

Text Length Adaptation in Sentiment Classification

Reinald Kim Amplayo*

University of Edinburgh, UK

REINALD.KIM@ED.AC.UK

Seonjae Lim

Samsung Electronics, South Korea

SUN.LIM@SAMSUNG.COM

Seung-won Hwang

Yonsei University, South Korea

SEUNGWONH@YONSEI.AC.KR

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Can a text classifier generalize well for datasets where the text length is different? For example, when short reviews are sentiment-labeled, can these transfer to predict the sentiment of long reviews (i.e., short to long transfer), or vice versa? While unsupervised transfer learning has been well-studied for cross domain/lingual transfer tasks, **Cross Length Transfer (CLT)** has not yet been explored. One reason is the assumption that length difference is trivially transferable in classification. We show that it is not, because short/long texts differ in context richness and word intensity. We devise new benchmark datasets from diverse domains and languages, and show that existing models from similar tasks cannot deal with the unique challenge of transferring across text lengths. We introduce a strong baseline model called BAGGEDCNN that treats long texts as bags containing short texts. We propose a state-of-the-art CLT model called **Length Transfer Networks (LETRANETS)** that introduces a two-way encoding scheme for short and long texts using multiple training mechanisms. We test our models and find that existing models perform worse than the BAGGEDCNN baseline, while LETRANETS outperforms all models.

1. Introduction

Text classification can be categorized according to the text length of the data, from sentence-level classification (Kim, 2014) to document-level classification (Manevitz and Yousef, 2001; Yang et al., 2016). One kind of such task is sentiment classification (Pang et al., 2002), a subtask of sentiment analysis (Pang and Lee, 2007; Liu, 2012) where we are to predict the sentiment/rating given a review written by a user. In some domains, the length of these reviews varies widely. For example, well-known review websites in East Asia such as Naver Movies¹ and Douban Movies² provide two channels for users to write reviews, depending on their preferred length. Figure 1 shows the review channels provided in Naver Movies.

The first channel is a *short review channel*, which contains large amounts of reviews, and enforces users to write short reviews accompanied by rating labels. Although labeled, these reviews lack expressiveness to extract useful information. In contrast, the second channel

* Most work done when both first and second authors were at Yonsei University.

1. <https://movie.naver.com/>
2. <https://movie.douban.com/>

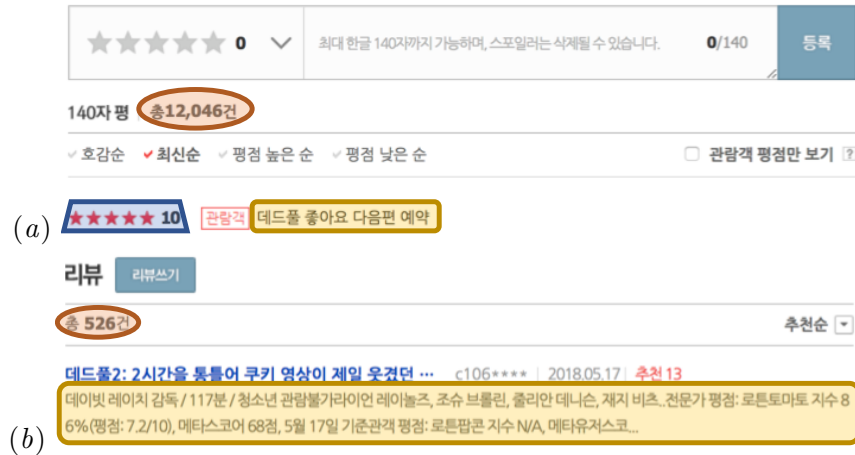


Figure 1: Short review (a) and long review (b) channels for users to write reviews in Naver Movie website. Highlighted areas emphasize the difference in **number of reviews**, **review text length**, and **sentiment label availability**.

is a *long review channel*, which contains few long and detailed reviews with descriptions of different aspects about the product/service. Despite being more expressive, long reviews are often not accompanied by sentiment labels, which most supervised sentiment classification models would require.

We study the “transferability” from one review channel to the other. That is, we try to answer whether a text classifier trained on a dataset with length α can predict a text with length β , where α and β differ by a large margin (e.g., sentences versus paragraphs). This is an important question because there are scenarios where we may have better and more sufficient labeled text data, but we want to classify unlabeled texts with different length. For long to short transfer, more expressive long reviews can be leveraged for training a sentiment classifier for short and context-sparse reviews. For short to long transfer, large amounts of short reviews can be used as supervision to train a classifier for long reviews.

To motivate the non-triviality of such transfer, we train an out-channel (OC) classifier that uses short texts to predict long texts, and an in-channel (IC) classifier that uses long texts on both training and prediction. We also experiment conversely. We use three kinds of classifiers: bag-of-words (BoW), convolutional neural networks (CNN; Kim, 2014), and BERT multilingual (Devlin et al., 2018). We calculate the transfer loss (TL; Glorot et al., 2011), which is the difference between the out-channel and in-channel classifier errors (i.e., TL = 0 means trivially transferable). Table 1 shows that, though using a better inductive bias such as CNN and BERT seems to slightly lower TL, it remains significantly high, consistently suggesting that length transfer is non-trivial.

Our first contribution is thus to define a new task called Cross Length Transfer (CLT). CLT is a task similar to Cross Domain (Blitzer et al., 2007) and Cross Lingual Transfer (Mihalcea et al., 2007) where the difference between the source and target texts is the *text length*, having non-trivial influence that is shown in Table 1. Our second contribution is to show that models from similar tasks (e.g. Cross Domain Transfer and Multiple

	MOV_EN		RES_EN		MOV_KO	
	L>S	S>L	L>S	S>L	L>S	S>L
BoW	4.2	9.8	5.8	9.8	17.5	13.0
CNN	5.5	5.7	5.8	11.8	6.3	14.5
BERT	5.3	5.7	4.5	8.5	3.0	10.0
LETRANETS	-1.2	0.5	3.2	-1.5	0.6	4.5

Table 1: Transfer loss (TL) of different classifiers for each transfer task for all datasets described in Section 2.1.

Instance Learning) are not effective for CLT and even yield negative transfer, as we elaborate in Section 2.2 and empirically show in Section 4. Finally, we present two new models specifically for CLT: a strong baseline called BAGGEDCNN that treats long texts as bags containing short texts, and a state-of-the-art CLT model called Length Transfer Networks (LETRANETS). LETRANETS enables a two-way encoding scheme using multiple training mechanisms, and accepts both short and long text inputs, where one such input is created artificially through concatenation or segmentation. Table 1 shows that LETRANETS has the best transfer loss, and sometimes perform better than in-channel classifier (when TL is less than zero).

We test our models using the multiple benchmark datasets we gathered and show that models from other tasks perform worse than our proposed strong baseline and that LETRANETS performs the best among all models. To the best of our knowledge, we are the first to study CLT.

2. Cross Length Transfer

Cross Length Transfer (CLT) is an unsupervised transfer learning task in which the setting is that the sampling distributions of the training and test data are different because the texts lengths are different (e.g., sentences and paragraphs).

Formally, we suppose two sets of texts: a source set \mathcal{S} in which we have labels, and a target set \mathcal{T} in which we want to predict the labels. Moreover, we know that the text length distributions of \mathcal{S} and \mathcal{T} are different, such that an equality case exists as $|\mathcal{S}| = r|\mathcal{T}|$, where $|\mathcal{X}|$ is the mean length of the set \mathcal{X} , and $r \neq 1$ is a non-negative rate of difference between two mean lengths. There are two subtasks: **long to short transfer** where $r > 1$ and thus \mathcal{S} contains longer texts, and **short to long transfer** where $r < 1$ and thus \mathcal{S} contains shorter texts. A CLT model should effectively learn to predict the labels of \mathcal{T} , on both scenarios.

A concrete and simple example is when \mathcal{S} contains labeled sentence reviews and \mathcal{T} contains unlabeled paragraph reviews. A CLT model uses \mathcal{S} for training to effectively predict labels of reviews in \mathcal{T} . Also, the same CLT model should be able to do effective prediction vice versa, i.e., when \mathcal{S} are paragraph reviews and \mathcal{T} are sentence reviews.

Previous unsupervised transfer learning tasks, i.e. Cross Domain Transfer (Blitzer et al., 2007) and Cross Lingual Transfer (Wan, 2009), are similar to CLT but have concrete differences. Generally, the goal of these tasks is to map semantic domains, contextually or linguistically, of both \mathcal{S} and \mathcal{T} into a shared space, by aligning the vocabulary (Pan et al.,

Dataset	Length	#Train	#Test	#Unlabeled	Words/Instance
MOV_EN	short	1600	400	8000	22
	long	1600	400	8000	372
RES_EN	short	1600	400	8000	22
	long	1600	400	8000	249
MOV_KO	short	1600	400	4000	12
	long	1600	400	0	57

Table 2: Dataset statistics.

2010; Shi et al., 2018), expanding domain-specific lexicons (Qiu et al., 2009; He et al., 2011), generating labeled samples (Yu and Jiang, 2016), and learning to indiscriminate between domains (Chen et al., 2016; Liu et al., 2018). These methods are generally symmetric; i.e., even when \mathcal{S} and \mathcal{T} interchange, the same method can be applied easily.

However, in CLT, both \mathcal{S} and \mathcal{T} are already in the same contextual and linguistic domains, thus previous methods would not work. Also, CLT brings two new challenges against devising a symmetric model. First, texts with different context richness may have different properties they focus on: hierarchical structures (Yang et al., 2016) may be more important for document-level reviews while finding lexical/phrasal cues (Kim, 2014) may be more important for sentence-level reviews. Second, words on texts with different lengths may have different semantic intensity. For example, “*good*” may have a very high positive sentiment intensity on short texts, and a relatively low positive sentiment intensity on long ones.

2.1. Benchmark Datasets

We provide three pairs of short/long datasets from different domains (movies and restaurants) and from different languages (English and Korean) suitable for the task: MOV_EN, RES_EN, and MOV_KO. Most of the datasets are from previous literature and are gathered differently. The MOV_EN datasets are gathered from different websites; the short dataset consists of hand-picked sentences by Pang and Lee (2005) from document-level reviews from the Rotten Tomatoes website, while the long dataset consists of reviews from the IMDB website obtained by Tang et al. (2015b). The RES_EN dataset consists of reviews from Yelp, where the short dataset consists of reviews with character lengths less than 140 from Amplayo and Hwang (2017), while reviews in the long dataset are gathered from Tang et al. (2015b). We also share new short/long datasets MOV_KO, which are gathered from two different channels, as shown in Figure 1, available in Naver Movies.

Unlike previous datasets (Blitzer et al., 2007; Glorot et al., 2011) where they used polarity/binary (e.g., positive or negative) labels as classes, we also provide fine-grained classes, with five classes of different sentiment intensities (e.g., 1 is strong negative, 5 is strong positive), for RES_EN and MOV_KO. Following the Cross Domain Transfer setting (Blitzer et al., 2007; Ziser and Reichart, 2017; Ruder and Plank, 2018), we limit the size of the dataset to be small-scale to focus on the main task at hand. This ensures that models focus on the transfer task, and decrease the influence of other factors that can be found when using larger datasets. Finally, following Glorot et al. (2011), we provide

additional unlabeled data for those models that need them (Blitzer et al., 2007; Ziser and Reichart, 2017), except for the long dataset of MOV_KO, where the labeled reviews are very limited. We show the dataset statistics in Table 2, and share the datasets here: <https://github.com/rktamplayo/LeTraNets>.

2.2. Possible Existing Solutions

Cross Domain Transfer (CDT) CDT offers models that effectively transfer domain-independent features from two different domains. The most popular non-neural CDT model is Structural Correspondence Learning (SCL; Blitzer et al., 2007), a method that identifies feature correspondence from different domains using pivot features. A recent neuralized extension is Neural SCL (NEUSCL; Ziser and Reichart, 2017), in which an autoencoder module is integrated to SCL. The CDT literature is vast, and we refer the readers to Pan et al. (2012) and Tan et al. (2018) for overviews. Although these models may see improvements due to a possible difference in vocabulary (especially when the review channels are different), these improvements may be marginal since the domain of the datasets is the same.

Multiple Instance Learning (MIL) MIL is a task where given the labels of a bag of multiple instances, we are to label the individual instances (Zhou et al., 2009). In the text classification domain, MIL is often devised as segment-level classification (Kotzias et al., 2015; Angelidis and Lapata, 2018), where documents are bags and sentences in the documents are segments. The most recent MIL model is the Multiple Instance Learning Network (MILNET; Angelidis and Lapata, 2018), where they used attention-based polarity scoring to identify segment labels. MIL models can be used in *long to short transfer*, where we assume that segment labels in long texts can be used to label short reviews. However, they (a) assume that segments from long data, which rely on inter-sentence semantics, are comparable to self-contained short texts, and (b) are ineffective on short to long transfer because it needs multiple sentences to train components of the model for document-level classification.

Weak Supervision A simple yet possible solution for *short to long transfer* is a three-step approach where we (1) cluster the short texts into several long texts, (2) infer the class labels of the clusters, and (3) use the labeled clusters as weak supervision to create a classifier. Micro Aspect Sentiment Model (MASM; Amplayo and Hwang, 2017) does (1) and (2) automatically. For (3), we can train a classifier such as CNNs (Kim, 2014) to predict labels of long texts. One critical issue of this solution is that since both clustering and class labels are inferred, there is a high chance that at least one of them is incorrect. This thus creates compounding errors that decrease the performance of the model.

3. Our Models

3.1. BaggedCNN: A Strong Baseline

We present BAGGEDCNN, a simple yet strong baseline to the CLT task. BAGGEDCNN is a model derived from MILNET (Angelidis and Lapata, 2018). MILNET uses CNN to encode segments, BiGRU (Bahdanau et al., 2014) to calculate attention weights, and gated polarity

to calculate document-level probabilities. We refer the readers to the original paper for more details. We improve using two key modifications: (a) removing the sequential connections (i.e., BiGRU) between segments, and (b) using a single classifier for both the segments and full document.

For each document divided into segments $D = \{S_i\}$, BAGGEDCNN starts by encoding the segments using a CNN classifier called CNN_{bag} . Then, we pool the segment encodings into one vector using attention mechanism. Finally, we use a logistic regression classifier that can be used to classify either the segments or the document. This is possible since the vectors of both segments and document are in the same vector space:

$$\begin{aligned} s_i &= \text{CNN}_{bag}(S_i) \\ a_{s_i} &= \text{softmax}(v^\top \tanh(W_a s_i + b_a)) \\ d &= \sum_i s_i * a_{s_i} \\ y_d &= \text{softmax}(W_c d + b_c) \\ y_{s_i} &= \text{softmax}(W_c s_i + b_c) \end{aligned}$$

The model is trained differently depending on the transfer task: For *long to short transfer*, we minimize the cross-entropy loss between the actual and predicted class of the document \mathcal{L}_d . For *short to long transfer*, we minimize the mean cross entropy loss between the actual and predicted class of the segments $\sum \mathcal{L}_{s_i}/n, 1 \leq i \leq n$. Note that BAGGEDCNN is reduced to a model where average pooling is done instead of the attention mechanism. At test time, we use y_d for classification.

While it has been shown that removing the sequential structure in the document level (i.e., BiGRU in the case of MILNET) decreases the performance of the document classifier (Tang et al., 2015a; Yang et al., 2016), we argue that this removal is effective on the CLT task because of inter-segment independence. That is, sentences in the document are treated similar to short texts. We also show in our experiments that BAGGEDCNN performs better than MILNET.

However, BAGGEDCNN still fails to consider two things. First, while the model relaxes the strong assumption on similarity between segments and short texts, by removing the sequential connections, most segments cannot be treated as stand-alone short texts. For example, the segment “*Yet it is salty.*” is not a stand-alone short review. Second, when doing short to long transfer, the input short text is just one segment, thus the model is reduced into a weaker hierarchical CNN classifier.

3.2. LeTraNets: Length Transfer Networks

We improve BAGGEDCNN by proposing a model called Length Transfer Networks (LETRANETS), as shown in Figure 2. LETRANETS is composed of two classifiers: a stand-alone CNN classifier with text encoder CNN_{lone} , and BAGGEDCNN, which includes a segment-level text encoder CNN_{bag} . The CNN_{lone} encoder is used to capture holistic textual features, while the CNN_{bag} encoder is used to capture segment-level textual features, assuming there is a bigger text that owns the segments.

For each data instance, LETRANETS accepts two kinds of inputs: a long text $D = w_d$ and a set of short texts $S = w_{s_0}, \dots, w_{s_n}$. However, the task setting only provides either one

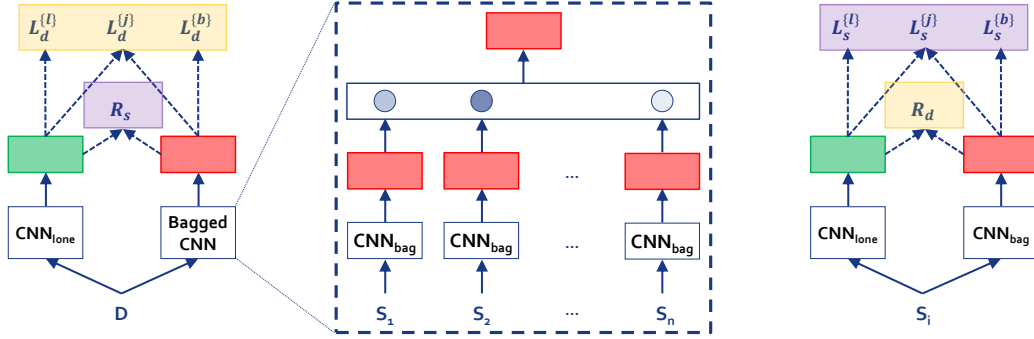


Figure 2: Full architecture of LeTRANETS. $\mathcal{L}_x^{\{a\}}$ refers to cross entropy loss of classifier a ($l: p(y^{\{l\}})$, $b: p(y^{\{b\}})$, $j: p(y^{\{j\}})$) using text x (s : short text, d : long text). \mathcal{R}_x is the prediction regularization. The highlighted boxes refer to the **long to short** and **short to long** transfer-specific configurations. We share our implementation here: <https://github.com/rktamplayo/LeTraNets>.

of long texts or short texts as input. We thus create pseudo-texts from the available text data through the following methods. In the long to short transfer task, we use segments in long texts as pseudo-short texts, as used in BAGGEDCNN. In the short to long transfer task, we concatenate a random number of short texts to create pseudo-long texts. The latter amounts to a possibly infinite number of long texts we can use for training.

The short texts are encoded by both CNN_{lone} and CNN_{bag} as $s_i^{\{l\}}$ and $s_i^{\{b\}}$. The long texts are encoded using both CNN_{lone} and BAGGEDCNN as $d^{\{l\}}$ and $d^{\{b\}}$:

$$\begin{aligned} s_i^{\{l\}} &= \text{CNN}_{lone}(S_i) \\ s_i^{\{b\}} &= \text{CNN}_{bag}(S_i) \\ d^{\{l\}} &= \text{CNN}_{lone}(D) \\ d^{\{b\}} &= \text{BAGGEDCNN}(D) \end{aligned}$$

The encoded long text vectors $d^{\{l\}}$ and $d^{\{b\}}$ and short text vectors $s_i^{\{b\}}$ and $s_i^{\{l\}}$ are used to classify their labels using softmax classifiers specific to the CNN encoders:

$$\begin{aligned} p(y_d^{\{l\}}) &= \text{softmax}(W_c^{\{l\}}d^{\{l\}} + b_c^{\{l\}}) \\ p(y_d^{\{b\}}) &= \text{softmax}(W_c^{\{b\}}d^{\{b\}} + b_c^{\{b\}}) \\ p(y_{s_i}^{\{l\}}) &= \text{softmax}(W_c^{\{l\}}s_i^{\{l\}} + b_c^{\{l\}}) \\ p(y_{s_i}^{\{b\}}) &= \text{softmax}(W_c^{\{b\}}s_i^{\{b\}} + b_c^{\{b\}}) \end{aligned}$$

Training Mechanisms There are two main issues when training the model in the CLT setting. First, both the stand-alone CNN classifier and BAGGEDCNN are disconnected, acting as two individual classifiers. Second, the model needs both labels for both short and long text data, but we are only given labels for one kind of data during training for each

transfer setting. Solving the second issue is crucial for short to long transfer, as we cannot train the full model if we do not have labels for long data. To this end, we use three training mechanisms below that help mitigate these issues.

We connect them on different levels. In the word-level, we use the same word embedding space for both classifiers. Beyond word-level, we use a training mechanism called **Joint Training (JT)**. This concatenates the encoded text vectors, and creates another logistic regression classifier for the concatenated vector. This creates a connection between classifiers at the classification-level.

$$\begin{aligned} p(y_d^{\{j\}}) &= \text{softmax}(W_c^{\{j\}}[d^{\{l\}}; d^{\{b\}}] + b_c^{\{j\}}) \\ p(y_{s_i}^{\{j\}}) &= \text{softmax}(W_c^{\{j\}}[s_i^{\{l\}}; s_i^{\{b\}}] + b_c^{\{j\}}) \end{aligned}$$

Beyond word-level, we introduce **Prediction Regularization (PR)** mechanism to train encoders with no labels. This regularizes the predictions of a weaker classifier based on the predictions of a stronger classifier. We consider BAGGEDCNN as the stronger classifier for long to short transfer, and CNN_{long} as the stronger classifier for short to long transfer. We use Kullback-Leibler divergence as the regularization function.

$$\begin{aligned} \mathcal{R}_d &= \sum_i \mathcal{D}_{KL}(y_{s_i}^b || y_{s_i}^l) \\ \mathcal{R}_s &= \mathcal{D}_{KL}(y_d^l || y_d^b) \end{aligned}$$

Finally, using the PR mechanism directly might not work because predictions from the stronger classifier may not be optimized yet. Hence, we use **Stepwise Pretraining (SP)** mechanism to pretrain specific parts of the model in a step-by-step fashion. First, we pretrain the stronger classifier, then the weaker classifier with PR mechanism, and finally the classifier of the JT mechanism. After pretraining, we train the full model.

The training configurations are different depending on the transfer task, which is also shown in Figure 2. For *long to short transfer*, we use $p(y_d^{\{j\}})$ for the JT mechanism and R_d for the PR mechanism. For *short to long transfer*, we use $p(y_{s_i}^{\{j\}})$ for the JT mechanism and R_s for the PR mechanism.

The final training objective is to minimize the loss function, depending on the text length:

$$\begin{aligned} \mathcal{L}_d &= \mathcal{L}_d^{\{l\}} + \mathcal{L}_d^{\{b\}} + \mathcal{L}_d^{\{j\}} + \lambda \mathcal{R}_d \\ \mathcal{L}_s &= \sum_i (\mathcal{L}_{s_i}^{\{l\}} + \mathcal{L}_{s_i}^{\{b\}} + \mathcal{L}_{s_i}^{\{j\}}) / n + \lambda \mathcal{R}_s \end{aligned}$$

where $\mathcal{L}_x^{\{a\}}$ is the cross-entropy loss between the actual and predicted values of the classifier $p(y_x^{\{a\}})$, and λ is tuned using a development set. At test time, we use $p(y_d^{\{j\}})$ and $p(y_{s_i}^{\{j\}})$ to classify the sentiment for long to short and short to long transfer, respectively.

4. Experiments

Experimental Settings The dimensions of word vectors are set to 300. We use pre-trained GloVe embeddings³ (Pennington et al., 2014) to initialize our English word vectors,

3. <https://nlp.stanford.edu/projects/glove/>

Model	MOV_EN		RES_EN		MOV_KO		
	P_ACC	P_ACC	F_ACC	F_RMSE	P_ACC	F_ACC	F_RMSE
<i>Long to Short Transfer Task</i>							
CNN	0.728	0.750	0.388	1.440	0.600	0.228	1.678
CNNx2	0.735	0.756	0.401	1.431	0.603	0.228	1.674
SCL	0.640	0.685	0.340	1.857	0.438	0.208	1.754
NEUSCL	0.645	0.665	0.373	1.765	0.428	0.203	1.709
SCL+CNN	0.710	0.788	0.390	1.280	0.578	0.230	1.811
NEUSCL+CNN	0.730	0.798	0.425	1.444	0.600	0.230	1.683
MILNET	0.757	0.785	0.400	1.426	0.620	0.212	1.685
BAGGEDCNN	0.767	0.765	0.468	1.190	0.625	0.245	1.614
LeTraNets	0.795*	0.863*	0.502*	1.081*	0.652	0.265	1.591
<i>Short to Long Transfer Task</i>							
CNN	0.758	0.780	0.390	1.374	0.552	0.453	0.973
CNNx2	0.763	0.784	0.396	1.372	0.559	0.460	0.960
SCL	0.700	0.738	0.343	1.361	0.488	0.220	1.568
NEUSCL	0.725	0.725	0.385	1.251	0.480	0.245	1.546
SCL+CNN	0.720	0.753	0.403	1.190	0.555	0.345	1.250
NEUSCL+CNN	0.783	0.778	0.448	1.366	0.568	0.460	0.885
MASM+CNN	0.570	0.600	0.235	2.322	0.543	0.045	1.658
BAGGEDCNN	0.792	0.838	0.438	1.224	0.585	0.465	0.865
LeTraNets	0.810	0.858	0.478*	1.138*	0.625*	0.493	0.802

Table 3: Accuracy and RMSE of competing models on polarity (P_ACC) and fine-grained (F_ACC and F_RMSE) datasets. Items in red are performances worse than the no-transfer CNN baseline. An asterisk (*) indicates that LeTraNets is significantly better than the second best model ($p < 0.05$).

and pre-trained FastText embeddings⁴ (Grave et al., 2018) to initialize our Korean word vectors. For all CNNs, we set $h = 3, 4, 5$, each with 100 feature maps, following (Kim, 2014). We use dropout (Srivastava et al., 2014) on all non-linear connections with a dropout rate of 0.5. We set the batch size to 32. We use stochastic gradient descent over shuffled mini-batches with the Adadelta update rule (Zeiler, 2012) with l_2 constraint of 3. We experiment with a 5-fold cross-validation on the given source training set and report the average results.

Comparison Models We compare our models with the models from similar tasks as discussed in Section 2.2. Specifically, we compare with (a) Cross Domain Transfer (CDT) models SCL (Blitzer et al., 2007) and NEUSCL (Ziser and Reichart, 2017), (b) CDT models with a CNN classifier integration (Yu and Jiang, 2016) (SCL+CNN and NEUSCL+CNN), (c) a multiple-instance learning (MIL) model MILNET (Angelidis and Lapata, 2018), (d) a weakly supervised model MASM+CNN Amplayo and Hwang (2017). We remind that MILNET is only applicable to long to short transfer, and MASM+CNN is only applicable to short to long transfer. We use the available code provided by previous authors⁵. Finally, we also compare with CNN (Kim, 2014), and a combination of two CNNs (CNNx2) as no-transfer baselines.

4. <https://fasttext.cc/>

5. SCL and NEUSCL: <https://github.com/yftah89/Neural-SCL-Domain-Adaptation>, MILNET: <https://github.com/stangelid/milnet-sent>, MASM: <https://github.com/rktamplayo/MicroASM>

Dataset and Evaluation We use the datasets described in Table 2 for all our experiments. We use the following evaluation metrics. For all datasets, we use accuracy (ACC) to measure the overall sentiment classification performance. Additionally, for fine-grained datasets, we use root mean squared error (RMSE) to measure the divergence between the predicted and ground truth sentiment scores. Finally, in order to compare models in an integrated manner, we report the average transfer ratio (TR; Glorot et al., 2011), a version of the transfer loss which is more adaptive to averaging, calculated as the average quotient between the transfer error and the in-domain baseline error, i.e. $TR = \sum_x e(\mathcal{S}_x, \mathcal{T}_x) / e_b(\mathcal{T}_x, \mathcal{T}_x)$, where \mathcal{S}_x and \mathcal{T}_x are the source and domain of dataset x , respectively, e and e_b are accuracy errors from the competing model and the baseline CNN model.

Long to Short Transfer We show the results for long to short transfer in the first part of Table 3. Results show that Cross Domain Transfer models do not perform well, which confirms our hypothesis that they are not well suited for this task. MILNET performs well on polarity tasks, but performs poorly on fine-grained tasks, having worse performance than the no-transfer CNN baseline. This shows that although Multiple Instance Learning models are effective in classifying positive or negative sentiments, they are not flexible to fine-grained sentiment intensities, which differs when text lengths are different. On the other hand, BAGGEDCNN performs better than MILNET, proving that simplifying MIL models work well on CLT. Overall, LETRANETS performs the best among all models, having the best accuracies and RMSEs on all datasets and settings.

Short to Long Transfer We report the results for short to long transfer in the second part of Table 3. Results show that Cross Domain Transfer models perform much worse compared to their performance in the long to short transfer task. The weak supervised model MASM+CNN performs the worst, having worse results than the no-transfer CNN baseline on all datasets. BAGGEDCNN also performs well in this task, even though it does not use its attention mechanism. This shows that BAGGEDCNN is a very tough-to-beat baseline for the CLT task. Finally, LETRANETS also outperforms all the models on this subtask.

Transfer Ratio (TR) Figure 3 shows the average transfer ratio (TR) of all competing models, where $TR = 1$ means trivially transferable. The figure shows that the CDT models SCL and NEUSCL both obtain a larger transfer ratio compared to the no-transfer CNN baseline. The transfer ratios improve when CNN is integrated into both models, but does not improve much from the baseline. MILNET and BAGGEDCNN perform comparably on the long to short transfer task, where BAGGEDCNN performs slightly better. LETRANETS performs the best among the models, having transfer ratios less than 1.1.

5. Analyses

Ablation on Training Mechanisms We investigate the performance of LETRANETS when the training mechanisms are not used. Specifically, we perform ablation tests on the Joint Training (JT), Prediction Regularization (PR), and Stepwise Pretraining (SP) mechanisms. The results in Table 4 show that LETRANETS performs the best when all training mechanisms are used. Also, when used individually, all the training mechanisms

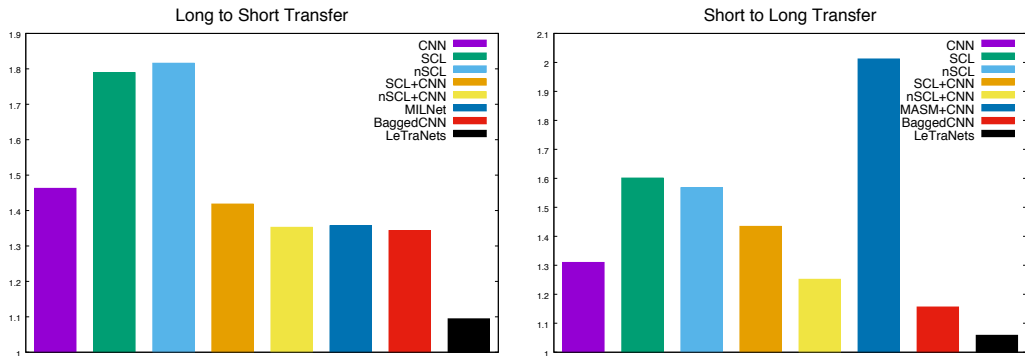


Figure 3: Average transfer ratio of all competing models for both transfer task. Lower is better.

	MOV_EN P_ACC	RES_EN P_ACC	F_ACC	MOV_KO P_ACC	F_ACC
<i>Long to Short Transfer Task</i>					
-	0.767	0.800	0.470	0.630	0.240
JT	0.782	0.838	0.480	0.637	0.255
PR	0.777	0.848	0.487	0.637	0.253
SP	0.782	0.845	0.495	0.635	0.260
All	0.795	0.863	0.502	0.652	0.265
<i>Short to Long Transfer Task</i>					
-	0.780	0.813	0.435	0.595	0.465
JT	0.797	0.835	0.453	0.620	0.468
PR	0.813	0.830	0.448	0.610	0.465
SP	0.808	0.835	0.475	0.610	0.468
All	0.810	0.858	0.478	0.625	0.493

Table 4: Effect (in ACC) of using the training mechanisms Joint Training (JT), Prediction Regularization (PR), and Stepwise Pretraining (SP) in LETRANETS.

boost up the performance of the model. Hence, we confirm that the training mechanisms help LETRANETS achieve good performance on the task.

Performance per Text Length We check the capability of LETRANETS to transfer across text lengths, by looking at its performance as the text length increases. Specifically, we compare the performance per text length of LETRANETS and CNN models, trained on either short texts (LETRANETS_{short} and CNN_{short}) or long texts (LETRANETS_{long} and CNN_{long}), on RES_EN short/long datasets. Figure 4 shows the results. CNN performs well when the text length is similar to the training dataset and performs poorly otherwise. LETRANETS, however, performs similarly on all kinds of text lengths although it is trained purely on a dataset of a specific length. More interestingly, LETRANETS_{short} performs *better* than LETRANETS_{long} on longer texts, and unexpectedly performs *worse* on shorter texts. This suggests that LETRANETS weakens its ability to classify texts with the same length

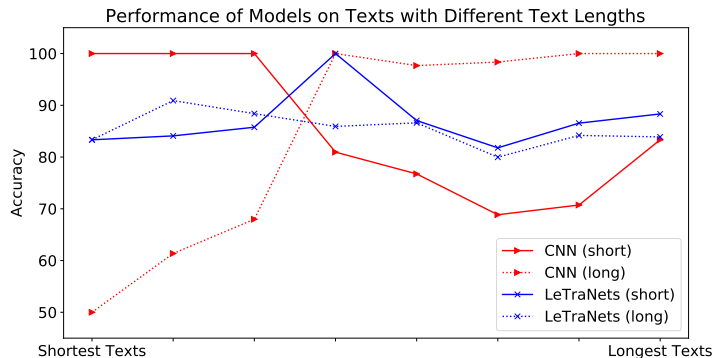


Figure 4: Accuracy of different models across different text lengths on the RES_EN dataset. The legend `model (length)` means that the `model` is trained on `length` dataset.

and improves its ability to classify texts with different length. This property is acceptable in our problem setup since we care on effectively classifying short (or long) texts more, assuming we only have access to long (or short) texts as training data. However, future work should explore on CLT models that perform well on both text lengths.

On Topic Diversity Longer texts can discuss diverse topics, while shorter texts are limited to few (or one) topics. In the sentiment classification domain, longer reviews may mention positive sentiments towards an aspect of a product, and then talk about negative sentiments towards another aspect. With this hypothesis, we examine whether LETRANETS can handle longer texts with diverse topics when trained on short texts. Specifically, we compare the performance per topic diversity of LETRANETS and CNN models, trained on short texts of RES_EN dataset. We measure topic diversity as the Shannon index (Shannon, 1948) of the topic distribution inferred by an LDA topic model (Blei et al., 2003) fit using the unlabeled data. Figure 5 shows the results. Results indicate that the performance increase of LETRANETS over CNN increases as the diversity of topics increases. This shows that for short to long transfer, LETRANETS is able to handle texts with topics that are more diverse, even when trained on short texts, which tend to have less diverse topics.

Cross Domain and Length Transfer Which between domain and text length should we consider to achieve a better performance? To answer this question, we combine Cross Domain Transfer (CDT) and Cross Length Transfer (CLT) into one task: Cross Domain and Length Transfer (CDLT) and compare the performance of CDT and CLT models on the task. We use the MOV_EN and RES_EN datasets to create four CDLT datasets, and check which between the CDT model NEUSCL+CNN and the CLT model LETRANETS achieves a higher increase in performance. The results are shown in Table 5. We find that NEUSCL+CNN performs worse, obtaining accuracies worse than that of the no-transfer CNN baseline. LETRANETS performs better, obtaining significant increase in performance from the baseline. This shows that solving the non-transferability of length is more important to achieve a more effective sentiment classifier.

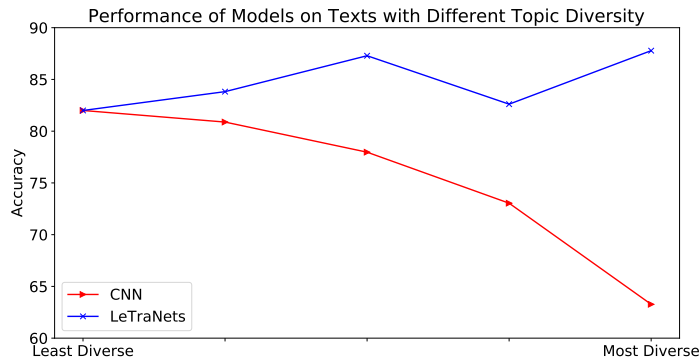


Figure 5: Accuracy of different models across different topic diversity on the RES_EN dataset.

Model	Long to Short		Short to Long	
	M>R	R>M	M>R	R>M
CNN	0.755	0.620	0.755	0.728
NEUSCL+CNN	0.710	0.638	0.695	0.585
LeTraNets	0.828	0.745	0.790	0.785

Table 5: Accuracies of models on four CDLT datasets from M:MOV_EN and R:RES_EN datasets.

6. Conclusions

We defined a new task called Cross Length Transfer (CLT) to check the transferability across lengths of classification models. We set the grounds by defining the task, providing three benchmark datasets from different domains and languages, and introducing models from related tasks. We proposed two models: a strong baseline model called BAGGEDCNN, and LETRANETS, a model that improves over the weakness of BAGGEDCNN. Our multiple experiments show that LETRANETS demonstrates superior performance over all competing models. We aim to apply the CLT to other classification tasks, such as natural language inference (Bowman et al., 2015), where text length is influential towards overall model performance (Williams et al., 2018).

Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. Amplayo is grateful to be supported by a Google PhD Fellowship. This research was supported by MSIT (Ministry of Science and ICT), Korea, under ITRC program (IITP-2019-2016-0-00464) supervised by IITP. Hwang is the corresponding author.

References

- Reinald Kim Amplayo and Seung-won Hwang. Aspect sentiment model for micro reviews. In *ICDM*, pages 727–732, 2017. doi: 10.1109/ICDM.2017.83. URL <https://doi.org/10.1109/ICDM.2017.83>.
- Stefanos Angelidis and Mirella Lapata. Multiple instance learning networks for fine-grained sentiment analysis. *TACL*, 6:17–31, 2018. URL <https://transacl.org/ojs/index.php/tacl/article/view/1225>.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL <http://arxiv.org/abs/1409.0473>.
- David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007. URL <http://aclweb.org/anthology/P07-1056>.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, pages 632–642, 2015. URL <http://aclweb.org/anthology/D/D15/D15-1075.pdf>.
- Xilun Chen, Ben Athiwaratkun, Yu Sun, Kilian Q. Weinberger, and Claire Cardie. Adversarial deep averaging networks for cross-lingual sentiment classification. *CoRR*, abs/1606.01614, 2016. URL <http://arxiv.org/abs/1606.01614>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL <http://arxiv.org/abs/1810.04805>.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520, 2011. URL http://www.icml-2011.org/papers/342_icmlpaper.pdf.
- Edouard Grave, Piotr Bojanowski, Prakhara Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *LREC*, 2018. URL <http://aclweb.org/anthology/L18-1550>.
- Yulan He, Chenghua Lin, and Harith Alani. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *ACL*, pages 123–131, 2011. URL <http://www.aclweb.org/anthology/P11-1013>.
- Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, 2014. URL <http://aclweb.org/anthology/D/D14/D14-1181.pdf>.
- Dimitrios Kotzias, Misha Denil, Nando de Freitas, and Padhraic Smyth. From group to individual labels using deep features. In *KDD*, pages 597–606, 2015. doi: 10.1145/2783258.2783380. URL <https://doi.org/10.1145/2783258.2783380>.

- Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012. doi: 10.2200/S00416ED1V01Y201204HLT016. URL <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>.
- Qi Liu, Yue Zhang, and Jiangming Liu. Learning domain representation for multi-domain sentiment classification. In *NAACL-HLT*, pages 541–550, 2018. URL <https://aclanthology.info/papers/N18-1050/n18-1050>.
- Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *JMLR*, 2:139–154, 2001. URL <http://www.jmlr.org/papers/v2/manevitz01a.html>.
- Rada Mihalcea, Carmen Banea, and Janyce Wiebe. Learning multilingual subjective language via cross-lingual projections. In *ACL*, 2007. URL <http://aclweb.org/anthology/P07-1123>.
- Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *WWW*, pages 751–760, 2010. doi: 10.1145/1772690.1772767. URL <https://doi.org/10.1145/1772690.1772767>.
- Weike Pan, Erheng Zhong, and Qiang Yang. Transfer learning for text mining. In *Mining Text Data*, pages 223–257. Springer, 2012. doi: 10.1007/978-1-4614-3223-4_7. URL https://doi.org/10.1007/978-1-4614-3223-4_7.
- Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124, 2005. URL <http://aclweb.org/anthology/P/P05/P05-1015.pdf>.
- Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2007. doi: 10.1561/1500000011. URL <https://doi.org/10.1561/1500000011>.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*, 2002. URL <https://aclanthology.info/papers/W02-1011/w02-1011>.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. URL <http://aclweb.org/anthology/D/D14/D14-1162.pdf>.
- Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding domain sentiment lexicon through double propagation. In *IJCAI*, pages 1199–1204, 2009. URL <http://ijcai.org/Proceedings/09/Papers/202.pdf>.
- Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. In *ACL*, pages 1044–1054, 2018. URL <https://aclanthology.info/papers/P18-1096/p18-1096>.
- Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

- Bei Shi, Zihao Fu, Lidong Bing, and Wai Lam. Learning domain-sensitive and sentiment-aware word embeddings. In *ACL*, pages 2494–2504, 2018. URL <https://aclanthology.info/papers/P18-1232/p18-1232>.
- Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014. URL <http://dl.acm.org/citation.cfm?id=2670313>.
- Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *ICANN*, pages 270–279, 2018. doi: 10.1007/978-3-030-01424-7_27. URL https://doi.org/10.1007/978-3-030-01424-7_27.
- Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pages 1422–1432, 2015a. URL <http://aclweb.org/anthology/D/D15/D15-1167.pdf>.
- Duyu Tang, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL-IJCNLP*, pages 1014–1023, 2015b. URL <http://aclweb.org/anthology/P/P15/P15-1098.pdf>.
- Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *ACL-IJCNLP*, pages 235–243, 2009. URL <http://www.aclweb.org/anthology/P09-1027>.
- Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, pages 1112–1122, 2018. URL <https://aclanthology.info/papers/N18-1101/n18-1101>.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *NAACL-HLT*, pages 1480–1489, 2016. URL <http://aclweb.org/anthology/N/N16/N16-1174.pdf>.
- Jianfei Yu and Jing Jiang. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *EMNLP*, pages 236–246, 2016. URL <http://aclweb.org/anthology/D/D16/D16-1023.pdf>.
- Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL <http://arxiv.org/abs/1212.5701>.
- Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *ICML*, pages 1249–1256, 2009. doi: 10.1145/1553374.1553534. URL <https://doi.org/10.1145/1553374.1553534>.
- Yftah Ziser and Roi Reichart. Neural structural correspondence learning for domain adaptation. In *CoNLL*, pages 400–410, 2017. doi: 10.18653/v1/K17-1040. URL <https://doi.org/10.18653/v1/K17-1040>.