})$  and  $p_w(\hat{s})$ , and  $PI_w(\hat{T}; \hat{s}_k(s_i, s_{i+1}, \dots, s_j))$ . (2) Let  $M(s_m, 1, k) = PI_w(\hat{T}; \hat{s}_k(s_1, s_2, \dots, s_m))$ , where  $k \in \{1, 2, \dots, p\}$ .  $M(s_m, L, k_L) = \max_{i,j} [M(s_{i-1}, L-1, k_{L/j}) + PI_w(\hat{T}; \hat{s}_{Ali_d(s'_L)=j}(s_i, s_{i+1}, \dots, s_m))]$ , where  $0 \leq m \leq n_d$ ,  $1 < L < p$ ,  $1 \leq i \leq m+1$ ,  $k_L \in \text{Set}(p, L)$ , which is the set of all  $\frac{p!}{L!(p-L)!}$  combinations of  $L$  segments chosen from all  $p$  segments,  $j \in k_L$ , the set of  $L$  segments chosen from all  $p$  segments, and  $k_{L/j}$  is the combination of  $L-1$  segments in  $k_L$  except  $j$ . (3) Finally,  $M(s_{n_d}, p, k_p) = \max_{i,j} [M(s_{i-1}, p-1, k_{p/j}) + PI_w(\hat{T}; \hat{s}_{Ali_d(s'_L)=j}(s_i, s_{i+1}, \dots, s_{n_d}))]$ , where  $k_p$  is combination of all segments and  $1 \leq i \leq n_d+1$  which is the optimal  $I_w$  and the corresponding segmentation is the best.

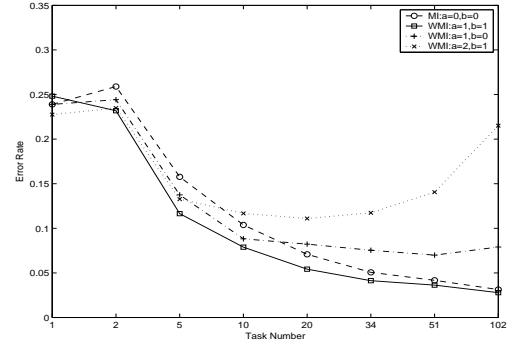
## 4. EXPERIMENTS

In this section, we refer to the method using  $I$  as  $MI_k$ , and  $I_w$  as  $WMI_k$ , where  $k$  is the number of term clusters. If  $k = l$ , no term clustering is required. The first data set is used in previous research. We use the previous evaluation criterion for comparison and tested the case with the known segment number. Table 1 shows the results with different parameters and previous approaches. For single-task  $WMI_l$ , term weights are computed as:  $w_i = 1 - \frac{E_{\hat{S}}(\hat{t})}{\max_{\hat{t}' \in \hat{T}} (E_{\hat{S}}(\hat{t}'))}$ . Obviously, our methods  $MI_l$  and  $WMI_l$  both outperform the previous approaches. We found using term co-clustering is worse.

The data set for multi-task has 102 samples and 2264 sentences totally. Each is the introduction of a report from

**Table 2: Error Rates of Multi-task Segmentation**

#Task	$MI_l$	$WMI_l$	$k$	$MI_k$	$WMI_k$
102	3.14%	<b>2.78%</b>	300	4.68%	6.58%
51	4.17%	<b>3.63%</b>	300	17.83%	22.84%
34	5.06%	<b>4.12%</b>	300	18.75%	20.95%
20	7.08%	<b>5.42%</b>	250	20.40%	21.83%
10	10.38%	<b>7.89%</b>	250	21.42%	21.91%
5	15.77%	<b>11.64%</b>	250	21.89%	22.59%
2	25.90%	<b>23.18%</b>	50	25.44%	25.49%


**Figure 1: Error rates for different hyper parameters of term weights w/o term clustering.**

Biol 240W, Penn. State Univ. Each has two segments. Some only have one segment or have a reverse order. We labelled each sentence manually for evaluation. The criterion is:  $p(\text{err}|\text{pred}, \text{real}) = \sum_{d \in D, s \in S_d} 1_{(\text{pred}_s \neq \text{real}_s)} / \sum_{d \in D} n_d$ .

We compared our method with different parameters on different partitions of the data set. Except the cases that the task number is 102 or one, we randomly divided the set into partitions, each with 51, ..., or 2 samples. Then we applied our methods. Results are shown in Table 2, we can see that when the task number increases, all methods are better.  $WMI_l$  is always better than  $MI_l$ . Using term clustering is worse. We also tested  $WMI_l$  with different parameters of  $a$  and  $b$ , shown in Figure 1, and  $a = 1, b = 1$  gave the best results.

## 5. REFERENCES

- [1] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *Proc. ICML*, 2005.
- [2] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.
- [3] F. Choi. Advances in domain independent linear text segmentation. In *Proc. NAACL*, pages 26–33, 2000.
- [4] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Systems*, 1990.
- [5] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. In *Proc. SIGKDD*, pages 89–98, 2003.
- [6] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. UAI*, 1999.
- [7] X. Ji and H. Zha. Domain-independent text segmentation using anisotropic diffusion and dynamic programming. In *Proc. SIGIR*, pages 322–329, 2003.
- [8] M. Utiyama and H. Isahara. A statistical model for domain-independent text segmentation. In *Proc. ACL*, pages 491–498, 1999.