International Journal of Smart Computing and Artificial Intelligence
International Institute of Applied Informatics
2023, Vol. 7, No. 2, IJSCAI701

Improving Abstractive Summarization by Transfer Learning with Adaptive Document Selection

Masato Shirai [*] , Kei Wakabayashi [†]

Abstract

Abstractive document summarization based on neural networks is a promising approach to generate a flexible summary but requires a large amount of training data. While transfer learning can address this issue, there is a potential concern about the negative transfer effect that deteriorates the performance when we use training documents irrelevant to the target domain, which has not been explicitly explored in document summarization tasks. In this paper, we propose a method that selects training documents from the source domain that are expected to be useful for the target summarization. The proposed method is based on the similarity of word distributions between each source document and a set of target documents. We further propose an adaptive approach that builds a custom-made summarization model for each test document by selecting source documents similar to the test document. In the experiment, we confirmed that the negative transfer actually happens also in the document summarization tasks. Additionally, we show that the proposed method effectively avoids the negative transfer issue and improves summarization performance.

Keywords: Abstractive Summarization, Transfer learning, Negative Transfer

1 Introduction

Document summarization methods fall into two types: extractive summarization and abstractive summarization. The extractive summarization methods select sentences as a summary from a given document. The abstractive summarization methods generate summaries based on the pattern of relationships between input text and output summary by analyzing training pairs. Although abstractive summarization methods potentially can generate a flexible summary because it is allowed to generate new sentences, it requires a large amount of training data to generate an appropriate summary. The training data preferably should be documents in the domain that is the same as the domain the target document belongs to; however, it is not easy to create a large training set in the target domain.

When sufficient amount of training documents cannot be obtained on the target domain, transfer learning is a promising approach to compensate for the lack of training data. Transfer learning exploits training documents on a source domain, which is different from

[*] Shimane University, Shimane, Japan
[†] University of Tsukuba, Ibaraki, Japan

the target domain but contains a lot of training instances, to boost the model performance on the target domain. It is anticipated that the transfer learning is helpful also in the document summarization task for improving the quality of the summary to be generated. However, it is also known that a negative transfer phenomenon possibly happens, in which the performance is rather deteriorated than the case we only use the small number of training instances on the target domain, because of a transfer of irrelevant information [1]. The negative transfer generally happens when the relevance between the two domains is poor. Therefore, selecting relevant training instances in the source domain is crutial to obtain a good performance.

In this study, we consider inductive transfer learning that assumes a small set of training data is available in the target domain and a large one is available in the source domain. We address the following three research questions on inductive transfer learning in the document summarization task.

- Does the negative transfer truly happen in the document summarization task?
- Can we improve the performance of the summarization by selecting source documents that are similar to the target documents?
- How much can the summarization performance be improved by using transfer learning?

In this paper, we propose a method for selecting source documents that contribute to the improvement of the document summarization model on the target domain. The proposed method selects the source documents based on the similarity of the probability distributions of words that compares each source document and the set of target documents. We further propose an adaptive approach that builds a custom-made summarization model for each test document by selecting source documents similar to the test document. In the experiment, we confirm that the proposed document selection methods improve the summarization performance compared to standard transfer learning strategies. In addition, we show that the negative transfer happens and the performance of the summary generation degrades when we use dissimilar source documents for transfer learning.

2 Related Work

Inductive transfer learning is a kind of transfer learning in which training data exists in both the source domain and the target domain. In the natural language processing field, transfer learning methods that adopt model parameters to different word distributions have been proposed [2, 3, 4] to overcome the mismatching of word distribution that depends on the document topic. As a previous study of inductive transfer learning, Rai et al. [5] have proposed a method for training a model using the source domain and target domains. In this problem formulation, the distribution of source domain requires to be the same as the target domain, and the case of the dissimilar distributions between domains is not considered. Zhu et al. [6] proposed a method to dynamically update importance weight of the knowledge obtained from source domain. This method directly uses the data in the source domain as the training data for the target domain. In these methods, erroneous information is potentially propagated when the relevance of each domain is low so that the accuracy may deteriorate. Chattopadhyay et al. [7] have proposed a framework for conducting transfer learning and active learning for adopting domain simultaneously by solving a single convex optimization problem. This method can handle situations where labeled data is not available in the target domain at initial moment. However, there is a disadvantage that the computational cost for active learning is enormous because training data is selected one by one from the source domain.

In recent years, transfer learning has been successfully applied to document classification and named entity extraction [8, 9]. These studies show that transfer learning improves accuracy, but suggests that negative transfer potentially occurs. Tan et al. [10] have proposed a method for avoiding negative transfer by passing through the intermediate domain connecting the source domain and the target domain. For text summarization, Keneshloo et al. [11] have proposed a method for leveraging a dataset other than the target domain as training data. This method shows that summarization accuracy can be improved by using the training data of the source domain for training the model of the target domain. However, they did not examine the effects of the dissimilarity between the domains. Therefore, the possibilities and effects of negative transfer in document summarization are still unclear.

3 Transfer Learning for Summarization based on Source Document Selection

In this study, we use the pointer-generator networks as a generative document summarization model [12]. For training the model in an inductive transfer learning setting, we propose a method that selects documents from the source domain and uses them for training along with the training data in the target domain. In the document selection process, we choose documents that are similar to the training data in the target domain in terms of the KL divergence of the word distribution.

3.1 Problem Setting

We aim at constructing a document summarization model for the target domain using transfer learning. We have three types of documents: a large set of documents in the source domain $S = \{s_1, s_2, ..., s_k\}$ along with summary, a small number of training documents in the target domain $L = \{l_1, l_2, ..., l_n\}$ along with summary, and the test documents in the target domain $U = \{u_1, u_2, ..., u_m\}$. A naive way of transfer learning is to train a document summarization model by using all available documents L and S together. However, we hypothesize that a negative transfer potentially happens if we use all the documents in S because it may include documents that are considerably dissimilar to L and U. In order to examine (and overcome) the negative transfer effect, we consider a process that selects the source documents to be used for training.

3.2 Document Summarization using Pointer-Generator Networks

The pointer-generator networks is an abstractive summarization model with an attention distribution that represents the parts of interest and a copying mechanism that determines the parts to be copied from the text [12]. The pointer-generator networks are suitable for transfer learning with the target domain and the source domain, because of the copying mechanism that can generate a summary containing out-of-vocabulary words[11].

Let $D = \{d_1, ..., d_N\}$ be a set of documents, $d_i = \{x_1, ..., x_v\}$ be the ith document, and x_j is the one-hot vector of dimension of the vocabulary size. Each encoder receives the embedding of the word x as input and generates the output state h_x. The decoder generates the corresponding output by obtaining h_{x_v}, which is the final state of the sentence from the encoder. The attention vector α_j, the context vector c_j, and the output distribution p_{vocab} are computed by the following formula;

$$\begin{aligned}
e_{ij} &= v^T tanh(W_h h_i + W_s s_j + b_{attn}) && (1)\\
\alpha_j &= softmax(e_j) && (2)\\
c_j &= \sum_i^{T_w} \alpha_{ij} h_i && (3)\\
p_{vocab} &= softmax(V'(V[s_j \oplus c_j + b]) + b') && (4)
\end{aligned}$$

where v, b, W_h, and W_s are the parameters of the model to be trained. In the seq2seq model with attention distribution, cross entropy loss is calculated using p_{vocab}. Since p_{vocab} deals only with the distribution of words in the vocabulary, out-of-vocabulary words cannot be considered, but the pointer-generator networks mitigate this problem by switching word choices from the vocabulary or the target document with a probability of $(1 - p_{gen})$.

$$p_{gen} = \sigma(W_c c_j + W_s s_j + W_x x_j + b_{ptr}) \tag{5}$$

$$p^*_j = p_{gen} p_{vocab} + (1 - p_{gen} \sum_{i=1}^{T_e} \alpha_{ij}) \tag{6}$$

W_c, W_x, and b are parameters of the model to be trained. σ is a sigmoid function. If the word x_j is an out-of-vocabulary word, then $p_{vocab} = 0$ and the model selects the appropriate word from the target document based on the attention distribution. The final cross-entropy loss is calculated from the following formula.

$$Ł_{CE} = -\sum_{t=1}^{T} log p^*_\theta(y_t | e(y_{t-1}), s_t, c_{t-1}, \mathbf{X}) \tag{7}$$

θ is a set of trainable parameters in the model, and $e(.)$ represents a specific word embedding.

3.3 Document Selection based on Word Distribution

We use KL divergence to select the documents used for the training of the pointer-generator networks. First, we calculate the probability distribution of word P from all the training data in the target domain L. The probability of occurrence of each word in L is calculated by using the following equation.

$$P_L(w|L) = \frac{n_{w,L} + \alpha}{\sum_w^V n_{w,L} + V\alpha} \tag{8}$$

where $n_{w,L}$ is the number of occurrences of the word w in the document set L, V is the vocabulary size, and α is a smoothing parameter. We select a document in the source domain that has a similar word distribution to P_L. The probability distribution of words is calculated for each document in the source domain $s \in S$ and compared with the probability distribution P_L of the target domain. We use the KL divergence for comparing the probability distributions. KL divergence is a measure of the difference between probability

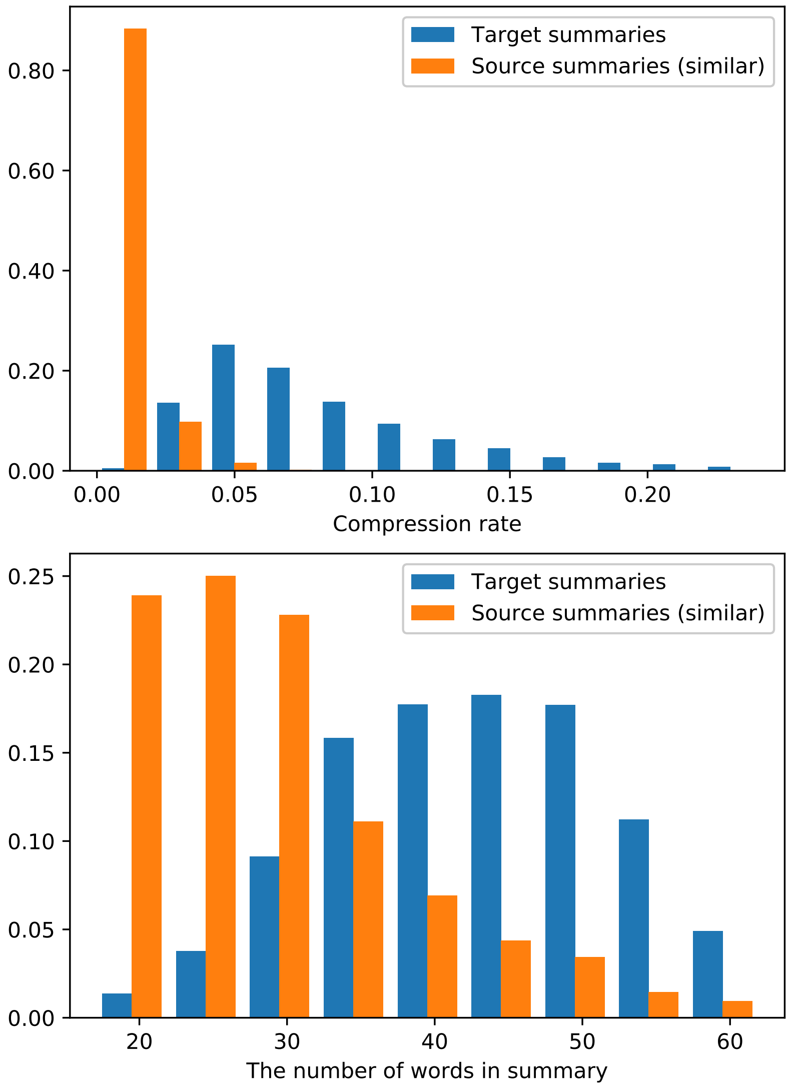

Figure 1: Distributions of compression rate (**Top**) and the number of words in summary (**Bottom**).

distributions. Given a probability distribution P_s and a probability distribution P_L, the KL divergence is calculated by using the following formula.

$$KL(P_s//P_L) = \sum_i P_{s,i} log \frac{P_{s,i}}{P_{L,i}} \tag{9}$$

The value of the KL divergence increases as the difference between the two probability distributions increases. When P_s and P_L are equal, the value of the KL divergence is 0. In the proposed method, the top N documents that have the lowest KL divergence are used for training.

3.4 Document Selection based on Length of Summary

Another characterization of training data is the length of summary. In general, the tendency of the length of the summary can be different between the target domain and the source domain. Figure 1 shows the distributions of the compression rate and the number of words that are calculated on the datasets we present in the experiment section. The figure indicates that the distributions calculated by using the training documents in the target domain L and the most similar (in terms of the word distribution) 10,000 source documents are considerably different. Since the summarization models are expected to learn the patterns of document-to-summary transformation in the training data, the difference of the summary length may cause a negative transfer.

To examine whether the difference of the length of summary between the target domain and the source domain causes negative transfer, we apply a process that selects source documents so that the distribution becomes the same as the target documents. We consider two features to characterize a training document, called length features; (i) the number of words in the summary, and (ii) the compression rate that means the ratio of the length of the summary and the input text.

First, we make a histogram of the length feature for the target documents L with a specific range of bins. The range of each bin is denoted by $\{f_1, f_2, ..., f_k\}$. We calculate the probability that the length feature of a document in L falls into a range f_i, denoted by $r^L_{f_i}$.

In the document selection of the source domain, we decide the number of source documents to be selected based on $r^L_{f_i}$. Let S_{f_i} be a set of source documents that have a length feature that is in f_i. When we intend to choose N source documents, we choose the most similar $Nr^L_{f_i}$ source documents from S_{f_i}. This process yields a set of documents $S' \subset S$ that have the same length feature distribution, i.e., $r^{S'}_{f_i} \simeq r^L_{f_i}$ for all f_i.

3.5 Document Selection depending on each test document

Document selection methods presented in the previous sections filter the source documents S once depending on the word distribution calculated using the whole set of documents in the target domain. However, this selection process possibly drops source documents having important words for a specific test document. For example, documents about Japanese culture might be excluded from the selection process because it is a minor topic, even though they will be helpful for summarizing test documents on the same topic. As motivated by this consideration, we propose a method that builds a custom-made summarization model for each test document by adaptively selecting source documents depending on the word distribution of the test document.

In advance, we train a pointer-generator network using both the training documents in the target domain L and the training documents selected from S by a method proposed in the previous sections. We call the trained network a base summarization model. Recall that we assume a set of test documents to be summarized $U = \{u_1, u_2, ..., u_m\}$ in the target domain. For each test document $u \in U$, the proposed method trains a custom-made summarization model by the following steps:

1. N most similar source documents are selected in terms of the KL divergence $KL(P_s//P_u)$, where P_u is the word distribution of u calculated in the same way as Eq. 8. We denote the selected source documents by $S_u \subset S$ ($|S_u| = N$).

2. Additional training epochs are applied to the base summarization model using the training documents in the target domain L and the selected source documents S_u. The updated model is called a custom-made summarization model.

Table 1: Average number of words in each domain

	Average number of words in summary	Average number of words in document	Average compression rate
Training data of target domain	41.9	653	0.0847
Test data of target domain	41.9	653	0.0850
Source domain	27.1	673	0.0756

In this way, the custom-made summarization model can exploit knowledge from the source documents having words in u, which are possibly excluded from the selected documents for training the base summarization model.

4 Experiments

In the experiment, we use the CNN dataset [13] as the target domain and the CORNELL NEWSROOM dataset [14] as the source domain. In order to examine the effect of selection of the source documents, we compare the ROUGE scores with different selection strategies as we explain in section 4.3.

4.1 Experiment preparation

The CNN dataset contains the text of CNN news articles and their summaries. We chose 20% of the CNN dataset (18,516 documents) as the training data in the target domain and another 20% of the CNN dataset as the test data. The CORNELL NEWSROOM dataset is a large dataset containing 1.3 million pairs of the article and its summary written by authors and editors from 38 publications. We use the 995,040 documents in the training set of the CORNELL NEWSROOM dataset as the source documents. All words are converted to lowercase as a pre-processing of the document. The KL divergence is calculated using words that appear more than 4 times in the experimental data. The number of iterations of training for the pointer-generator networks is 5000.

The statistics about the length of the documents and summaries in the dataset are shown in the Table 1. The compression rate is the number of words in the summary sentence divided by the number of words in the body text. The statistics of the training data and test data in the target domain are almost the same. When comparing the data of the source domain and the target domain, the average number of words included in the documents is almost the same, but the summary of the source domain tends to be shorter.

In the document selection based on compression rate, match the ratio of document of the training data of the source domain and the target domain in each section of the compression rate 0 to 0.24 that includes 99% of the training data of the target domain. The length of each section is 0.02. In the document selection based on the number of words in the summary, match the ratio of the number of words in the summary of the training data of the source domain and the target domain in each section of the number of the words 17 to 62 that includes 99% of the training data of the target domain. The length of each section is 5.

4.2 Evaluation methods

4.2.1 ROUGE score

We use the ROUGE score [15] to evaluate experiments. The ROUGE score is a measure used for evaluating document summaries and machine translations, and expresses the accuracy of summarization by comparing the correct answer of the summaries with the generated summaries. ROUGE-N evaluates the word match in the N-gram, and ROUGE-L evaluates the longest match. ROUGE-N and ROUGE-L scores are calculated by the following formulas.

$$R\text{-}N_R = \frac{C_{match}(summary_{words}, reference_{words})}{C(reference_{words})} \tag{10}$$

$$R\text{-}N_P = \frac{C_{match}(summary_{words}, reference_{words})}{C(summary_{words})} \tag{11}$$

$$R\text{-}N_F = \frac{1}{N}\sum_{i}^{N} \frac{2 * R\text{-}N_{R_i} * R\text{-}N_{P_i}}{R\text{-}N_{R_i} + R\text{-}N_{P_i}} \tag{12}$$

$$R\text{-}L_R = \frac{LSC(summary_{words}, reference_{words})}{C(reference_{words})} \tag{13}$$

$$R\text{-}L_P = \frac{LSC(summary_{words}, reference_{words})}{C(summary_{words})} \tag{14}$$

$$R\text{-}L_F = \frac{1}{N}\sum_{i}^{N} \frac{2 * R\text{-}L_{R_i} * R\text{-}L_{P_i}}{R\text{-}L_{R_i} + ROUGE\text{-}L_{P_i}} \tag{15}$$

Recall indicates how well the true summary could be reproduced, and Precision indicates how well the generated summary is included in the true summary. The f-value is the harmonic average of Recall and Precision. In this study, we use 1-gram and 2-gram as N-gram.

4.2.2 Human evaluation using crowdsourcing

We conduct an human evaluation of summaries by using crowdsourcing on Amazon Mechanical Turk. We randomly select 1,000 documents from the test data for the evaluation. Three workers (reviewers) are assigned to evaluate each summary. In the evaluation task, the original document and multiple summaries are displayed to the workers, and they answer questions "*How much you agree with the statements: The summary adequately expresses the important points of the original text.*". Reviewers read the text summary and rate whether the summary represents an important point in the original text, with a score of 1-5 points (higher is better). We calculate the average score for each method.

4.3 Methods in Comparison

We compare the performance of the pointer-generator summarization with the following conditions.

- T+S(Similar10K): 10,000 documents with the highest similarities in the source domain are used as training data in addition to the training data in the target domain.

Table 2: ROUGE-1

	$ROUGE\text{-}1_{recall}$	$ROUGE\text{-}1_{precision}$	$ROUGE\text{-}1_{fvalue}$
T+S(Similar10K)	**0.377**	0.301	0.325
T+S(Dissimilar10K)	0.213	0.267	0.233
T+S(Random10K)	0.329	0.316	0.315
T+S(All995K)	0.306	0.301	0.297
T only	0.275	**0.324**	0.292
S(Similar10K) only	0.345	0.309	0.317
S(Dissimilar10K) only	0.057	0.058	0.054
T+S(CR-Similar10K)	0.272	0.319	0.288
T+S(CW-Similar10K)	0.277	0.315	0.290
T+S(Adaptive10k)	0.367	0.323	**0.335**

Table 3: ROUGE-2

	$ROUGE\text{-}2_{recall}$	$ROUGE\text{-}2_{precision}$	$ROUGE\text{-}2_{fvalue}$
T+S(Similar10K)	**0.137**	0.106	0.116
T+S(Dissimilar10K)	0.043	0.054	0.047
T+S(Random10K)	0.113	0.107	0.107
T+S(All995K)	0.104	0.101	0.100
T only	0.074	0.087	0.079
S(Similar10K) only	0.120	0.106	0.109
S(Dissimilar10K) only	0.003	0.003	0.003
T+S(CR-Similar10K)	0.065	0.076	0.069
T+S(CW-Similar10K)	0.069	0.078	0.072
T+S(Adaptive10k)	0.130	**0.113**	**0.118**

- T+S (Dissimilar10K): 10,000 documents with the lowest similarities in the source domain are used as training data in addition to the training data in the target domain.

- T+S (Random10K): 10,000 documents that randomly selected from the source domain are used as training data in addition to the training data in the target domain.

- T+S (All995K): all documents in the source domain are used as training data in addition to the training data in the target area.

- T only: the training data in the target domain is used for training.

- S (Similar10K) only: 10,000 documents with the highest similarities in the source domain are used as training data.

- S (Dissimilar10K) only: 10,000 documents with the lowest similarities in the source domain are used as training data.

- T+S (CR-Similar10K): 10,000 documents are chosen from the source domain to have the same compression rate distribution to the training data in the target domain. The selected documents are used as training data in addition to the training data in the target domain.

Table 4: ROUGE-L

	$ROUGE\text{-}L_{recall}$	$ROUGE\text{-}L_{precision}$	$ROUGE\text{-}L_{fvalue}$
T+S(Similar10K)	**0.335**	0.266	0.288
T+S(Dissimilar10K)	0.180	0.227	0.197
T+S(Random10K)	0.290	0.277	0.277
T+S(All995K)	0.269	0.264	0.261
T only	0.252	**0.297**	0.268
S(Similar10K) only	0.304	0.272	0.279
S(Dissimilar10K) only	0.055	0.056	0.052
T+S(CR-Similar10K)	0.247	0.289	0.261
T+S(CW-Similar10K)	0.250	0.285	0.261
T+S(Adaptive10k)	0.327	0.288	**0.298**

Table 5: Evaluation by crowdsourcing. The asterisk (*) indicates that the average score is statistically significant compared against any other methods ($p < 0.001$).

	Average score	Standard deviation
Reference	3.539*	0.915
T+S(Similar10K)	3.629*	0.922
T+S(Dissimilar10K)	3.204*	1.086
T only	3.294*	1.008

- T+S (CW-Similar10K): Same to T+S(CR-Similar10K) except for checking the distribution of the number of words in summary instead of the compression rate distribution.

- T+S (Adaptive10K): A base summarization model is trained by using T+S(Similar10K). In addition, the model is updated using the 10,000 documents with the highest similarity for each test data and the training data of the target domain as described in Section 3.5.

4.4 Results

Table 2 shows the ROUGE-1 scores of the summaries generated by the methods in comparison. From the Table 2, we can see the f-values by "T+S(Similar10K)" are 0.325. The highest f-value of ROUGE-1 (0.335) is obtained by the "T+S(Adaptive10K)" setting. In "T+S(Adaptive10K)", the f-values is improved by 1.0% by updating the model depending on each test document from the setting of 'T+S(Similar10K)". Since the f-values by "T only" and "T+S(All995K)" are 0.292 and 0.297, the additional training data in a naive setting that uses all the available documents does not to contribute significantly to improving the performance. The "T+S(Similar10K)" method improves the f-value by 3.3% from "T only" and 1.0% from "T+S(Random10K)". This result shows the effectiveness of the proposed method that selects similar documents based on the KL divergence.

In Table 2, the f-value by "T+S(Dissimilar10K)" are clearly degraded by 5.9% from the "T only" performance. This fact indicates that a negative transfer effect exists in the document summarization task when we transfer the knowledge from the documents in the

Table 6: Example of summaries. For this document, three crowd workers gave scores (4, 4, 5) for T+S(Similar10K), while they gave (4, 3, 3) for Reference. (For the workers, the display order of the summaries were randomized.)

Original text: (CNN) – Kanye West has high expectations from his fans at concerts. And if you can't meet them, he's going to need to know why. While performing in Sydney on Friday, West paused his performance to request that all audience members stand before he went on with the show. It's not unusual for an artist to ask the audience to physically participate, but West's insistence at his concert Friday led to a very uncomfortable moment. ”I've decided, I can't do this song, I can't do the rest of this show until everybody stands up,” West said, adding an exception for those with handicaps. ”No, seriously – I won't go on with the show unless y'all stand up.” As video from the concert shows, West really wasn't kidding, as he took several moments to scan the audience for anyone sitting down. When he spotted two people who still weren't standing, West initially griped, ”This was the longest I've had to wait to do a song. It's unbelievable.” According to The Hollywood Reporter, one of those individuals had a prosthetic limb, and another was in a wheelchair. After sending someone over to investigate why the two concertgoers weren't standing, West quickly clarified that ”if you're using a wheelchair, then it's fine.... Only if he's in a wheelchair.” Once he'd confirmed that those not standing really couldn't, West continued with his show. The stunt was met with an immediate backlash on Twitter. According to the Daily Mail, it wasn't the first time West has been criticized for making that kind of demand. Opinion: Kanye West and proving your disabilities
Reference: Kanye west has been criticized for insisting the audience stand up during a concert. The rapper was in australia when he made the demand. He held up the show until he 'd confirmed that everyone who could was standing.
T+S(Similar10K): Kanye west has high expectations from his fans at concerts. And if you can't meet them, he's going to need to. While performing in sydney on friday, west paused his performance to request that all audience members stand before with the show. It's not unusual for an artist to ask the audience to physically participate, but west's insistence at his concert friday led to a very uncomfortable moment.
T+S(Dissimilar10K): The insistence to the artist of this fans of concerts, west paused, the going, one, the the hollywood shows, the audience, all not unusual, as he took n't meet expectations.
T only: Kanye paused has high expectations from his artist at concerts. It's not unusual for an show backlash, but west said says. ”stunt at his concert friday led to a very wheelchair moment.
T+S(Adaptive10k): Kanye west has high expectations from his fans at concerts to know why. While performing in sydney on friday, west paused his performance to request that all audience members stand before he went on with the show.

source domain that have low similarity. This observation also demonstrates that the choice of the documents in the source domain is crucial to obtain a better performance in the target domain.

Tables 3 and 4 show the ROUGE-2 and ROUGE-L scores, respectively. The same tendencies we confirmed in the ROUGE-1 scores are observed in Table 3 and 4.

In order to investigate the effect of the length of summary of each domain, we compared the performance of the methods that selects the documents in the source domain considering the compression rate “T+S(CR-Similar10K)” and the number of words in the summary “T+S(CW-Similar10K)”. However, the performance is worse compared to “T only”. When we select 10,000 documents so that the compression rate has a similar distribution, the document having the lowest similarity among the 10,000 documents was the 145,055th document in the similarity ranking. For this reason, the proportion of documents with a high similarity, which is considered to contribute to performance improvement, is decreasing. As we show in Table 1, the Newsroom dataset considerably tends to have shorter sentences than the CNN dataset. However, the result indicates that the word similarity is more important than the length similarity in the transfer learning in document summarization task.

Table 5 shows the average scores of the human evaluation by crowdsourcing. “Ref-

Table 7: ROUGE-1 score when varying the number of documents selected from source domain in T+S(Similar10K) setting

	5000	10000	20000	30000
$ROUGE\text{-}1_{recall}$	0.287	0.377	0.359	0.345
$ROUGE\text{-}1_{precision}$	0.331	0.301	0.311	0.309
$ROUGE\text{-}1_{fvalue}$	0.302	0.325	0.325	0.317

erence" represents the summaries given in the CNN dataset as the ground truth. Av-erage score of "T+S(Similar10K)" is higher than "T+S(Dissimilar10K)" and "T only". "T +S(Dissimilar10K)" has a lower score than "T only". It indicates dissimilar sources do-main reduces the quality of summaries. Table 6 shows example of summaries. "T +S(Similar10K)" has a higher score than "Reference", because the summary is longer than the reference. "T+S(Dissimilar10K)" and "T only" have not been able to generate a proper summary. Table 7 shows the ROUGE-1 score when the number of documents selected from the source domain is varied from 5000 to 30000. The f-value of ROUGE-1 is 0.302 for "T+S(Similar5k)", 0.325 for "T+S (Similar10k)", 0.325 for "T+S (Similar20k)", and 0.317 for "T+S(Similar30k)". The f-value is highest when the number of documents selected from the information sources is 10,000 to 20,000. Since the number of training documents in the target domain is 18516, it is considered that the influence of the additional documents was limited in 5000 documents. In the 30000 documents, the number of training data of the source domain is far larger than that of the training data in the target domain, so it is considered that the f-value was not improved.

5 Conclusion

In this paper, we have proposed multiple document selection methods for transfer learning in the document summarization task to avoid the negative transfer issue. We have empir-ically shown that the negative transfer actually happens when the source documents are dissimilar to those in the target domain. Additionally, we have confirmed that the proposed document selection methods improve the ROUGE score compared to the standard way to use available training data in the transfer learning setting. The experimental results have also shown that building custom-made summarization models by using source documents selected adaptively depending on each test document improves the summarization perfor-mance.

References

[1] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *NIPS 2005 workshop on transfer learning*, vol. 898, 2005, pp. 1–4.

[2] X. Shi, W. Fan, and J. Ren, "Actively transfer domain knowledge," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 342–357.

[3] M. Shirai, J. Liu, and T. Miura, "Transfer learning using latent domain for document stream classification," in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*. IEEE, 2016, pp. 82–88.

[4] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang, "Dual transfer learning," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 540–551.

[5] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian, "Domain adaptation meets active learning," in *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 27–32.

[6] Z. Zhu, X. Zhu, Y. Ye, Y.-F. Guo, and X. Xue, "Transfer active learning," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 2169–2172.

[7] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Joint transfer and batch-mode active learning," in *International Conference on Machine Learning*, 2013, pp. 253–261.

[8] T. Semwal, P. Yenigalla, G. Mathur, and S. B. Nair, "A practitioners' guide to transfer learning for text classification using convolutional neural networks," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 513–521.

[9] B. Y. Lin and W. Lu, "Neural adaptation layers for cross-domain named entity recognition," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2012–2022.

[10] B. Tan, Y. Song, E. Zhong, and Q. Yang, "Transitive transfer learning," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1155–1164.

[11] Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, "Deep transfer reinforcement learning for text summarization," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 675–683.

[12] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1073–1083.

[13] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in neural information processing systems*, 2015, pp. 1693–1701.

[14] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2018, pp. 708–719.

[15] C.-Y. Lin and E. Hovy, “Manual and automatic evaluation of summaries,” in *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*. Association for Computational Linguistics, 2002, pp. 45–51.