Text Categorization by Learning Predominant Sense of Words as Auxiliary Task

Kazuya Shimura[1], **Jiyi Li**[2,3] **and Fumiyo Fukumoto**[2]
Graduate School of Engineering, University of Yamanashi[1]
Interdisciplinary Graduate School, University of Yamanashi[2]
4-3-11, Takeda, Kofu, 400-8511 Japan
RIKEN AIP[3], Tokyo, 103-0027 Japan
{g17tk008,jyli,fukumoto}@yamanashi.ac.jp

Abstract

Distributions of the senses of words are often highly skewed and give a strong influence of the domain in a document. This paper follows the assumption and presents a method for text categorization by leveraging the predominant sense of words depending on the domain, i.e., domain-specific senses. The key idea is that the features learned from predominant senses are possible to discriminate the domain of the document and thus improve the overall performance of text categorization. We propose a multi-task learning framework based on the neural network model, transformer, which trains a model to simultaneously categorize documents and predicts a predominant sense for each word. The experimental results using four benchmark datasets including RCV1 show that our method is comparable to the state-of-the-art categorization approach, especially our model works well for categorization of multi-label documents.

1 Introduction

Text categorization has been intensively studied since neural network methods have attracted much attention. Most of the previous work on text categorization relies on the use of representation learning where the words are mapped to an implicit semantic space (Wang et al., 2015; Liu et al., 2017a). The Word2Vec is a typical model related to this representation (Mikolov et al., 2013). It learns a vector representation for each word and captures semantic information between words. Pre-training by using the model shows that it improves overall performance in many NLP tasks including text categorization. However, the drawback in the implicit representation is that it often does not work well on polysemous words.

The sense of a word depends on the domain in which it is used. The same word can be used differently in different domains. Distributions of the senses of words are often highly skewed and a predominant sense of a word depends on the domain of a document (McCarthy et al., 2007; Jin et al., 2009). Suppose the noun word, "court". The predominant sense of a word "court" would be different in the documents from the "judge/law" and "sports" domains as the sense of the former would be "*an assembly (including one or more judges) to conduct judicial business*" and the latter is "*a specially marked horizontal area within which a game is played*" described in the WordNet 3.1. This indicates that the meaning becomes a strong clue to assign a domain to the document. However, in the implicit semantic space created by using the neural language model such as the Word2Vec, a word is represented as one vector even if it has several senses.

It is often the case that a word which is polysemous is not polysemous in a restricted subject domain. A restriction of the subject domain makes the problem of polysemy less problematic. However, even in texts from a restricted subject domain such as Wall Street Journal corpus (Douglas and Janet, 1992), one encounters quite a large number of polysemous words. Several authors focused on the problem and proposed a new type of deep contextualized word representation such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) that models not only syntax but also semantics including polysemies. Their methods work very well in many NLP tasks such as question answering and sentiment analysis, while their methods are unsupervised manners which they do not explicitly map each sense of a word to its domain. Motivated by solving this problem, we propose a method for text categorization that complements implicit representation by leveraging the predominant sense of a word.

We propose a multi-task learning method based on the encoder structure of the neural network

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1109–1119
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

model, transformer (Vaswani et al., 2017). The transformer works by relying on a self-attention mechanism. It can directly capture the relationships between two words regardless of their distance which is effective for detecting features to discriminate predominant sense of a word in the domain. In the model using multi-task learning, the auxiliary predominant sense prediction task helps text categorization by learning common feature representation of predominant senses for text categorization. The model adopts a multi-task objective function and is trained to simultaneously categorize texts and predicts a predominant sense for each word. In such a way, the predominant sense information can also help the model to learn better sense/document representations. The experimental results using four benchmark datasets support our conjecture that predominant sense identification helps to improve the overall performance of the text categorization task.

The main contributions of our work can be summarized: (1) We propose a method for text categorization that complements implicit representation by leveraging a predominant sense of a word. (2) We introduce a multi-task learning framework based on the neural network model, transformer. (3) We show our hypothesis that predominant sense identification helps to improve the overall performance of the text categorization task, especially our model is effective for categorization of documents with multi-label.

2 Text Categorization Framework

Our multi-task learning framework for predominant sense prediction and text categorization is illustrated in Figure 1.

2.1 Text Matrix by the Transformer Encoder

As shown in Figure 1, we use the transformer encoder to represent the text matrix (Vaswani et al., 2017). It is based on self-attention networks and each word is connected to any other word in the same sentence via self-attention which makes it possible to get rich information to predict domain-specific senses.

The encoder e typically stacks six identical layers. Each layer uses the multi-head attention and two sub-layers feed-forward network, combined with layer normalization and residual connection. For each word within a sentence, including the word itself, the multi-head attention computes attention weights, i.e., a softmax distribution shown in Eq. (1).

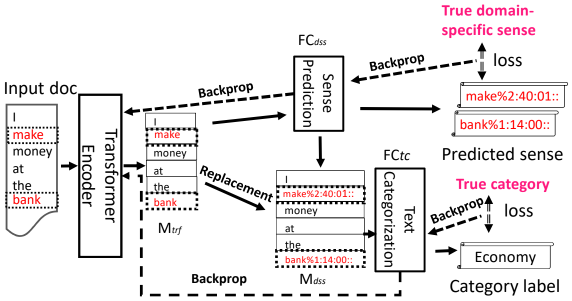


Figure 1: Multi-task Learning for Predominant Sense Prediction and Text Categorization: "make" and "bank" marked with red show the target word. "make%2:40:01::" and "bank%1:14:00::" show sense index obtained by the WordNet 2.0 and indicate the predominant sense of "make" and "bank" in the economy domain, respectively.

$$attention(\mathbf{Q},\mathbf{K},\mathbf{V}) \quad = \quad softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}. \quad (1)$$

The input are queries $\mathbf{Q}$, keys $\mathbf{K}$ of dimension d_k, and values $\mathbf{V}$ of dimension d_v. $\sqrt{d_k}$ refers to scaling factor. The inputs are linearly projected h times, in order to allow the model to jointly attend to information from different representation, concatinating the result,

$$multiHead(\mathbf{Q},\mathbf{K},\mathbf{V}) = Concat(\mathbf{head}_1,\cdots,\mathbf{head}_h)\mathbf{W}^{\mathbf{O}},$$

$$\text{where } \mathbf{head}_i = attention(\mathbf{Q}\mathbf{W}_i^{\mathbf{Q}},\mathbf{K}\mathbf{W}_i^{\mathbf{K}},\mathbf{V}\mathbf{W}_i^{\mathbf{V}}). \quad (2)$$

with parameter matrices $\mathbf{W}_i^{\mathbf{Q}} \in \mathbb{R}^{d_{model}\times d_k}$, $\mathbf{W}_i^{\mathbf{K}} \in \mathbb{R}^{d_{model}\times d_k}$, $\mathbf{W}_i^{\mathbf{V}} \in \mathbb{R}^{d_{model}\times d_v}$ and $\mathbf{W}^{\mathbf{O}} \in \mathbb{R}^{hd_v\times d_{model}}$. Here, d_{model} refers to the dimension of a word vector.

Let the output of $multiHead(\mathbf{Q},\mathbf{K},\mathbf{V})$ be $\mathbf{M}_{attn}$. On top of the multi-head attention, there is a feed-forward network that consists of two layers with a ReLU activation. Each encoder layer takes the output of the previous layer as input. It allows making attention to all positions of the previous layer. We obtain the output matrix $\mathbf{M}_{trf}$ shown in Figure 1 as an output of the encoder of the transformer.

2.2 Domain-Specific Sense Prediction

Each target word vector, i.e., the word which should be assigned a domain is extracted from the matrix $\mathbf{M}_{trf}$ and passed to the fully connected layer FC$_{dss}$. In Figure 1, "make" and "bank" denote the target words. The weighted matrix of FC$_{dss}$ is indicated as $\mathbf{W}_{dss} \in \mathbb{R}^{d_{model} \times d_{dss}}$ where d_{dss} is the number of the dimensions in the output which is equal to the number of domain-specific senses in all of the target words. The predicted sense vector $\mathbf{y}^{(dss)}$ is obtained as below:

$$\mathbf{y}^{(dss)} = softmax(\mathbf{M}_{trf} \cdot \mathbf{W}_{dss}). \tag{3}$$

We compute loss function by using $\mathbf{y}^{(dss)}$ and its true domain-specific sense vector $\mathbf{t}^{(dss)}$ which is represented as a one-hot vector. The loss function is defined by Eq. (4).

$$L_{dss}(\theta) = \begin{cases} -\frac{1}{n_{dss}} \sum_{i=1}^{n} \sum_{w=1}^{n_w} \sum_{s=1}^{d_{dss}} t_{iws}^{(dss)} \log(y_{iws}^{(dss)}) \\ \hfill (n_{dss} \geq 1), \\ 0 \hfill (n_{dss} = 0). \end{cases} \tag{4}$$

n refers to the minibatch size and n_w shows the number of words in a document. n_{dss} is the number of target words within the minibatch size and θ refers to the parameter used in the network. $t_{iws}^{(dss)}$ and $y_{iws}^{(dss)}$ show the value of the s-th domain-specific sense for the w-th target word in the i-th document within the minibatch size and its true value (1 or 0), respectively. As shown in Figure 1, we obtain text matrix $\mathbf{M}_{dss}$ by replacing each target vector ("make" and "bank") in the matrix $\mathbf{M}_{trf}$ to its domain-specific sense vector ("make%2:40:01::" and "bank%1:14:00::").

2.3 Text Categorization

We merged all the vectors of the matrix $\mathbf{M}_{dss}$ per dimension and obtained one document vector $\mathbf{D}_{sum}$. We passed it to the fully connected layers FC$_{tc}$. The number of the dimensions of the output vector $\mathbf{d}_{tc}$ obtained by FC$_{tc}$ equals to the total number of domains. Let the prediction vector $\mathbf{y}^{(tc)}$ be $\mathbf{W}_{tc} \times \mathbf{D}_{sum}$ where $\mathbf{W}_{tc} \in \mathbb{R}^{d_{model} \times d_{tc}}$ indicates the weight matrix of FC$_{tc}$. We applied softmax function for single label categorization task which is defined by:

$$\hat{p}_{ic}^{(tc)} = \frac{exp(y_{ic}^{(tc)})}{\sum_{c'=1}^{d_{tc}} exp(y_{ic}^{(tc')})} \tag{5}$$

Similarly, we used a sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ for multi-label categorization problem. The training objective is to minimize the following loss:

$$L_{tc}(\theta) = \begin{cases} -\frac{1}{n} \sum_{i=1}^{n} \sum_{c=1}^{d_{tc}} t_{ic}^{(tc)} \log(\hat{p}_{ic}^{(tc)}). \\ \hfill \text{Single-label} \\ -\frac{1}{n} \sum_{i=1}^{n} \sum_{c=1}^{d_{tc}} [t_{ic}^{(tc)} \log(\sigma(y_{ic}^{(tc)})) + \\ \quad (1 - t_{ic}^{(tc)}) \log(1 - \sigma(y_{ic}^{(tc)}))]. \\ \hfill \text{Multi-label} \end{cases} \tag{6}$$

Single-label and Multi-label in Eq. (6) denote the loss function for single-label and multi-label prediction, respectively. n refers to the minibatch size and θ shows parameter used in the network. $t_{ic}^{(tc)}$ and $y_{ic}^{(tc)}$ show the value of the c-th domain in the i-th document within the minibatch size and its true value (1 or 0), respectively.

In case of a single domain, a domain whose probability score is the maximum is regarded to the predicted domain. When the test data is the multi-label problem, we set a threshold value λ and domains whose probability score exceeds the threshold value are considered for selection.

2.4 Multi-task Learning

We assume that the auxiliary predominant sense prediction task helps the text categorization task by learning common feature representation of predominant senses for text categorization. The model adopts a multi-task objective function which is shown in Eq. (7). It is trained to simultaneously categorize texts and predicts a predominant sense for each word.

$$\begin{aligned} L^{(multi)}(\theta^{(sh)}, \theta^{(dss)}, \theta^{(tc)}) &= L^{(dss)}(\theta^{(sh)}, \theta^{(dss)}) \\ &\quad + L^{(tc)}(\theta^{(sh)}, \theta^{(tc)}) \end{aligned} \tag{7}$$

$\theta^{(sh)}$ in Eq. (7) refers to a shared parameter of the two tasks. $\theta^{(dss)}$ and $\theta^{(tc)}$ stand for a parameter estimated in domain-specific sense prediction and that of text categorization, respectively. Given a corpus, the parameters of the network are trained to minimize the value obtained by Eq. (7).

3 Experiments

3.1 Dataset

We performed the experiments on four benchmark datasets having domains to evaluate the properties

SFC	RCV1
Arts	Arts, Entertainment
Science	Science
Politics	Politics
Economy	Economics
Sports	Sports
Weather	Weather
Politics	Government
Industry	Corporate
Law	Law
Environment	Environment
Tourism	Travel
Military	War
Commerce	Market

Table 1: SFC and RCV1 correspondences

SFC	APW
Arts	Entertainment
Politics	Politics
Economy	Financial
Sports	Sports
Weather	Weather

Table 2: SFC and APW(AQUAINT) correspondences

of our framework: RCV1 (Lewis et al., 2004), 20 Newsgroups[1], 1999 APW[2] from the AQUAINT corpus[3], and AG's corpus of news articles[4].

The data for domain-specific sense prediction is based on the senses provided by the all-words task in SensEval-2 (Palmer et al., 2001) and SensEval-3 (Snyder and Palmer, 2004). Magnini et al (Magnini and Cavaglia, 2000; Magnini et al., 2002) created a lexical resource where WordNet 2.0 synsets were annotated with Subject Field Codes (SFC). Especially, 96% of WordNet synsets for nouns are annotated. We assigned each domain described in their SFC list to the sense of the all-words task in SensEval-2 and SensEval-3 data. Moreover, we assigned SFC labels to four benchmark datasets having domains. The SFC consists of 115,424 words assigning 168 domain labels which include some of the four datasets' domains. We manually corresponded these domains to SFC labels which are shown in Tables 1, 2, 3 [5], and 4.

The dataset statistics are summarized in Table 5 and examples of domain-specific sense-tagged data are shown in Table 6. RCV1 consists of 806,701 documents, one-year corpus from Aug 20th, 1996 to Aug 19th, 1997. RCV1 is a large volume of data compared to the other three data. We thus reserved eight months of the RCV1 data to learn word-embedding model. The model is also used for the other three datasets because they are the same genre as the RCV1, news stories. We divided the remaining data into three. The division is the same as the other three datasets: we reserved 60% of the data to train the models, 20% of the data is used for tuning hyperparameters, and the remaining 20% is used to test the models. All the documents are tagged by using Stanford CoreNLP Toolkit (Manning et al., 2014).

SFC	20News
Arts	Rec.autos, Rec.motercycles Rec.sport.baseball, Rec.sport.hockey
Science	Sci.crypt, Sci.electronics, Sci.med, Sci.space
Politics	Talk.politics.mis, Talk.politics.guns Talk.politics.mideast

Table 3: SFC and 20News correspondences: 20News contains seven top categories. Of these, we used three, each of which corresponds to SFC.

SFC	AG
Arts	Entertainment
Science	Science
Sports	Sports

Table 4: SFC and AG correspondences

3.2 Baselines

We compared our method to three baseline methods: (i) TRF-Single which is a text categorization based on the transformer but without domain-specific sense prediction, (ii) TRF-Sequential, a method first predicts domain-specific senses and then classify documents by using the result, and (iii) TRF-Delay-Multi, which is a model to start learning predominant sense model at first until the stable, and after that it adapts text categorization simultaneously. This is a mixed method of TRF-Sequential with fully separated training and TRF-Multi with fully simultaneously training. We compared our method with these approaches.

For multi-label text categorization by using RCV1 data, we chose XML-CNN as a baseline method because their method is simple but powerful and attained at the best or second best compared to the seven existing methods including Bow-CNN (Johnson and Zhang, 2015) on six

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

[2] We did not use 1998 and 2000 APW as the domains are not assigned to these data.

[3] http://catalog.ldc.upenn.edu/LDC2002T31

[4] https://github.com/mhjabreel/CharCnn_Keras/tree/master/data/ag_news_csv

[5] {"autos", "motorcycles"}, and "sport" are assigned to different SFC labels. However, we followed 20News categorization and grouped into one.

Datasets	N	D	L	W	S	$\hat{S}$	M	$\hat{M}$
RCV1	502,383	13	2.4	565	992	3,800,197	38,645	3,831
APW	46,032	5	1	397	586	877,400	9,206	1,497
20News	10,228	3	1	404	563	46,410	3,409	82
AG	95,700	3	1	390	562	124,885	31,900	222

Table 5: Data Statistics: N is the number of documents, D shows the number of domains, L is the average number of domains per document, W refers to the number of different target words, S is the number of different target senses, and $\hat{S}$ denotes the total number of target senses in the documents, M shows the average number of documents per domain, and $\hat{M}$ is the average number of documents per target sense.

Domain	Document
Arts	jonathan think there be a earlier russian film movie%1:10:00:: on tv just say it be base on a gogol .
Science	the usaf of this program%1:10:02:: be very open to ssato and will about 50m next year for study%1:09:03:: .
Politics	i do not think the suffering of some jew during wwius justify the commit by the israeli government%1:14:00:: .

Table 6: Sense-tagged training data (20News): Words marked with "%" indicates sense index obtained by the WordNet 2.0. Each word is lemmatized by using CoreNLP-Toolkit.

Hyperparameter	Value
The # of dimensions of a word vector (d_{model})	100
The # of epoch	100
Minibatch sizes (n)	32
Activation function	ReLu
Threshold value for Multi-label learning (λ)	0.5
Gradient descent	Adam

Table 7: Model settings: The hyperparameters commonly used in all of the method.

benchmark datasets where the label-set sizes are up to 670K (Liu et al., 2017a). Original XML-CNN is implemented by using Theano,[6] while we implemented our method by Chainer.[7] To avoid the influence of the difference in libraries, we implemented XML-CNN by Chainer and used it as a baseline. We followed the author-provided implementation in our Chainer's version of XML-CNN. To make a fair comparison, we used fastText (Joulin et al., 2017) as a word-embedding tool with all of the methods.

3.3 Model settings and evaluation metrics

The hyperparameters which are commonly used in all of the methods and their own estimated hyperparameters are shown in Tables 7 and 8, respectively[8]. These hyperparameters are optimized by using a hyperparameter optimization framework called Optuna[9]. They were independently determined for each dataset. In the experiments, we run five times for each model and obtained the averaged performance. We used standard recall, precision, and F1 measures. We further computed Macro-averaged F1 and Micro-averaged F1 and used them through the experiments.

3.4 Results

The performance of all methods in Micro-averaged F1 and Macro-averaged F1 on four datasets are summarized in Tables 9, and 10, respectively. Overall, both Micro and Macro-averaged F1 obtained by each method were very high except for the RCV1 data. Because these datasets consist of at most five domains and a single-label problem. The Micro and Macro-F1 obtained by TRF-Single were better than those obtained by XML-CNN except for APW corpus. This shows that text categorization based on the encoder of the transformer is effective for categorization. Sequential learning does not work well for text categorization. Because the average Macro-F1 obtained by TRF-Sequential (89.41%) was slightly worse than that of TRF-Single (89.74%), while Micro-averaged F1 obtained by TRF-Sequential (90.02%) was slightly better than TRF-Single (89.89%).

TRF-Delay-Multi was worse than TRF-Sequential. Especially, as shown in Tables 9 and 10, the results in RCV1 were worse than TRF-Single. One possible reason for the result is that predominant sense identification is more difficult task compared with text categorization. As shown in Table 5, for example, in RCV1, the average number of documents per target

[6] https://drive.google.com/file/d/1Wwy!MNkrJRXZM3WNZNywa94c2-iEh_6U/view

[7] https://chainer.org

[8] Our source code including Chainer's version of XML-CNN is available at: https://github.com/ShimShim46/TRF_Multitask

[9] https://github.com/pfnet/optuna

Data	XML-CNN			TRF-Single			TRF-Seq, TRF-Delay				TRF-Multi		
	fr	f	wd	h	e	wd	h	e	wd	ep	h	e	wd
RCV1	2, 3, 4	128	1.00×10^{-4}	10	1	1.00×10^{-4}	10	2	1.00×10^{-4}	75	10	1	1.00×10^{-4}
APW	1, 2, 3	256	1.18×10^{-10}	10	1	8.77×10^{-4}	10	1	4.39×10^{-4}	100	10	1	3.60×10^{-6}
20News	4, 5, 6	128	3.05×10^{-4}	5	1	1.42×10^{-10}	5	1	9.08×10^{-8}	75	10	1	4.39×10^{-8}
AG	3, 4, 5	256	4.15×10^{-4}	10	3	6.50×10^{-4}	10	2	2.00×10^{-4}	25	10	1	1.59×10^{-6}

Table 8: Model settings for each method: “TRF-Seq.” and “TRF-Delay” show TRF-Sequential and TRF-Delay-Multi, respectively. “fr” refers to filter region and “f” shows Filters. “wd” indicates Weight Decay. “h” shows multi-attention layers and “e” is a stack of encoders. “ep” refers to the number of epochs in the predominant sense prediction used in the baseline (iii). For instance, 75 indicates that we run predominant sense prediction task until 75 epochs, and then run multi-task learning.

	Methods				
Datasets	XML-CNN	TRF-Single	TRF-Sequential	TRF-Delay-Multi	TRF-Multi
RCV1	70.01	70.30	70.43	62.43	**71.92**
APW	98.96$*$	98.23	98.53	98.80$*$	**99.34**
20News	88.39	91.51	91.62	91.93$*$	**92.87**
AG	99.07	99.52$*$	99.52$*$	99.73$*$	**99.82**
Average	89.10	89.89	90.02	88.22	**90.98**

Table 9: Micro-averaged F1 (%): Bold font shows the best result with each line. The method marked with “$*$” indicates the score is not statistically significant compared to the best one. We used a t-test, p-value $<$ 0.05.

	Methods				
Datasets	XML-CNN	TRF-Single	TRF-Sequential	TRF-Delay-Multi	TRF-Multi
RCV1	56.59	70.03	68.52	62.43	**71.82**
APW	98.19	97.13	97.70	98.05	**99.14**
20News	88.04	**92.72**	91.94	91.60	92.62$*$
AG	96.61	99.08	99.51$*$	99.38$*$	**99.64**
Average	84.85	89.74	89.41	87.86	**90.80**

Table 10: Macro-averaged F1 (%): Bold font shows the best result with each line. The method marked with “$*$” indicates the score is not statistically significant compared to the best one. We used a t-test, p-value $<$ 0.05.

Datasets	TRF-Seq. TRF-Delay	TRF-Multi
RCV1	92.38	97.91
APW	95.51	98.82
20News	84.44	86.64
AG	91.26	92.03$*$
Average	90.90	93.85

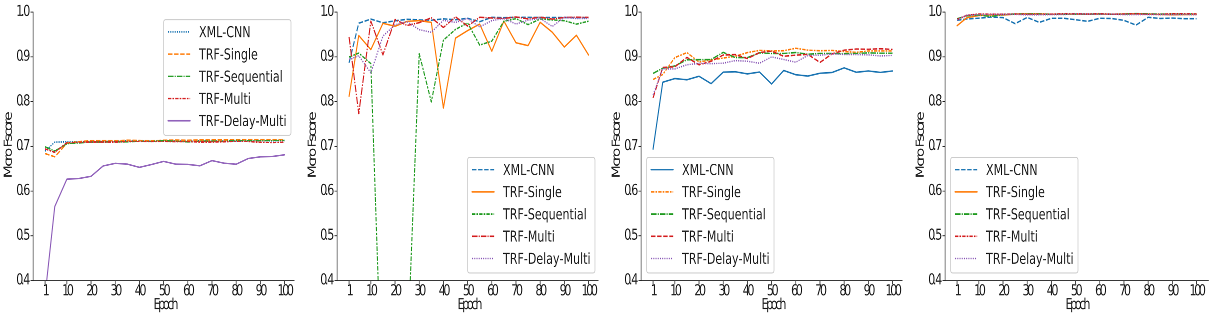
Table 11: Micro-averaged F1(%) of predominant sense prediction: The method marked with “$*$” indicates the score is not statistically significant compared to the best one. We used a t-test, p-value $<$ 0.05.

Datasets	TRF-Seq. TRF-Delay	TRF-Multi
RCV1	78.84	83.32
APW	75.38	79.70
20News	70.13	72.76
AG	77.54	80.73
Average	75.47	78.88

Table 12: Macro-averaged F1(%) of predominant sense prediction

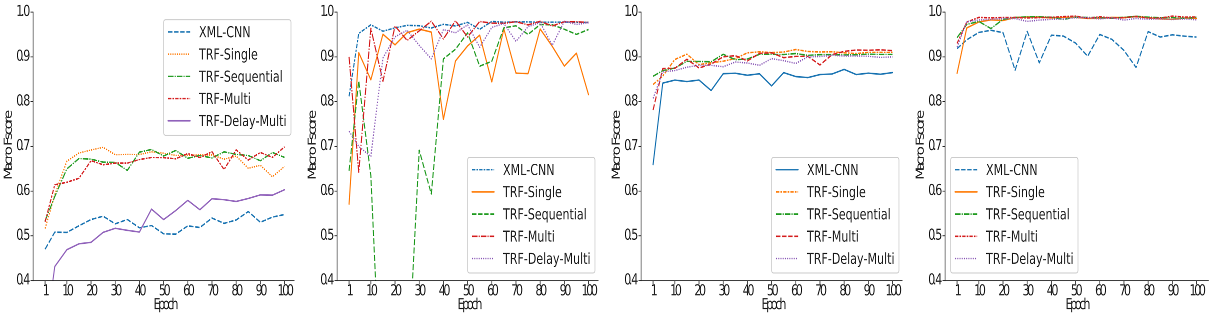
sense is 3,831, while the average number of documents per domain is 38,645. The training data for predominant senses is smaller than that of text categorization, which causes the overfitting problem. As a result, TRF-Delay-Multi does not work well and even worse than TRF-Single. This shows that separately learning predominant sense model at first until the stable, and after that, learning predominant sense prediction and text categorization simultaneously did not improve the overall performance.

Overall, the results obtained by TRF-Multi were the best among them by both Micro and Macro-averaged F1. This indicates that the predominant sense information through multi-task learning can help the model to learn better sense/document representations. On RCV1, the overall performance in each method was worse than those obtained by using other data as the categorization task is more difficult task compared with other data, i.e., multi-label problem. However, TRF-Multi is still better than other methods. The improvement was 1.49% $\sim$ 9.49% by Micro-F1 and 1.79% $\sim$ 15.23% by Macro-F1.



(a) Micro-F1 (RCV1) (b) Micro-F1 (APW) (c) Micro-F1 (20News) (d) Micro-F1 (AG)

Figure 2: Micro-F1 against the # of epochs obtained by using the test data: Multi-task learning stability.



(a) Macro-F1 (RCV1) (b) Macro-F1 (APW) (c) Macro-F1 (20News) (d) Macro-F1 (AG)

Figure 3: Macro-F1 against the # of epochs obtained by using the test data: Multi-task learning stability.

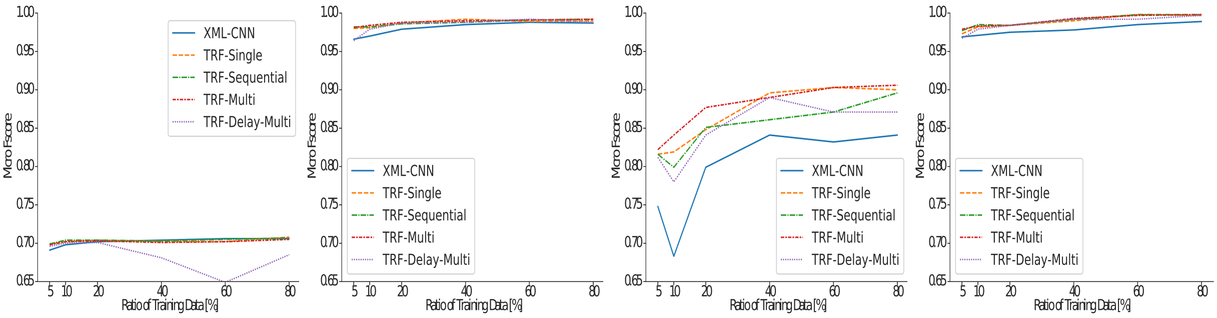
Tables 11 and 12 show the Micro and Macro-F1 of predominant sense prediction, respectively. The overall performance of multi-task learning was better to those of TRF-Seq. (TRF-Delay) by both measures except for Micro-F1 on AG data. This confirms our conjecture: to train the data in order to simultaneously categorize texts and predict domain-specific senses is effective for sense prediction.

Figures 2 and 3 show Micro and Macro-F1 against the number of epochs by using each of the four datasets. As we can see from these Figures, on 20News and AG corpus, each model except for XML-CNN are similar learning stability in both Micro and Macro-F1 curves. On RCV1, we have the same observation by Micro-F1 except for TRF-Delay-Multi and there is no significant difference in stability between TRF-Multi and TRF-Sequential by Macro-F1. On APW, TRF-Multi is similar to XML-CNN as they are stable after 60 epochs. In summary, TRF-Multi gets more stable through the datasets and in both measures.

We also examined the affection on each categorization performance by the ratio of predominant-sense tagged training data. For each domain and each predominant-sense, we count the total number of documents and obtained 5% to 80% of the training documents. The results by Micro and Macro-F1 are illustrated in Figures 4, and 5, respectively.

The Micro-F1 values except for 20News and for TRF-Delay-Multi on RCV1 are not a significant difference among methods and keep the performance until the ratio of training data decreased at 40%. Similarly, when the ratio is larger than 20%, the Macro-F1 on APW and AG obtained by all the methods do not differ significantly except for XML-CNN. The Micro and Macro-F1 curves obtained by 20news and Macro-F1 curve on RCV1 shows that more training data helps the performance. This is reasonable because the average number of training data per domain on 20news is 3,409 and it is extremely smaller than other datasets. RCV1 is also a multi-label problem.

The curves obtained by TRF-Multi drop slowly compared to other methods and it keeps the best performance by both evaluation measures and even in the ratio of 5%. From the observations,



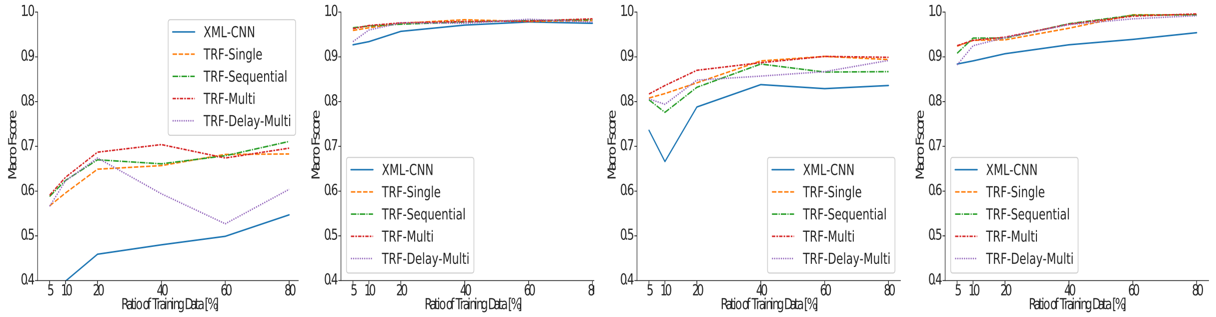
(a) Micro-F1 (RCV1)

(b) Micro-F1 (APW)

(c) Micro-F1 (20News)

(d) Micro-F1 (AG)

Figure 4: Micro-F1 against the ratio of training data



(a) Macro-F1 (RCV1)

(b) Macro-F1 (APW)

(c) Macro-F1 (20News)

(d) Macro-F1 (AG)

Figure 5: Macro-F1 against the ratio of training data

we can conclude that TRF-Multi learning model works well, especially in the cases that the number of training data per domain is small.

4 Related Work

Deep learning techniques have been great successes for automatically extracting context-sensitive features from a textual corpus. Many authors have attempted to apply deep learning methods including CNN (Kim, 2014; Zhang et al., 2015; Wang et al., 2015; Zhang and Wallace, 2015; Zhang et al., 2017; Wang et al., 2017), the attention based CNN (Yang et al., 2016), bag-of-words based CNN (Johnson and Zhang, 2015), and the combination of CNN and recurrent neural network (RNN) (Zhang et al., 2016) to text categorization. Most of these approaches demonstrated that neural network models are powerful for learning effective features from textual input. However, most of them for learning word vectors only allow a single context-independent representation for each word even if it has several senses. Peters et al. addressed the issue and proposed a model of deep contextualized word representation called ELMo derived from a bidirectional LSTM (Peters et al., 2018). They reported that their representation model significantly improves the state-of-the-art across six NLP problems. Similarly, Devlin proposed a model of deep contextualized word representation called BERT that can deal with syntax and semantics including polysemies (Devlin et al., 2018). Their methods attained amazing results in many NLP tasks. However, they do not explicitly map each sense of a word to its domain as their methods are unsupervised manner. Moreover, their model needs a large amount of corpus which leads to computational workload. Our model utilizes existing domain-specific senses (Magnini and Cavaglia, 2000; Magnini et al., 2002) as pseudo rough but explicit word representation data. It enables us to learn feature representations for both predominant senses and text categorization with a small amount of data.

Similar to the text categorization task, the recent upsurge of deep learning techniques have also contributed to improving the overall performance on Word Sense Disambiguation (WSD) (Yuan et al., 2016; Raganato et al., 2017; Peters et al., 2018). Melamud et al. proposed a method called Context2Vec which learns each sense annotation in the training data by using a bidirectional LSTM trained on an unlabeled corpus (Melamud et al., 2016). More recently, Vaswani et al. introduced the first full-attentional architecture called Transformer. It utilizes only the self-attention mechanism and demonstrated its effectiveness on neural machine translation. Since then, the transformer has been successfully applied to many NLP tasks including semantic role labeling (Strubell et al., 2018) and sentiment analysis (Ambartsoumian and Popowich, 2018). To the best of our knowledge, this is the first approach for predicting domain-specific senses based on a transformer that is trained with multi-task learning.

In the context of predominant sense prediction, several authors have attempted to use domain-specific knowledge to disambiguate senses and show that the knowledge outperforms generic supervised WSD (Agirre and Soroa, 2009; Faralli and Navigli, 2012; Taghipour and Ng, 2015). McCarthy et al. proposed a statistical method for assigning predominant noun senses (McCarthy et al., 2004, 2007). They find words with a similar distribution to the target word from parsed data. They tested 38 words containing two domains of Sports and Finance from the Reuters corpus (Rose et al., 2002). Similarly, Lau et al. (2014) proposed a fully unsupervised topic modeling-based approach to sense frequency estimation. Faralli and Navigli (2012) attempted to performing domain-driven WSD by a pattern-based method with minimally-supervised framework. While conceptually similar, our model differs from these approaches in that it is supervised learning by adopting existing domain-specific sense tags for creating the data.

In the context of multi-task learning, many authors have attempted to apply it to NLP tasks (Collobert and Weston, 2008; Glorot et al., 2011; Liu et al., 2015, 2016). Liu et al. proposed adversarial multi-task learning which alleviates the shared and private latent feature spaces from interfering with each other (Liu et al., 2017b). Xiao et al. attempted multi-task CNN which introduces a gate mechanism to reduce the interference (Xiao et al., 2018). They reported that their approach can learn selection rules automatically and gain a great improvement over baselines through the experiments on nine text categorization datasets. Both of them focused on text categorization task only as a multi-task and used the word embeddings which are initialized with Word2Vec or GloVe vectors. Aiming at text categorization with relatively small amounts of training data, we demonstrated a predominant sense of a word is effective for text categorization in the framework of multi-task learning with domain-specific sense identification and text categorization. This enabled us to obtain better explicit feature representations to classify documents.

5 Conclusion

We have presented an approach to text categorization by leveraging a predominant sense of a word depending on the domain. We empirically examined that predominant sense identification helps to improve the overall performance of text categorization in the framework on multi-task learning. The comparative results with the baselines showed that our model is competitive as the improvement was 1.49% $\sim$ 9.49% by Micro-F1 and 1.79% $\sim$ 15.23% by Macro-F1. Moreover, we found that our model works well, especially for the categorization of documents with multi-label.

Future work will include: (i) incorporating lexical semantics such as named entities for further improvement, (ii) comparing our model to other deep contextualized word representation such as ELMO and BERT, and (iii) applying the method to other domains for quantitative evaluation.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions. We also thank Dr. Bernardo Magnini who provided us WORDNET-DOMAINS-3.2 data. This work was supported by the Grant-in-aid for JSPS, Grant Number 17K00299, and Support Center for Advanced Telecommunications Technology Research, Foundation.

References

E. Agirre and A. Soroa. 2009. Personalizing Pagerank for Word Sense Disambiguation. In *Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41.

A. Ambartsoumian and F. Popowich. 2018. Self-Attention: A Better Building Block for Sentiment Analysis Neural Network Classifiers. In *Proc. of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 130–139.

R. Collobert and J. Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proc of the 25th International Conference on Machine Learning (ICML)*, pages 160–167.

J. Devlin, M-W. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *arXiv:1810.04805*.

B. P. Douglas and M. B. Janet. 1992. The Design for Wall Street Journal-based CSR Corpus. In *Proc of the HLT'91 Workshop on Speech and Natural Language*, pages 357–362.

S. Faralli and R. Navigli. 2012. A New Minimally-Supervised Framework for Domain Word Sense Disambiguation. In *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1411–1422.

X. Glorot, A. Bordes, and Y. Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *Proc of the 28th Ingernational Conference on Machine Learning*, pages 513–520.

P. Jin, D. McCarthy, R. Koeling, and J. Carroll. 2009. Estimating and Exploiting the Entropy of Sense Distributions. In *Proc of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT) 2009*, pages 233–236.

R. Johnson and T. Zhang. 2015. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In *Proc of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proc. of the 15th Conference of the European Chapter of the Association for Conputational Linguistics*, pages 427–431.

Y. Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

J. H. Lau, P. Cook, D. McCarthy, S. Gella, and T. Baldwin. 2014. Learning Word Sense Distribution, Detecting Unattested Senses and Identifying Novel Senses using Topic Models. In *Proc of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 259–270.

D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397.

J. Liu, W-C. Chang, Y. Wu, and Y. Yang. 2017a. Deep Learning for Extreme Multi-label Text Classification. In *Proc of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.

P. Liu, X. Qiu, and X. Huang. 2017b. Adversarial Multi-Task Learning for Text Classification. In *Proc of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1–10.

P. Liu, X. Qiu, and Z. Huang. 2016. Recurrent Neural Network for Text Classification with Multi-task Learning. In *Proc of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, pages 2873–2879.

X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y-Y. Wang. 2015. Representation Learning using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. In *Proc of the 2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–921.

B. Magnini and G. Cavaglia. 2000. Integrating Subject Field Codes into WordNet. In *Proc. of the International Conference on Language Resources and Evaluation*, pages 1413–1418.

B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. 2002. The Role of Domain Information in Word Sense Disambiguation. *Natural Language Engineering*, 8:359–373.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. 2014. The Stanford Core NLP Natural Language Processing Toolkit. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. Finding Predominant Word Senses in Untagged Text. In *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 279–286.

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2007. Unsupervised Acquisition of Predominant Word Senses. *Computational Linguistics*, 34(4):553–590.

O. Melamud, J. Goldberger, and I. Dagan. 2016. Context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proc. of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proc. of the International Conference on Learning Representations Workshop*.

M. Palmer, C. Cotton, S. L. Delfs, and H. T. Dang. 2001. English Tasks: All-Words and Verb Lexical Sample. In *Proc. of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems, Association for Computational Linguistics*, pages 21–24.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proc. of the 16th Anual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237.

A. Raganato, C. D. Bovi, and R. Navigli. 2017. Neural Sequence Learning Models for Word Sense Disambiguation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167.

T. Rose, M. Stevenson, and M. Whitehead. 2002. The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources. In *Proc. of Language Resources and Evaluation*.

B. Snyder and M. Palmer. 2004. The English All-Words Task. In *Proc. of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Association for Computational Linguistics*, pages 41–43.

E. Strubell, P. Verga, D. Andor, D. Weiss, and A. McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

K. Taghipour and H. T. Ng. 2015. Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. In *Proc. of the 2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 314–323.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention Is All You Need. In *Proc. of the NIPS*.

J. Wang, Z. Wang, D. Zhang, and J. Yan. 2017. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In *Proc. of the 26th International Joint Conference on Artificial Intelligence*, pages 2915–2921.

P. Wang, J. Xu, B. Xu, C-L. Liu, H. Zhang, F. Wang, and H. Hao. 2015. Semantic Clustering and Convolutional Neural Network for Short Text Categorization. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 352–357.

L. Xiao, H. Zhang, and W. Chen. 2018. Gated Multi-Task Network for Text Classification. In *Proc. of the 2018 Annual conference of the North American Chapter of the Association for Computational Linguistics*, pages 726–731.

Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 1480–1489.

D. Yuan, J. Richardson, R. Doherty, C. Evans, and E. Altendorf. 2016. Semi-Supervised Word Sense Disambiguation with Neural Models. In *Proc. of the 26th International Conference on Computational Linguistics*, pages 1374–1385.

R. Zhang, H. Lee, and D. Radev. 2016. Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 1512–1521.

X. Zhang, J. Zhao, and Y. LeCun. 2015. Character-Level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing systems*, pages 649–657.

Y. Zhang, M. Lease, and B. C. Wallace. 2017. Exploiting Domain Knowledge via Grouped Weight Sharing with Application to Text Categorization. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 155–160.

Y. Zhang and B. C. Wallace. 2015. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *Computing Research Repository*.