EnsLM: Ensemble Language Model for Data Diversity by Semantic Clustering

Zhibin Duan*[1], Hao Zhang*†[1], Chaojie Wang[1], Zhengjue Wang[1], Bo Chen†[1], Mingyuan Zhou[2]

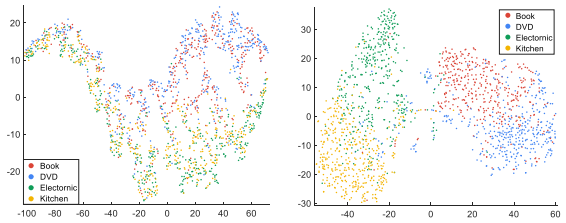
[1]National Laboratory of Radar Signal Processing, Xidian University, Xi'an, China
[2]McCombs School of Business The University of Texas at Austin, Austin, TX 78712, USA
{xd_zhibin, zhanghao_xidian}@163.com
bchen@mail.xidian.edu.cn, Mingyuan.Zhou@mccombs.utexas.edu

Abstract

Natural language processing often faces the problem of data diversity such as different domains, themes, styles and so on. Therefore, a single language model (LM) is insufficient to learn all knowledge from diverse samples. To solve this problem, we firstly propose an autoencoding topic model with mixture prior (mATM) to perform clustering for the data, where the clusters defined in semantic space describe the data diversity. Having obtained the clustering assignment for each sample, we develop the ensemble LM (EnsLM) with the technique of weight modulation. Specifically, EnsLM contains a backbone which is adjusted by a few modulated weights to fit for different sample clusters. As a result, the backbone learns the shared knowledge among all clusters while modulated weights extract the cluster-specific features. EnsLM can be trained jointly with mATM with flexible LM backbone. We evaluate the effectiveness of both mATM and EnsLM on different language understanding and generative tasks.

1 Introduction

It is common knowledge in modern natural language processing (NLP) that natural language varies greatly across domains, themes, styles, genres and many other linguistic nuances (Van der Wees et al., 2015; van der Wees, 2017; Niu et al., 2017). Generally, we call such nature of language as data diversity. Many existing works (Liu et al., 2017; Cai and Wan, 2019; Hu et al., 2019) have illustrated that data diversity will affect the performance of LMs if we just train a single LM over the entire dataset, even though fine-tuning a pre-trained LM (that has been pre-training on a very large corpus) such as Bert (Devlin et al., 2019) on current task (Aharoni and Goldberg, 2020).



(a) LDA (b) mATM

Figure 1: The distribution of samples on semantic space on 4 domains (different products) of Amazon dataset. The sample clustering characteristics of mATM can reflect the data diversity (domain in this example) in the corpus.

The domain diversity in dataset is a very common type of data diversity. In some cases, if we can obtain a well-defined domain label for each sample, some works (Jiang et al., 2020; Du et al., 2020; Wright and Augenstein, 2020) try to consider the multi-domain property of data in developing the LMs. However, these pre-defined domain labels are not always accurate or even available (Aharoni and Goldberg, 2020), especially for the wild datasets, in which data come from different sources, such as internet news, product reviews, and daily conversation. To this end, we hope to develop a LM that can explore the diversity from data automatically.

Data selection is a commonly used strategy to handle diversity in data (Moore and Lewis, 2010; Axelrod et al., 2011; Duh et al., 2013; Silva et al., 2018; Aharoni and Goldberg, 2020). This kind of method is developed from an assumption that samples belonging to the same cluster should own similar characteristics. According to the clustering assignment, models can select suitable data for training a LM for each cluster separately. Although, to some extend, data selection is an efficient strategy to alleviate the problem of data diversity, it may bring two disadvantages as follows. Firstly, the process of data selection is independent of the LM learning. In other words, the gradient signal generated by LM's training loss can not affect the

* Equal contribution. † Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 2954–2967
August 1–6, 2021.

data selection. Secondly, data selection only tells the hard cluster belongings of samples, ignoring a fact that some samples may belong to more than one clusters with soft (weighted) assignment.

Inspired by their works and to move beyond, in this paper, we find the semantics learned by topic modeling (Blei et al., 2003; Srivastava and Sutton, 2017) can infer sample clusters to a certain extent via K-means, but is not good enough, as shown in Fig. 1a . To jointly consider the clustering and topic modeling for better clustering (as shown in Fig. 1b) and for joint training with the following LM, we firstly introduce an autoencoding topic model with mixture priors (mATM). For each sample in the corpus, mATM can infer a soft clustering assignment. In order to jointly consider the learning of mATM with various LMs, we employ the weight modulation methods (Cong et al., 2020; Wen et al., 2020). Specifically, as shown in Fig. 3, given a LM as backbone, for each layer (convolutional or fully-connected), we introduce some modulated parameters. Guided by clustering assignment inferred from mATM, these parameters modulate the backbone single LM to multiple LMs, corresponding to different clusters. Therefore, our proposed model can be seen as a type of ensemble learning, and hence we call it ensemble language model (EnsLM).

Our proposed mATM and EnsLM enjoy the following distinguished properties:

- The mATM learns the mixture-prior latent semantic space to define a soft clustering assignment for each sample.
- Guided by clustering assignments that describe the data diversity, EnsLM learns both shared and cluster-specific knowledge by weight modulations.
- Joint training of mATM and EnsLM improves the performance of both on many NLP tasks.

2 Related work

For NLP, topic modeling (TM) (Blei et al., 2003; Zhou et al., 2012) and LMs are two common regimes with their own advantages. TM can discover the interpretable global semantics that are topics, while with pre-training on large corpus, LMs recently achieve the SOTA performance on many NLP tasks with more focuses on local dependencies. Therefore, some works consider to combine them to obtain benefits from both. Dieng et al. (2016) and Wang et al. (2020) incorporate the TM with RNN-based model to capture the long-range dependencies. To move beyond single-layer TM for RNNs, Guo et al. (2020) propose the recurrent hierarchical topic-guided RNN with the help of multi-layer TM (Zhou et al., 2015; Zhang et al., 2018). To extract explicit document semantics for summarization, Wang et al. (2020) propose three different modules to plug knowledge from TM into Transformer-based LMs (Vaswani et al., 2017; Devlin et al., 2018). Our work can be seen as a parallel work to combine their advantages together but focuses on dealing with data diversity in NLP without the ground-truth information such as domain labels. Meanwhile, our work can be applied for different LMs including CNNs, RNNs, and Transformer-based models.

3 Autoencoding topic model with mixture prior

We firstly describe one of the most popular topic models, latent Dirichlet allocation (LDA) (Blei et al., 2003), and its autoencoding inference (Srivastava and Sutton, 2017). Inspired by them, in order to jointly consider topic learning and sample clustering, we propose the autoencoding topic model with mixture prior (mATM).

3.1 LDA with autoencoding inference

For a document containing D words as $\boldsymbol{w} = \{w_d\}_{d=1}^{D}$, given K topics $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_K]$ where $\boldsymbol{\phi}_k$ is a probability distribution over the vocabulary, LDA defines the generative process of $\boldsymbol{w}$ in Algorithm 1, where $\boldsymbol{\theta} \in \mathbb{R}_+^K$ is the topic proportion with $\boldsymbol{\alpha}$ as the prior parameter. After collapsing

Algorithm 1 Generative process of LDA

for each document $\boldsymbol{w}$ **do**
  Draw topic proportion $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$
  for each word at position d **do**
    Sample a topic $i_d \sim \text{Multinomial}(1, \boldsymbol{\theta})$
    Sample a word $w_d \sim \text{Multinomial}(1, \boldsymbol{\phi}_{i_d})$

i_d, given $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$, we can represent the conditional likelihood of w_d as

$$w_d|\boldsymbol{\Phi}, \boldsymbol{\theta} \sim \text{Multinomial}(1, \boldsymbol{\Phi}\boldsymbol{\theta}). \quad (1)$$

Given $\boldsymbol{\Phi}$, a popular approximation for efficient inference of LDA is mean-field variational inference, which tries to maximize the evidence lower

bound (ELBO) of marginal data log likelihood as

$$\text{ELBO} = \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\boldsymbol{w}|\boldsymbol{\theta}, \boldsymbol{\Phi})] - KL[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})], \tag{2}$$

where $q(\boldsymbol{\theta})$ is the variational posterior. In particular, Srivastava and Sutton (2017) propose the autoencoding variational inference (AEVB) (Kingma and Welling, 2013) for LDA by using Laplace approximation (Hennig et al., 2012) for the Dirichlet prior, and building logistic-normal (LN) encoding posterior.

As shown in Fig. 1, we find that running clustering method such as K-means on semantic space $\boldsymbol{\theta}$ can not achieve satisfactory results. For jointly considering the learning of topics and sample clustering, we propose the mATM.

3.2 Generative process of mATM

Suppose the number of clusters is C, and the clustering prior parameter is $\boldsymbol{\pi} = [\pi_1, \cdots, \pi_C]$ with $\sum_{c=1}^{C} \pi_c = 1$, shown in Fig. 2a, mATM defines the generative process of $\boldsymbol{w}$ in Algorithm 2. Compared with LDA, mATM has a mixture Dirichlet prior with parameters $\{\boldsymbol{\alpha}^c\}_{c=1}^{C}$. In other words, mATM assumes that the $\boldsymbol{\theta}$ of different documents may come from different clusters, which is the basic thought to discover the data diversity from corpus automatically.

Algorithm 2 Generative process of mATM

for each document $\boldsymbol{w}$ **do**
  Draw cluster index $z \sim \text{Categorical}(\boldsymbol{\pi})$
  Draw topic proportion $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha}^z)$
  for each word at position d **do**
    Sample a topic $i_d \sim \text{Multinomial}(1, \boldsymbol{\theta})$
    Sample a word $w_d \sim \text{Multinomial}(1, \boldsymbol{\phi}_{i_d})$

3.3 Variational encoder of mATM

In order to infer the parameters in mATM and further develop the EnsLM by mATM, we introduce AEVB for mATM, whose detailed structure is shown in Fig. 2b.

3.3.1 Laplace approximation for mixture Dirichlet prior

Although Dirichlet prior of $\boldsymbol{\theta}$ is important to learn interpretable topics (Wallach et al., 2009), it is difficult to handle it within AEVB since AEVB needs effective reparameterization (RT) function for distributions. Inspired by the success of the Laplace approximation for Dirichlet distribution, we propose the mixture LN (mLN) distribution as the approximation of mixture Dirichlet distribution.

Specifically, Srivastava and Sutton (2017) have proved that a Dirichlet distribution $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$ can be well approximated by LN distribution as

$$p(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{LN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{3}$$

where the elements in mean vector $\boldsymbol{\mu}$ and diagonal covariance matrix $\boldsymbol{\Sigma}$ are

$$\begin{aligned} \mu_k &= \log \alpha_k - \frac{1}{K}\sum_{i=1}^{K} \log \alpha_i \\ \Sigma_k &= \frac{1}{\alpha_k}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_{i=1}^{K}\frac{1}{\alpha_i}. \end{aligned} \tag{4}$$

To go further, for inference of mATM, we construct the mLN distribution as

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_{c=1}^{C} \pi_c \mathcal{LN}(\boldsymbol{\mu}^c, \boldsymbol{\Sigma}^c) \\ \mu_k^c &= \log \alpha_k^c - \frac{1}{K}\sum_{i=1}^{K} \log \alpha_i^c \\ \Sigma_k^c &= \frac{1}{\alpha_k^c}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_{i=1}^{K}\frac{1}{\alpha_i^c}, \end{aligned} \tag{5}$$

which is used to approximate the mixture Dirichlet prior $p(\boldsymbol{\theta}|\{\boldsymbol{\alpha}^c, \pi_c\}_{c=1}^{C})$ in mATM. Therefore, for each document, the prior of $\boldsymbol{\theta}$ can be written as $\prod_{c=1}^{C} \mathcal{LN}(\boldsymbol{\mu}^c, \boldsymbol{\Sigma}^c)^{z_c}$. In practice, we build the $\boldsymbol{\mu}^c$ and $\boldsymbol{\Sigma}^c$ as

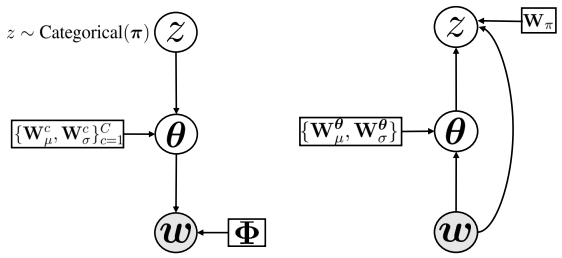
$$\boldsymbol{\mu}^c = f_{\mathbf{W}_\mu^c}(\boldsymbol{z}), \boldsymbol{\Sigma}^c = f_{\mathbf{W}_\sigma^c}(\boldsymbol{z}), \tag{6}$$

where $\boldsymbol{z} = [z_1, \cdots, z_C]$. Next, we build variational posterior for latent variables with easy RT function.

3.3.2 Variational encoding posterior

After collapsing $\{\boldsymbol{i}_d\}_{d=1}^{D}$ in mATM as (1) in LDA, given topics $\boldsymbol{\Phi}$, for document $\boldsymbol{w}$, there are two latent variables that need to be inferred: $\boldsymbol{\theta}$ and $\boldsymbol{z}$.

LN posterior for $\boldsymbol{\theta}$. We build the variational posterior of $\boldsymbol{\theta}$ as LN distribution $q(\boldsymbol{\theta}) = \mathcal{LN}(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$ with $\boldsymbol{\mu}' = f_{\mathbf{W}_\mu^{\boldsymbol{\theta}}}(\boldsymbol{x})$, $\boldsymbol{\Sigma}' = diag(f_{\mathbf{W}_\sigma^{\boldsymbol{\theta}}}(\boldsymbol{x}))$, where $diag$ converts a vector to a diagonal matrix, $f_{\mathbf{W}_\mu^{\boldsymbol{\theta}}}(\cdot)$ and $f_{\mathbf{W}_\sigma^{\boldsymbol{\theta}}}(\cdot)$ are two encoding networks, and $\boldsymbol{x}$ is a type of representation for document $\boldsymbol{w}$ such as original words or bag of words (Bow) vector. Morevoer, LN distribution has easy RT function as Normal distribution.



(a) Generation in mATM (b) Inference in mATM

Figure 2: Graphical model for the mATM, where the circle with white color, the circle with gray color and the rectangle denotes local latent variables, observations, and global parameters in mATM.

Gumbel softmax (GS) posterior for z. As categorical variable, z is difficult to build variational posterior under AEVB with accurate RT function. Instead, we employ GS distribution (Jang et al., 2016) as the variational posterior of z for efficient gradient propagation.

Specifically, suppose the posterior of z is Categorical($\boldsymbol{\pi}'$), after obtaining C i.i.d samples $\{g_1, \cdots, g_C\}$ drawn from Gumbel$(0, 1)$, then z can be sampled as

$$z = \arg\max_c \frac{\exp\left((\log(\pi'_c) + g_c)/\tau\right)}{\sum_{o=1}^{O} \exp\left((\log(\pi'_o) + g_o)/\tau\right)} \quad (7)$$

where τ is the temperature parameter. In order to build encoder for $\boldsymbol{\pi}'$, we let $\boldsymbol{\pi}' = f_{\mathbf{W}_\pi}(\boldsymbol{\theta}, \boldsymbol{w})$. For efficient gradient propagation, rather than sampling z from $\arg\max$ as (7), we obtain the variational posterior of soft assignment vector $\boldsymbol{z} = [z_1, \cdots, z_C]$ as $q(\boldsymbol{z})$:

$$[q(\boldsymbol{z})]_c = \frac{\exp\left((\log(\pi'_c) + g_c)/\tau\right)}{\sum_{o=1}^{O} \exp\left((\log(\pi'_o) + g_o)/\tau\right)}. \quad (8)$$

Besides the benefit of efficient gradient back-propagation, the soft assignment in (8) provides clustering belonging weights. In the following EnsLM, this property is useful for some ambiguous samples that may belong to different clusters.

3.3.3 ELBO of mATM

We obtain the ELBO of mATM as

$$\begin{aligned} \text{ELBO} = &\mathbb{E}_{q(\boldsymbol{\theta})q(\boldsymbol{z})}[\log p(\boldsymbol{w}|\boldsymbol{\theta}, \boldsymbol{\Phi}, \boldsymbol{z})] \\ &- KL[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{z})] - KL[q(\boldsymbol{z})|p(\boldsymbol{z}|\boldsymbol{\pi})] \end{aligned} \quad (9)$$

Similarly with Srivastava and Sutton (2017), instead of sampling $\boldsymbol{\Phi}$ from Dirichlet posterior in LDA, we parameterize it as $\boldsymbol{\Phi} = softmax(\mathbf{W}_t)$, where $\mathbf{W}_t = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K]$ and $softmax$ is operated for each topic $\{\boldsymbol{w}_k\}_{k=1}^K$ to ensure them on a probability simplex. Therefore, as shown in Fig. 2, all the parameters of mATM are $\boldsymbol{\Theta}_1 = \{\mathbf{W}_\mu^{\boldsymbol{\theta}}, \mathbf{W}_\mu^c, \mathbf{W}_\sigma^{\boldsymbol{\theta}}, \mathbf{W}_\sigma^c, \mathbf{W}_\pi, \mathbf{W}_t\}$ that can be learned by maximizing the ELBO in (9).

4 Ensemble language model

Recently, various advanced LMs for language understanding and generation have been introduced, most of which do not consider the data diversities in the corpus. In this paper, having obtained the clustering assignment vector $\boldsymbol{z}$ from mATM, given a single LM as backbone, we propose the ensemble LM (EnsLM) via $\boldsymbol{z}$-guided weight modulation. In other words, the EnsLM can modulate the backbone single LM to fit for different clusters.

4.1 Efficient weight modulation

Although LMs have many different types, basically, all of them build on convolutional (such as in CNN (Johnson and Zhang, 2015)) or fully-connected (such as in Transformer (Vaswani et al., 2017)) operations (ignoring the bias) as

$$\begin{aligned} \text{Convolution}: \mathbf{H}_2 &= f(\mathbf{W} * \mathbf{H}_1) \\ \text{Fully-connection}: \mathbf{H}'_2 &= f(\mathbf{W}'^T \mathbf{H}'_1). \end{aligned} \quad (10)$$

where, $\mathbf{H}_1 \in \mathbb{R}^{I_x \times I_y \times C_{in}}$ and $\mathbf{H}'_1 \in \mathbb{R}^{C_{in}}$ are the input features, $\mathbf{W} \in \mathbb{R}^{k_x \times k_y \times C_{in} \times C_{out}}$ and $\mathbf{W}' \in \mathbb{R}^{C_{in} \times C_{out}}$ are the convolutional kernel or full-connected weights[1]. Suppose the number of clusters (domains) in mATM is C, given a LM as backbone, we introduce a few modulation parameters to modulate the original parameters $\mathbf{W}$ or $\mathbf{W}'$ for different clusters.

Specifically, shown in Fig. 3, for a convolutional or fully-connected layer in (10), suppose that there are two dictionaries of modulation parameters as:

$$\begin{aligned} \mathbf{A} &= [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_C] \in \mathbb{R}^{C_{in} \times C} \\ \mathbf{B} &= [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_C] \in \mathbb{R}^{C_{out} \times C}, \end{aligned} \quad (11)$$

where $\{\alpha_c\}_{c=1}^C \in \mathbb{R}^{C_{in}}$ and $\{\boldsymbol{\beta}_c\}_{c=1}^C \in \mathbb{R}^{C_{out}}$. For a document $\boldsymbol{w}$ whose feature at current layer is $\mathbf{H}_1$, after archiving its domain assignment $\boldsymbol{z} \in \mathbb{R}^{C \times 1}$

[1]Fully-connected layer can be also seen as a convolution layer where the convolutional kernel is $\mathbf{W}' \in \mathbb{R}^{1 \times 1 \times C_{in} \times C_{out}}$ ($I_x = I_y = 1$)

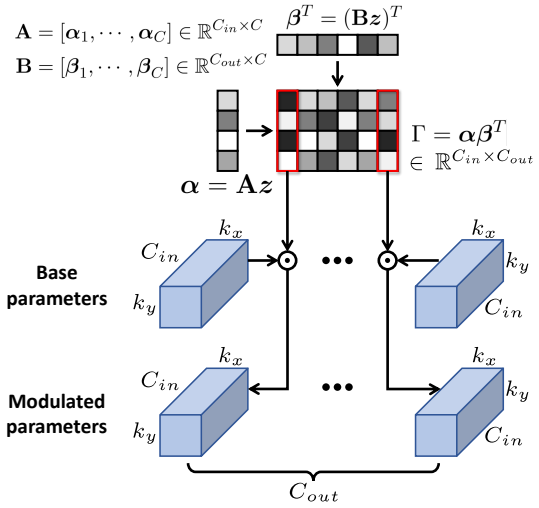


Figure 3: Illustration of weight modulation in EnsLM.

from (8), we feed $\mathbf{H}_1$ into the modulated layer as

$$\begin{aligned} \text{Convolution}: \mathbf{H}_2 &= f((\mathbf{W} \odot \Gamma) * \mathbf{H}_1) \\ \text{Fully-connection}: \mathbf{H}_2' &= f((\mathbf{W}'^T \odot \Gamma)\mathbf{H}_1'), \end{aligned} \tag{12}$$

where $\Gamma = \boldsymbol{\alpha}\boldsymbol{\beta}^T$, $\boldsymbol{\alpha} = \mathbf{A}\boldsymbol{z} \in \mathbb{R}^{C_{in} \times 1}$, $\boldsymbol{\beta} = \mathbf{B}\boldsymbol{z} \in \mathbb{R}^{C_{out} \times 1}$, and $\odot$ denotes matrix element-wise product (with broadcasting for convolution).

Explanation of (12). Intuitively, $\mathbf{W}$ and $\mathbf{W}'$ act as the backbone parameters in the original single LM, and $\boldsymbol{\Gamma}$ is the modulated parameters, which moves the backbone to fit different domains. If z is drawn from (7) that means $\boldsymbol{z}$ is a one-hot vector, then it denotes that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are chosen from the dictionaries $\mathbf{A}$ and $\mathbf{B}$, correspondingly. If $\boldsymbol{z}$ is drawn from (8) that means $\boldsymbol{z}$ is a soft assignment vector, then it denotes that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are weighted summation of all elements in $\mathbf{A}$ and $\mathbf{B}$, correspondingly. In practice, we use the soft assignment vector since i) it brings efficient gradient propagation during joint training of mATM and EnsLM, and ii) it considers the fact that there are some domain ambiguous samples in the dataset.

It is interesting to note that although EnsLM is developed for the problem that ground-truth priors of data diversity (such as domain label) is unavailable, it can be also used when we know the priors. For this scenario, rather than inferring the clustering assignment $\boldsymbol{z}$ from mATM via (8), we directly set $\boldsymbol{z}$ as the real one-hot assignment vector, which is illustrated in experiment in Sec. 5.2.

4.2 Joint training of mATM and EnsLM

Different from some strategies such as data selection that separate the calculation of assignment and the training of LM, our proposed mATM and EnsLM can be jointly trained in one framework.

Specifically, given a training set containing N sample $\{\boldsymbol{w}_n\}_{n=1}^N$, suppose that there is a label $\{\boldsymbol{y}_n\}_{n=1}^N$ for each sample. It should be noted that labels $\{\boldsymbol{y}_n\}_{n=1}^N$ can be different for different tasks, such as labels for document classification, golden summarization for abstractive summarization, or document itself for generation. As a result, the loss for joint training of mATM and EnsLM can be written as

$$\begin{aligned} \mathcal{L} = &\sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta}_n)q(\boldsymbol{z}_n)}[\log p(\boldsymbol{w}_n|\boldsymbol{\theta}_n, \boldsymbol{\Phi}, \boldsymbol{z}_n)] \\ &- \mathbb{E}_{q(\boldsymbol{z}_n)}[\mathcal{L}_{LM}(\boldsymbol{w}_n, \boldsymbol{y}_n, \boldsymbol{z}_n)] \\ &- KL[q(\boldsymbol{\theta}_n)||p(\boldsymbol{\theta}_n)] - KL[q(\boldsymbol{z}_n)|p(\boldsymbol{z}_n)], \end{aligned} \tag{13}$$

where, without loss of generality, $\mathcal{L}_{LM}$ denotes the loss for LM. All learnable parameters are i) parameters of mATM: $\boldsymbol{\Theta}_{mATM} = \{\mathbf{W}_\mu^{\boldsymbol{\theta}}, \mathbf{W}_\sigma^{\boldsymbol{\theta}}, \mathbf{W}_\mu^{\boldsymbol{u}}, \mathbf{W}_\sigma^{\boldsymbol{u}}, \mathbf{W}_\pi\}$ and ii) parameters of LM: $\boldsymbol{\Theta}_{LM}$. These parameters can be jointly trained by stochastic gradient descend with low-variance gradient estimation since LN and GS distributions have easy RT function.

5 Experiments

In this section, we evaluate the effectiveness and efficiency of our proposed mATM and EnsLM on different NLP tasks including document clusters, text classification, language generation and abstractive document summarization. Our code is available at https://github.com/BoChenGroup/EnsLM

5.1 Document clusters

The basic idea of mATM and EnsLM is that mATM can automatically discover the sample clusters which describe the data diversity. Therefore, we firstly evaluate the document clustering performance of mATM.

Datasets Following Yao et al. (2019), we consider two widely used document clustering datasets, 20News and R8 . This two datasets[2] can be found in the open source code of Yao et al. (2019).

[2]https://github.com/yao8839836/text_gcn

20News has 20 classes and consists of 18,846 documents with a vocabulary size of 61,188, partitioned into a training set of 11,314 documents and a test set of 7,532 ones. R8 is a subset of the Reuters 21578 dataset, which has 8 classes and was split into 5,485 training and 2,189 test documents. For these two datasets, we remove the stop words and use the 2,000 most frequent terms as the vocabulary. For all methods, we set the number of clusters as the number of classes.

Comparison models and implementation details To verify the effectiveness of mATM for clustering, three types of document clustering models are compared. i) **Raw+kmeans** performs K-means on raw BoW vectors, and **PCA+kmeans** uses PCA extract low-dimensional features and then uses K-means for clustering; ii) Train a topic model and then perform K-means for clustering on topic proportions, where we consider **LDA+kmeans** (Blei et al., 2003), **AVITM+kmeans** (Srivastava and Sutton, 2017), and **PFA+kmeans** (Zhou et al., 2012); iii) Deep neural network based clustering methods, including **Deep clustering** (Xie et al., 2016), and **DCN** (Yang et al., 2017), which jointly consider the feature extracting and clustering. Besides Raw+kmeans performing clustering on original inputs, others are on a latent feature space (For topic modeling, feature is the topic proportion). Following (Xie et al., 2016; Yang et al., 2017), the dimension of feature space equals to the number of clusters.

Table 1: Results of AC and NMI for document clustering task.

Model	20News		R8	
	AC	NMI	AC	NMI
Base+kmeans	30.2	37.0	40.1	30.2
PCA+kmeans	33.1	39.1	44.1	32.1
LDA+kmeans	37.4	38.1	53.8	36.9
PFA+kmeans	38.4	39.2	54.7	37.6
AVITM+kmeans	40.2	41.2	56.3	38.3
DeepCluster	42.2	43.5	58.23	41.02
DCN	44.8	48.4	59.34	43.2
mATM	**46.44**	**49.86**	**62.15**	**48.12**

Results Following Yang et al. (2017), since we know the ground-truth label and set the clustering number as the number of classes, we measure the clustering performance by accuracy (AC) and normalized mutual information (NMI), both of which are the higher the better. The results are shown in Table 1. Compared with the Base+kmeans, PCA+kmeans performs better since it extracts effective principal components. Benefiting from the learning of semantics for documents, the second group including three types of topic modeling outperforms PCA. Compared with the first two groups, the third group jointly considers the feature learning and clustering, thus achieving higher AC and NMI. Combined the advantages of topic modeling in extracting efficient features from documents and joint learning of feature extractor and clustering, mATM gets the SOTA performance for document clustering tasks on these two datasets.

The clustering results support our motivation of using mATM to discover the data diversity. In the following experiments, we evaluate the performance of both mATM and EnsLM on different language understanding and generation tasks.

5.2 Multi-domain sentiment classification

Sentiment classification (positive or negative) for different products is a fundamental language understanding task in NLP. For this task, the data diversity mainly arises from different domains (products) (Blitzer et al., 2007), which brings the problem that data from different domains may have different distributions.

Datasets To evaluate the performance of mATM and EnsLM in capturing the multi-domain property for sentiment classification, following Cai and Wan (2019), we perform experiments on the dataset released by Liu et al. (2017), which consists of product and movie reviews in 16 different domains. The data in each domain is randomly split into training set, development set and test set according to the proportion of 70%, 10%, 20%, whose statistics of the 16 datasets are listed in Appendix A.1.

Comparison models and implementation details Following (Cai and Wan, 2019), we firstly consider three base models, **BiLSTM** (Adhikari et al., 2019), **TextCNN** (Kim, 2014) and **BERT** (Devlin et al., 2019), which perform classification on every domains separately. Secondly, combining data from different domains together, we train the above three models named as **BiLSTM-mix**, **TextCNN-mix** and **DocBERT-mix**. Having obtained the ground-truth domain label, the previous works regard the multi-domain problem

as the multi-task learning (MTL) including **DA-MTL** (Zheng et al., 2018), **ASP-MTL** (Liu et al., 2017),and **MDAE** (Cai and Wan, 2019). All these works are developed from BiLSTM model. For our proposed EnsLM, we use TextCNN, BiLSTM and DocBERT as the backbone of EnsLM. We perform experiments on two types of EnsLM: i) with ground-truth (GT) domain label, we directly set $\boldsymbol{z}$ as the one-hot assignment vector (do not infer $\boldsymbol{z}$ from mATM), which is named as **BiLSTM-EnsLM-GT**, **TextCNN-EnsLM-GT**, and **BERT-EnsLM-GT**; ii) without GT domain label, we use mATM to infer $\boldsymbol{z}$, which is named as **BiLSTM-EnsLM-mATM**, **TextCNN-EnsLM-mATM**, and **BERT-EnsLM-mATM**. For model using mATM, we set the number of topics as 16. More detailed settings and implementation details can be found in Appendix B.1.

Table 2: Accuracy of sentiment classification.

Models	ACC	Models	ACC
TextCNN	84.3	TextCNN-Mix	85.3
BiLSTM	83.7	BiLSTM-Mix	86.6
BERT	88.1	BERT-Mix	91.3
TextCNN-EnsLM-GT	88.2	DA-MTL	88.2
BiLSTM-EnsLM w-GT	89.4	ASP-MTL	87.2
BERT-EnsLM w-GT	92.9	MDAE	90.1
TextCNN-EnsLM-mATM	88.8		
BiLSTM-EnsLM-mATM	90.2	-	-
BERT-EnsLM-mATM	**93.5**		

Results The results of averaged accuracy on all domains are given in Table 2, where the results except ours are obtained from Cai and Wan (2019). Comparing results on the first row, we can see that joint training models on all domains outperform separate training on each domain. Compared with BiLSTM-mix, having obtained the GT domain label, DA-MTL, ASP-MTL and MDAE (all of them are developed based on BiLSTM) consider the real domain knowledge in word embedding, feature extractor and attention layers, achieving higher accuracy. Similarly, with GT domain label, three models equipped with our proposed EnsLM performs better than their basic counterparts with a large margin. Assuming that GT domain labels are unavailable, we use mATM to infer the clustering assignment to guide the learning of EnsLM, which obtains the SOTA performance on all three basic models, even better than the models using GT domain label. We attribute it to the fact that compared with the hard GT domain label, mATM infers the soft clustering assignment, which not only reflect the domain characteristic of samples but also describe the samples having confused domain characteristics. For example samples from DVD may be similar with the ones from Electronics.

Table 3: Comparison of perplexity on four datasets.

Methods	APNEWS	IMDB	BNC	COCO
LSTM	60.13	65.16	95.73	21.34
Transformer-XL	58.73	60.11	97.14	19.32
TGVAE	48.73	57.11	87.86	-
rGBN-RNN	42.71	51.36	79.13	-
GPT-2	35.78	44.71	46.04	13.58
GPT-2-EnsLM-mATM	**23.67**	**35.48**	**40.79**	**12.45**

5.3 Language generation

Datasets In order to verify the effectiveness of our model on datasets of different lengths, we consider four publicly available corpora: APNEWS, IMDB, BNC, and COCO. Following Lau et al. (2017), we tokenize words and sentences using Stanford CoreNLP (Klein and Manning, 2003), lowercase all word tokens, and filter out word tokens that occur less than 10 times. For the topic model, we additionally exclude stopwords. All these corpora are partitioned into training, validation, and testing sets, whose summary statistics are provided in Appendix A.2.

Comparison models and implementation details We consider the following baseline models: **LSTM**, A standard LSTM language model (Hochreiter and Schmidhuber, 1997); **Tansnsformer-XL** enables learning dependency beyond a fixed length by introducing a recurrence mechanism and a novel position encoding scheme into the Transformer architecture (Dai et al., 2019); **TGVAE** (Wang et al., 2019), combines a variational auto-encoder based natural sequence model with a neural topic model; **rGBN-RNN** (Guo et al., 2020), extracts recurrent hierarchical semantic structure via a dynamic deep topic model to guide natural language generation; **GPT-2** (Radford et al., 2019) is a generative pre-training of a Transformer-based LM on a diverse set of unlabeled text. For our proposed model, **GPT-2-EnsLM-mATM** first uses mATM to infer semantic clusters for each sample, and then introduce this diversity information to pre-trained GPT2 by efficient weight modulation naturally. In the experiments, we use the Adam optimizer (Kingma and Ba, 2014) with learning rate 10^{-6}. The length of an input sample is limited to

Cluster #	Representive topics	Original sentences	Generated sentences
1	['kite', 'flying', 'sky', 'air', 'holding'] ['man', 'child', 'people', 'person', 'young'] ['beach', 'water', 'outside', 'near', 'park']	A child flying a pink kite on the beach. Person flying a kite high over a sea inlet. The bird is on a branch on the tree.	A man in a yellow and white outfit flying a kite. A young child flying a kite with a frisbee in the air. A person flying a kite near the water in a body of water.
2	['cake', 'slice', 'piece', 'chocolate', 'cream'] ['table', 'plate', 'fork', 'cup', 'eaten'] ['white', 'large', 'small', 'blue', 'red']	A women receives a cake that is blue. A piece of a chocolate cake on a plate. A small bird perched on a thin branch.	Two cakes with frosting on top sit on a red plate. A sandwich on a platter with a pickle and some fruit. A cake that has various decorations on it.
5	['baseball', 'bat', 'player', 'ball', 'game'] ['man', 'holding', 'batter', 'swinging', 'field'] ['pitch', 'boy', 'plate', 'catcher', 'swing']	A baseball player stands with a baseball bat. A baseball player is holding a baseball bat. A baseball player is swinging a baseball bat.	A man on a baseball field swinging a bat. A baseball player swinging a bat on a field. A batter is getting ready to hit the ball.

Figure 4: Example topics and their segment clusters inferred by a mATM from the COCO corpus, and the generated sentences under segment cluster guidance. For each cluster, top topics are shown in the column 2 respectively, original sentence are shown in the column 3 , and generated sentences are shown in the column 4.

1024. We set the mini-batch size as 8, the number of training epochs as 5. The clustering number of mATM is set to 64 for the first three datasets, while 80 for COCO dataset. More detailed settings and implementation details can be found in Appendix B.2

Results For fair comparison, we use standard language model perplexity as the evaluation metric. The results of all models on four datasets are given in Table 3, where the results of existing models are obtained from Guo et al. (2020). In the first group, Transformer-XL gets better result, which shows that the transformer-based model have better modeling capabilities. In terms of capturing the document global semantic information, the second group can improve performance significantly, which indicates that the topic model is effective in capturing document global information. Pre-training on massive data, the GPT-2 can obtains better results compared with above models. Although GPT-2 gets a good result, the GPT-2-EnsLM-mATM can improve performance significantly by capturing data diversity. It illustrates that even pre-training on large scale of corpus, EnsLM can further improve the performance of pre-trained LM via exploring data diversity. A similar phenomenon also appeared in the experiments conducted by Gururangan et al. (2020)

Sentence generation of EnsLM Given the learned GPT-2-EnsLM-mATM, we can sample the sentences conditioned on semantic clusters. Shown in the in Fig. 5, we select the top-3 topics to represent this cluster, and select original sentences according to the clustering results. we can see that most of the generated sentences conditioned on a semantic clusters are highly related to the given topics in terms of their semantic meanings but not necessarily in key words, indicating the LM is successfully guided by the cluster assignment. These observations suggest that GPT-2-EnsLM-mATM has successfully captured syntax and global semantics simultaneously for natural language generation. Similar to Fig. 5, we also provide other semantic clusters generated sentences in Appendix C.

Table 4: ROUGE scores on CNN/DM and Xsum test set, where the results are cited from Liu and Lapata (2019) and Wang et al. (2020)

Model	CNN/DM			XSUM		
	R1	R2	RL	R1	R2	RL
PTGEN	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+Cov	39.53	17.28	36.38	28.10	8.02	21.72
Transformer	40.21	17.76	37.09	29.41	9.77	23.01
BertSUM	42.13	19.60	39.18	38.81	16.50	31.27
BertSUM+TA	43.06	20.58	39.67	39.77	17.39	32.39
BertSUM+EnsLM	**43.34**	**20.78**	**39.83**	**40.01**	**17.62**	**32.57**

5.4 Abstractive summarization

Datasets We evaluate the effectiveness and efficiency of proposed model on two benchmark datasets, including the CNN/DailyMail (CNN/DM) (Hermann et al., 2015) and the XSum (Narayan et al., 2018). The summary styles of these datasets varies from highlights, composed of several sentences, to very brief one sentence. See more detailed descriptions in Appendix A.3. We perform data pre-processing following Liu and Lapata (2019).

Comparison models and implementation details We consider some baseline models, including LSTM based models **PTGEN** and **PTGEN+Cov** (See et al., 2017); Transformer based models **Tansformer**, **BertSUM** (Liu and Lapata, 2019); and **BertSUM+TA** which combine pre-trained model with topic model (Wang et al., 2020). We combine EnsLM with BertSUM on the abstractive summarization task. The clustering number of mATM is set to 64 for all datasets. Given BertSUM

checkpoints[3] on CNN/DM and XSum provided by Liu and Lapata (2019), we further fine-tune BertSUM+EnsLM. Besides, we adopt the settings in the BertSUM. Following Liu and Lapata (2019), in the test stage, we use beam search with size 5, select the top-3 checkpoints based on their evaluation loss on the validation set, and report the averaged results on the test set. More detailed settings and implementation details can be found in Appendix B.3.

Results ROUGE scores on CNN/DM, XSum have been exhibited in Tables 4, respectively. Focusing on the models without pre-training in the first group, Transformer achieves better performance compared with LSTM-based model, attributing to stronger sequence modeling capabilities. Further, the outperformance of BertSUM illustrates the fact that the combination of a pre-trained Bert encoder and a Transformer decoder is a better choice of sequence-to-sequence structure. Despite owning the same structure as the BertSUM, the BertSUM+TA employs a topic model to capture global document segment diversity, and achieving higher scores. Different from BertSUM+TA that introduces document semantic diversity by adding topic information, BertSUM+mATM combines BertSUM with EnsLM model, result in a better performance. Compared with BertSUM+TA, the performance improvement of our model is not enough promising is because they have been incorporated the topical information into the BertSum model which considering the segment diversity and contextual information. Note that the performance of our model improves significantly compared with BertSum, which can prove the effectiveness of our model.

6 Conclusion

In this paper, we first propose mATM to infer latent semantic clusters from raw text corpus, and then combine it with LM with efficient weight modulation, resulting in a more powerful EnsLM, which can be naturally extended to other LMs. In the future, we will study the effectiveness of EnsLM on other NLP tasks, such as the multi domain translation, and investigate whether EnsLM can be applied to the pre-training stage of Transformer.

[3] https://github.com/nlpyang/PreSumm

Acknowledgments

Bo Chen acknowledges the support of NSFC (61771361), Shaanxi Youth Innovation Team Project, the 111 Project (No. B18039) and the Program for Oversea Talent by Chinese Central Government. We acknowledge all the anonymous reviewers for their valuable comments and suggestions.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Rethinking complex neural network architectures for document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4046–4051.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

version 3 (BNC XML Edition) British National Corpus. 2007. Distributed by oxford university computing services on behalf of the bnc consortium.

Yitao Cai and Xiaojun Wan. 2019. Multi-domain sentiment classification based on domain-aware embedding and attention. In *IJCAI*, pages 4904–4910.

Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. 2020. Gan memory with no forgetting. *arXiv preprint arXiv:2006.07543*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2016. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683.

Dandan Guo, Bo Chen, Ruiying Lu, and Mingyuan Zhou. 2020. Recurrent hierarchical topic-guided rnn for language generation. In *International Conference on Machine Learning*, pages 3810–3821. PMLR.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Philipp Hennig, David Stern, Ralf Herbrich, and Thore Graepel. 2012. Kernel topic models. In *Artificial Intelligence and Statistics*, pages 511–519.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime G Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1834, Online. Association for Computational Linguistics.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430.

Jey Han Lau, Timothy Baldwin, and Trevor Cohn. 2017. Topically driven neural language model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 355–365, Vancouver, Canada. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of*

the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Catarina Cruz Silva, Chao-Hong Liu, Alberto Poncelas, and Andy Way. 2018. Extracting in-domain training corpora for neural machine translation using data selection methods. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 224–231.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Hanna Wallach, David Mimno, and Andrew McCallum. 2009. Rethinking lda: Why priors matter. *Advances in neural information processing systems*, 22:1973–1981.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 166–177, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. 2020. Friendly topic assistant for transformer based abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 485–497.

van der Wees. 2017. What's in a domain?: Towards fine-grained adaptation for machine translation. *Ph.D. thesis, University of Amsterdam*.

Marlies Van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. 2015. What's in a domain? analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 560–566.

Yeming Wen, Dustin Tran, and Jimmy Ba. 2020. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*.

Dustin Wright and Isabelle Augenstein. 2020. Transformer based multi-source domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7963–7974, Online. Association for Computational Linguistics.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. 2018. Whai: Weibull hybrid autoencoding inference for deep topic modeling. *arXiv preprint arXiv:1803.01328*.

Renjie Zheng, Junkun Chen, and Xipeng Qiu. 2018. Same representation, different attentions: Shareable sentence representation learning from multiple tasks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 4616–4622. AAAI Press.

Mingyuan Zhou, Yulai Cong, and Bo Chen. 2015. The poisson gamma belief network. *Advances in Neural Information Processing Systems*, 28:3043–3051.

Mingyuan Zhou, Lauren Hannah, David Dunson, and Lawrence Carin. 2012. Beta-negative binomial process and poisson factor analysis. In *Artificial Intelligence and Statistics*, pages 1462–1471.

Appendix

A Dataset descriptions

A.1 Multi-domain sentiment classification Dataset:

we perform experiments on the dataset[4] released by Liu et al. (2017), which consists of product and movie reviews in 16 different domains. The data in each domain is randomly split into training set, development set and test set according to the proportion of 70%, 10%, 20%. Statistics of the 16 datasets is shown in Table. 5.

A.2 Language Generation Datasets

In experiments, we evaluate the models on four benchmark language generation datasets. They are the APNEWS, IMDB, BNC, and COCO Caption. APNEWS is a collection of Associated Press news articles from 2009 to 2016. IMDB is a set of movie reviews collected by Maas et al. (2011). BNC is the written portion of the British National Corpus (British National Corpus, 2007), which contains documents from journals, books,letters, essays, memoranda, news and other types of text. COCO Caption has 80 object categories, and there are caption to describe the scene of the image (Lin et al., 2014). All these corpora are partitioned into training, validation, and testing sets, whose summary statistics are provided in Table. 6. The AGNEWS, IMDB and BNC datasets can be found in the release code[5] of **?**. And for COCO dataset, we will give processed dataset in our release code.

A.3 Abstractive Summarization Dataset

In experiments, we evaluate the models on two benchmark summarization datasets. The datasets[6] can be fround in the release code of Liu and Lapata (2019) They are the CNN/DailyMail news (CNN/DM) (Hermann et al., 2015) and XSum (Narayan et al., 2018).

CNN/DM CNN/DM consists of news and associated sentence highlights, that is a brief overview composed of a few sentences. Following the standard training/validation/testing splits in Hermann et al. (2015) without anonymizing entities, we perform our experiments. We splits sentences using the Stanford CoreNLP toolkit[7] and pre-process the dataset following Liu and Lapata (2019). .

XSum XSum includes $226,711$ news articles, each of which is associated with a one-sentence summary. We use the standard training/validation/testing splits $(204,045/11,332/11,334)$ and follow the pre-processing in Narayan et al. (2018). To satisfy the maximum capacity of the encoder in the base model, such as 512 for BertSUM, we use truncated document as the encoder input. Statistics of summarization datasets is shown in Table. 7.

B Implementation Details

B.1 Multi-domain sentiment classification Models

Note that we remove stop words to obtain the bag-of-word (BOW) vector for each document, and then use the BOW vectors to infer the mATM model.

CNN/BiLSTM-EnSLM-mATM: To reduce both computation and storage costs, we introduce a learnable key vector as $\boldsymbol{W}^{(t)}$, which can be combined with mATM by efficient weight modulation, leading to a CNN/BiLSTM-EnSLM-mATM. More specifically, we adopt 1-layer CNN/BiLSTMCNN with the channel/hidden size of 150 in CNN/BiLSTM-EnSLM-mATM equipped with 300-dimensional word embedding vecotrs. For optimization, the Adam optimizer is utilized here (Kingma and Ba, 2014) with a learning rate of 0.001. To avoid overfitting, we utilize the dropout and set its rate as 0.5. We set the size of minibatch as 50 in all experiments.

Bert-EnsLM-mATM: As a transformer-based model, the main component of Bert is query, key and value layer. And these component as MLP layer, we can combine Bert with mATM by efficient weight modulation easily. Specially, to re-

[4] https://github.com/FrankWork/fudan_mtl_reviews

[5] https://github.com/jhlau/topically-driven-language-model

[6] https://github.com/nlpyang/PreSumm

[7] https://stanfordnlp.github.io/CoreNLP/

Table 5: Statistics of the 16 datasets. The columns 2-4 denote the number of samples in training, development, and test sets. The last two columns represent the average length and vocabulary size of corresponding dataset.

Dataset	Train	Dev.	Test	Avg.L	Vocab	**Dataset**	Train	Dev.	Test	Avg.L	Vocab
Books	1400	200	400	159	62K	Toys	1400	200	400	90	28K
Elec.	1398	200	400	101	30k	Video	1400	200	400	156	57K
DVD	1400	200	400	173	69K	Baby	1300	200	400	104	26K
Kitchen	1400	200	400	89	28K	Mag.	1370	200	400	117	30K
Apparel	1400	200	400	57	21K	Soft.	1315	200	400	129	26K
Camera	1397	200	400	130	26K	Sports.	1400	200	400	94	30K
Health	1400	200	400	81	26K	IMDB	1400	200	400	269	44K
Music	1400	200	400	136	60K	MR	1400	200	400	21	12K

Table 6: Statistics of data for language generation task.

Collection	**Training**		**Development**		**Test**	
	Docs	Tokens	Docs	Tokens	Docs	Tokens
AGNEWS	50K	15M	2K	0.6M	2K	0.6M
IMDB	75K	20M	12.5K	0.3M	12.5K	0.3M
BNC	15K	18M	1K	1M	1K	1M
COCO	400K	4.1M	14K	0.2M	202K	2.1M

Table 7: Statistics of summarization datasets.

Datasets	Train	Dev.	Test	Doc Avg.L	Sum.Avg.L
CNN	90,266	1,220	1,093	760.50	45.70
DM	196,961	12,148	10,396	8080.04	54.65
XSUM	204,045	11,332	11,334	431.07	23.26

duce the amount of new parameters, we only introduce segment diversity information to query layer. For optimization, the Adam optimizer is utilized here (Kingma and Ba, 2014) with a learning rate of 0.00001. To avoid overfitting, we utilize the dropout and set its rate as 0.3. We set the size of minibatch as 16 in all experiments.

B.2 Language Generation Models

For language generation, we propose GPT-2-EnsLM-mATM which combine mATM with pretrained model GPT-2. And we introduce segment diversity information to query, key and value for each layer. We use the Adam optimizer (Kingma and Ba, 2014) with learning rate 10^{-6}. The length of an input sample is limited to 1024. We set the mini-batch size as 8, the number of training epochs as 5. The clustering number of mATM is set to 64 for the first three datasets, while 80 for COCO dataset.

B.3 Abstractive Summarization Models:

For abstractive summarization, we combine BertSum with mATM, which include a pretrained encoder and a transformer decoder. Specially, we introduce segment diversity information to query, key and value for each layer. We set the hyperparameters following the original papers and their public codes, where BertSUM[8] is referred to Liu and Lapata (2019). We fine-tune all models in four Nvidia GeForce RTX2080 TI GPUs. The experiments are performed with mini-batch size including 200 summary tokens with gradient accumulation every six iterations. Model checkpoints were saved and evaluated on the validation set every 1000 updates. Totally, we update the model $250,000$ times. Following Liu and Lapata (2019), we select the top-3 checkpoints based on their evaluation loss on the validation set, and report the averaged results on the test set. During decoding we used beam search

[8] https://github.com/nlpyang/BertSUM

Cluster#	Original sentences	Generated sentence
0	A horse has a harness on its face. A person on a horse jumping over some poles.	A horse that is eating some grass in a field. A horse drawn carriage traveling down a street.
4	A glass vase with flowers in it on a glass table. A green vase filled with yellow flowers on a table.	A bouquet of flowers in a vase on a table. A vase filled with lots of different colored flowers.
7	A fighter jet is flying through a clear sky. An airline jet flies beneath a cloudy sky.	A fighter jet flying through a cloudy sky. A jet airplane flying through a cloudy sky.
8	A street sign on a pole on a city street near a tree. A couple of buildings near a busy street.	A group of people walking down a street with umbrellas. A man riding on the back of a motorcycle on a street.
10	A bus is parked beside a bus station. One double decker bus is passinga parked bus.	A red double decker bus driving down a street. A double decker bus driving down a street

Figure 5: Example topics and their segment clusters inferred by a mATM from the COCO corpus, and the generated sentences under segment cluster guidance. For each cluster, original sentence are shown in the column 2, and generated sentence are shown in the column 3.

with size 5, and tuned the α for the length penalty between 0.6 and 1 on validation set. It is worth noting that our decoder applies neither a copy nor a coverage mechanism, despite their popularity in abstractive summarization.

C More Generation Examples

As shown in Fig. 5, we provide semantic clusters generated sentences by GPT-2-EnsLM-mATM on the coco corpus.