Combining Deep Generative Models and Multi-lingual Pretraining for Semi-supervised Document Classification

Yi Zhu[♡] **Ehsan Shareghi**[♠♡] **Yingzhen Li**[♢*] **Roi Reichart**[♣] **Anna Korhonen**[♡]
[♡] Language Technology Lab, University of Cambridge
[♠] Department of Data Science & AI, Monash University
[♢] Department of Computing, Imperial College London
[♣] Faculty of Industrial Engineering and Management, Technion, IIT
{yz568,alk23}@cam.ac.uk, ehsan.shareghi@monash.edu
yingzhen.li@imperial.ac.uk, roiri@technion.ac.il

Abstract

Semi-supervised learning through deep generative models and multi-lingual pretraining techniques have orchestrated tremendous success across different areas of NLP. Nonetheless, their development has happened in isolation, while the combination of both could potentially be effective for tackling task-specific labelled data shortage. To bridge this gap, we combine semi-supervised deep generative models and multi-lingual pretraining to form a pipeline for document classification task. Compared to strong supervised learning baselines, our semi-supervised classification framework is highly competitive and outperforms the state-of-the-art counterparts in low-resource settings across several languages. [1]

1 Introduction

Multi-lingual pretraining has been shown to effectively use unlabelled data through learning shared representations across languages that can be transferred to downstream tasks (Artetxe and Schwenk, 2019; Devlin et al., 2019; Wu and Dredze, 2019; Conneau and Lample, 2019). Nonetheless, the lack of labelled data still leads to inferior performance of the same model compared to those trained in languages with more labelled data such as English (Zeman et al., 2018; Zhu et al., 2019).

Semi-supervised learning is another appealing paradigm that supplements the labelled data with unlabelled data which is easy to acquire (Blum and Mitchell, 1998; Zhou and Li, 2005; McClosky et al., 2006, *inter alia*). In particular, deep generative models (DGMs) such as variational autoencoder (VAE; Kingma and Welling (2014)) are capable of capturing complex data distributions at scale with rich latent representations, and they have been used for semi-supervised learning in various tasks in NLP (Xu et al., 2017; Yin et al., 2018; Choi et al., 2019; Xie and Ma, 2019), as well as inducing cross-lingual word embeddings (Wei and Deng, 2017), and representation learning in combination with Transformers via pretraining (Li et al., 2020).

To leverage the benefits of both worlds, we propose a pipeline method by combining semi-supervised DGMs (SDGMs) based on M1+M2 model (Kingma et al., 2014) with multi-lingual pretraining. The pretrained model serves as multi-lingual encoder, and SDGMs can operate on top of it independently of encoding architecture. To highlight such independence, we experiment with two pretraining settings: (1) our LSTM-based cross-lingual VAE, and (2) the current stat-of-the-art (SOTA) multi-lingual BERT (Devlin et al., 2019).

Our experiments on document classification in several languages show promising results via the SDGM framework with different encoders, outperforming the SOTA supervised counterparts. We also illustrate that the end-to-end training of M1+M2 that was previously considered too unstable to train (Maaløe et al., 2016) is possible with a reformulation of the objective function.

2 Semi-supervised Learning with DGMs

Variational Autoencoder. VAE consists of a stochastic neural encoder $q_\phi(\mathbf{z}|\mathbf{x})$ that maps an input $\mathbf{x}$ to a latent representation $\mathbf{z}$, and a neural decoder $p_\theta(\mathbf{x}|\mathbf{z})$ that reconstructs $\mathbf{x}$, jointly trained by maximising the evidence lower bound (ELBO) of the marginal likelihood of the data:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] - \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big) \quad (1)$$

where the first term (reconstruction) maximises the expectation of data likelihood under the posterior distribution of $\mathbf{z}$, and the Kullback-Leibler (KL) divergence regulates the distance between the learned posterior and prior of $\mathbf{z}$.

[*] Work done while at Microsoft Research Cambridge.
[1] Code is available at https://github.com/cambridgeltl/mling_sdgms.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 894–908
April 19 - 23, 2021.

$$\mathcal{L}(\mathbf{x}, y) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1)]}_{\text{Reconstruction}} - \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,y)}[\log \frac{q_\phi(\mathbf{z}_2|\mathbf{z}_1, y)}{p(\mathbf{z}_2)} + \log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2, y)}]}_{\text{KL}} + \underbrace{\log p(y)}_{\text{Constant}}$$

$$\mathcal{U}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1)]}_{\text{Reconstruction}} - \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\mathbf{z}_1)q_\phi(\mathbf{z}_2|\mathbf{z}_1,y)}[\log \frac{q_\phi(\mathbf{z}_2|\mathbf{z}_1, y)}{p(\mathbf{z}_2)} + \log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2, y)} + \log \frac{q_\phi(y|\mathbf{z}_1)}{p(y)}]}_{\text{KL}}$$

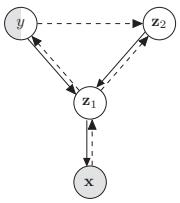

Table 1: Labelled and unlabelled objectives for M1+M2 model (left), and its corresponding graphical model (right).

Semi-supervised Learning with VAEs. The SDGM we use for semi-supervised learning is M1+M2 (Kingma et al., 2014), a graphical model (Table 1 (right)), with two layers of stochastic variables $\mathbf{z}_1$ and $\mathbf{z}_2$, with each being an isotropic Gaussian distribution. The first layer encodes the input sequence $\mathbf{x}$ into a deterministic hidden representation $\mathbf{h}$, and outputs the posterior distribution of $\mathbf{z}_1$:

$$q_\phi(\mathbf{z}_1|\mathbf{x}) = \mathcal{N}\Big(\boldsymbol{\mu}_\phi(\mathbf{h}), \text{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{h}))\Big) \quad (2)$$

As our SDGM is independent of the encoding architecture, we use different pretrained multi-lingual models to obtain $\mathbf{h}$, $\boldsymbol{\mu}_\phi(\mathbf{h})$, and $\boldsymbol{\sigma}^2_\phi(\mathbf{h})$, described in §3. The second layer computes the posterior distribution of $\mathbf{z}_2$, conditioned on sampled $\mathbf{z}_1$ from $q_\phi(\mathbf{z}_1|\mathbf{x})$ and a class variable y.

When we use labelled data, i.e. y is observed, $q_\phi(\mathbf{z}_2|\mathbf{z}_1, y)$ can be directly obtained. With unlabelled data, we calculate the posterior $q_\phi(\mathbf{z}_2, y|\mathbf{z}_1) = q_\phi(y|\mathbf{z}_1)q_\phi(\mathbf{z}_2|\mathbf{z}_1, y)$ by inferring y with the classifier $q_\phi(y|\mathbf{z}_1)$, and integrate over all possible values of y. Therefore, the ELBO for the labelled data $\mathcal{S}_l = \{\mathbf{x}, y\}$ is $\mathcal{L}(\mathbf{x}, y)$:

$$\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},y)}\Big[\log \frac{p_\theta(\mathbf{x}, y, \mathbf{z}_1, \mathbf{z}_2)}{q_\phi(\mathbf{z}_1, \mathbf{z}_2|\mathbf{x}, y)}\Big] \leq \log p(\mathbf{x}, y)$$

and for the unlabelled data $\mathcal{S}_u = \{\mathbf{x}\}$ is $\mathcal{U}(\mathbf{x})$:

$$\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|\mathbf{x})}\Big[\log \frac{p_\theta(\mathbf{x}, y, \mathbf{z}_1, \mathbf{z}_2)}{q_\phi(\mathbf{z}_1, \mathbf{z}_2, y|\mathbf{x})}\Big] \leq \log p(\mathbf{x})$$

where the generation part is $p_\theta(\mathbf{x}, y, \mathbf{z}_1, \mathbf{z}_2) = p(y)p(\mathbf{z}_2)p_\theta(\mathbf{z}_1|\mathbf{z}_2, y)p_\theta(\mathbf{x}|\mathbf{z}_1)$, $p(y)$ is uniform distribution as the prior of y, $p(\mathbf{z}_2)$ is standard Gaussian distribution as the prior of $\mathbf{z}_2$, and $p_\theta(\mathbf{x}|\mathbf{z}_1)$ is the decoder, which can have different architectures depending on the encoder (§4).

The objective function maximises both the labelled and unlabelled ELBOs while training directly the classifier with the labelled data as well:

$$\mathcal{J} = \sum_{(\mathbf{x},y)\in\mathcal{S}_l} \big(\mathcal{L}(\mathbf{x}, y) + \alpha\mathcal{J}_{cls}(\mathbf{x}, y)\big) + \sum_{\mathbf{x}\in\mathcal{S}_u} \mathcal{U}(\mathbf{x})$$

where $\mathcal{J}_{cls}(\mathbf{x}, y) = \mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})}[q_\phi(y|\mathbf{z}_1)]$, and α is a hyperparameter to tune. Considering the factorisation of the model according to the graphical model, we can rewrite the $\mathcal{L}(\mathbf{x}, y)$ and $\mathcal{U}(\mathbf{x})$ as shown in Table 1(left). The reconstruction term is the expected log likelihood of the input sequence $\mathbf{x}$, same for both ELBOs. The KL term regularises the posterior distributions of $\mathbf{z}_1$ and $\mathbf{z}_2$ according to their priors. Additionally for $\mathcal{U}(\mathbf{x})$, as mentioned before, we first infer y and treat it as if it were observed, so we need to compute the expected KL term over $q_\phi(y|\mathbf{z}_1)$ regularised by $\text{KL}(q_\phi(y|\mathbf{z}_1)\|p(y))$.

Due to its training difficulty, M1+M2 is trained layer-wise in Kingma et al. (2014), where the first layer is trained according to Eq. 1 and fixed, before the second layer is trained on top. However, in our experiments (§4.1) we found that M1+M2 is easier to train end-to-end. We attribute this to our mathematical reformulation of the objective functions, giving rise to a more stable optimisation schedule.

3 SDGMs with Multi-lingual Pretraining

LSTM-based Encoder with VAE Pretraining. Our pretraining is based on the framework of Wei and Deng (2017), in which they pretrain a cross-lingual VAE with parallel corpus as input. However, the parallel corpus is expensive to obtain, and only the resulting cross-lingual embeddings rather than the whole encoder could be used due to the parallel input limitation of the model. To address these shortcomings, we propose non-parallel cross-lingual VAE (NXVAE), which has the same graphical model as the vanilla VAE. Each language i is associated with its own word embedding matrix, and its input sequence $\mathbf{x}_i$ is processed via a two layer BiLSTM (Hochreiter and Schmidhuber, 1997) shared across languages. We use the concatenation of the BiLSTM last hidden states as $\mathbf{h}$, and compute $q_\phi(\mathbf{z}|\mathbf{x}_i)$ with Eq. 2, so that $\mathbf{z}$ becomes the joint cross-lingual semantic space. A language specific bag-of-word decoder (BOW; Miao et al. (2016)) is then used to reconstruct the input sequence. Additionally, we optimise a language discriminator as an adversary (Lample et al., 2018a) to encourage the mixing of different language representations and keep the shared encoder language-agnostic. After pretraining NXVAE, we transfer the whole encoder, including $\boldsymbol{\mu}_\phi(\mathbf{h})$ and

$\boldsymbol{\sigma}^2_\phi(\mathbf{h})$, directly into our SDGM framework and treat it as $q_\phi(\mathbf{z}_1|\mathbf{x})$ component of the model (§4.1).

Multi-lingual BERT Encoder. To show that our SDGM is effective with other encoding architectures, we use the pretrained multi-lingual BERT (mBERT; Devlin et al. (2019))[2] as our encoder. Given an input sequence, the pooled `[CLS]` representation is used as $\mathbf{h}$ to compute $q_\phi(\mathbf{z}_1|\mathbf{x})$ (Eq. 2). Different from NXVAE, we initialise the parameters of $\boldsymbol{\mu}_\phi(\mathbf{h})$ and $\boldsymbol{\sigma}^2_\phi(\mathbf{h})$ randomly.

4 Experiments

We perform document classification on the class balanced multilingual document classification corpus (MLDoc; Schwenk and Li (2018)). Each document is assigned to one of the four news topic classes: *corporate/industrial* (C), *economics* (E), *government/social* (G), and *markets* (M). We experiment with five representative languages: EN, DE, FR, RU, ZH, and use 1k instance training set along with the standard development and test set. For experiments with varying labelled data size, the rest training data from 1k corpus is used as unlabelled data. The full statistics are shown in Table 2. Three languages (EN, DE, FR) are tested for LSTM encoder with VAE pretraining (§4.1) and all five languages for mBERT encoder (§4.2). All documents are lowercased. We report *accuracy* for evaluation following Schwenk and Li (2018).

For all experiments, We use Adam (Kingma and Ba, 2015) as optimiser, but with different learning rates for both settings and pretraining. We implemented the model with Pytorch[3] 1.10 (Paszke et al., 2019), and use GeForce GTX 1080Ti GPUs. See the Appendix for details about model configurations and training.

4.1 LSTM Encoder with VAE Pretraining

Experimental Setup. For pretraining NXVAE, we use three language pairs: EN-DE, EN-FR and DE-FR constructed from Europarl v7 parallel corpus (Koehn, 2005),[4] where only two language pairs are available: EN-DE and EN-FR, which consist of four datasets in total: $(\text{EN, DE})_{\text{EN-DE}}$, and $(\text{EN, FR})_{\text{EN-FR}}$. For DE-FR, we pair $\text{DE}_{\text{EN-DE}}$ and $\text{FR}_{\text{EN-FR}}$ directly as pseudo parallel data. We trim all datasets into exactly the same sentence size, and preprocess them with: tokenization, lowercasing, substituting digits with 0, and removing all punctuations, redundant spaces and empty lines. We randomly sample a small part of parallel sentences to build a development set. For models which do not require parallel input, e.g. NXVAE, we mix the two datasets of a language pair together. To avoid KL-collapse during pretraining, a weight α on the KL term in Eq. 1 is tuned and fixed to 0.1 (Higgins et al., 2017; Alemi et al., 2018). We only run one trial with fixed random seed for both pretraining and document classification. Training details can be found in the Appendix.

	C	E	G	M	Total
EN	270	234	252	244	1000
	228	238	266	268	1000
	991	1000	1030	979	4000
DE	270	240	245	245	1000
	229	268	266	237	1000
	984	1026	1022	968	4000
FR	227	262	258	253	1000
	257	237	237	269	1000
	999	973	998	1030	4000
RU	261	288	184	267	1000
	265	272	204	259	100
	1073	1121	706	1100	4000
ZH	294	286	109	311	1000
	324	300	93	283	1000
	1169	1215	363	1253	4000

Table 2: Statistics of MLDoc in five languages. Instance numbers for each class along with the total numbers are shown. For each language, three rows are training, development and test set instance numbers.

As our supervised baselines we compare with the following two groups: (I) NXVAE-based supervised models which are pretrained NXVAE encoder with a multi-layer perceptron classifier on top (denoted by NXVAE-z_1 ($q_\phi(y|\mathbf{z}_1)$) or NXVAE-h ($q_\phi(y|\mathbf{h})$) depending on the representation fed into the classifier; or NXVAE-z_1 models initialised with different pretrained embeddings: random initialisation (RAND), mono-lingual fastText (FT; Bojanowski et al. (2017)), unsupervised cross-lingual MUSE (Lample et al., 2018b), pretrained embeddings from Wei and Deng (2017) (PEMB), and our resulting embeddings from pretrained NXVAE (NXEMB).[5] (II) We also pretrain a word-based BERT (BERTW) with parameter size akin to NXVAE on the same data, and fine-tune it directly.[6]

For our semi-supervised experiments, we test

[2] https://github.com/google-research/bert/blob/master/multilingual.md.

[3] https://pytorch.org/.

[4] https://www.statmt.org/europarl/.

[5] All embeddings are pretrained on the same Europarl data.

[6] We also trained subword-based models for BERT and NXVAE, and observed similar trends. See the Appendix.

Word pair	Lang	kNNs ($k = 3$)
president (EN)	EN	mr, madam, gentlemen
	DE	präsident, herr, kommissar
präsident (DE)	EN	president, mr, madam
	DE	herr, kommissar, herren
great (EN)	EN	deal, with, a
	DE	große, eine, gute
groß (DE)	EN	striking, gets, lucrative
	DE	gering, heikel, hoch
said (EN)	EN	already, as, been
	DE	gesagt, mit, dem
sagte (DE)	EN	he, rightly, said
	DE	vorhin, kollege, kommissar

Table 3: Cosine similarity-based nearest neighbours of words (left column) in embedding spaces of EN and DE.

two types of decoders with different model capacities: BOW and GRU (Cho et al., 2014). We use M1+M2+BOW (GRU) to denote the model with joint training using a specific decoder, and M1+M2 to denote the original model in Kingma et al. (2014) with layer-wise training.[7] We also add a semi-supervised self-training method (McClosky et al., 2006) for BERTW to leverage the unlabelled data (BERTW+ST), where we iteratively add predicted unlabelled data when the model achieves a better dev. accuracy until convergence.

Qualitative Results. Table 3 illustrates the quality of the learned alignments in the cross-lingual space of NXVAE for EN-DE word pairs.

Classification Results. Table 4 (**EN-DE**) shows that within supervised models the NXVAE-z_1 substantially outperforms other supervised baselines with the exception of BERTW. The fact that NXVAE-z_1 is significantly better than NXVAE-h, suggests that pretraining has enabled $\mathbf{z}_1$ to learn more general knowledge transferable to this task. Combining with SDGMs, our best pipeline outperforms all baselines across data sizes and languages, including BERTW+ST with bigger gaps in fewer labelled data scenario. We observe the same trend of performance in both supervised and semi-supervised DGM settings on **EN-FR** and **DE-FR**.

For decoder, BOW outperforms the GRU, a finding in line with the results of Artetxe et al. (2019) which suggests a few keywords seem to suffice for this task. The poor performance of the original M1+M2, implies the domain discrepancy between

[7] We also compared this against a more complex Skip Deep Generative Model (Maaløe et al., 2016), but found that end-to-end M1+M2 performs better. Details in the Appendix.

# Labels	32	64	128	1K	32	64	128	1K
EN-DE	EN				DE			
NXVAE-h	56.5	61.7	59.5	78.4	53.6	66.7	78.9	87.2
NXVAE-z_1	63.9	71.4	77.0	91.6	**65.0**	73.8	82.7	93.0
RAND	50.1	54.2	62.3	82.5	47.2	60.8	69.0	84.8
FT	36.3	49.4	61.1	80.9	45.0	54.3	69.2	86.1
MUSE	59.8	65.4	71.8	88.4	45.1	66.2	79.7	90.4
PEMB	36.4	53.9	50.9	84.4	39.4	52.0	69.0	86.7
NXEMB	61.5	62.0	68.6	85.4	53.4	71.2	75.9	88.8
BERTW	**67.7**	**72.7**	**84.6**	**91.8**	58.1	**77.5**	**89.2**	**94.0**
M1+M2	56.6	67.1	70.3	-	52.6	67.2	76.8	-
M1+M2+BOW	**79.8**	**81.7**	**87.2**	-	70.5	79.6	**89.7**	-
M1+M2+GRU	75.3	79.4	84.9	-	**75.1**	**80.0**	87.1	-
BERTW+ST	68.4	73.9	86.4	-	59.6	79.7	89.4	-
EN-FR	EN				FR			
NXVAE-h	**71.4**	73.8	78.6	88.0	62.8	72.7	79.9	88.9
NXVAE-z_1	71.2	**75.3**	**80.4**	**91.2**	**68.3**	**75.0**	**81.4**	**91.7**
M1+M2	71.8	73.5	76.5	-	66.2	78.7	79.7	-
M1+M2+BOW	**81.0**	**85.5**	**88.2**	-	80.3	**86.0**	**88.8**	-
M1+M2+GRU	75.3	81.4	83.8	-	**80.7**	82.3	87.4	-
DE-FR	DE				FR			
NXVAE-h	42.4	53.3	74.3	85.7	39.8	51.8	58.5	86.9
NXVAE-z_1	**63.3**	**75.4**	**81.3**	**92.1**	**60.1**	**71.1**	**78.4**	**91.4**
M1+M2	59.1	70.6	75.4	-	48.5	57.4	60.7	-
M1+M2+BOW	**78.0**	**83.2**	**88.3**	-	**81.4**	**84.5**	**88.4**	-
M1+M2+GRU	74.6	80.5	86.2	-	66.2	77.2	81.9	-

Table 4: MLDoc test accuracy for EN-DE, EN-FR and DE-FR pairs. The best results for supervised and semi-supervised models are in bold.

pretraining and task data, and highlights the impact of fine-tuning. In addition, our NXEMB, as a byproduct of NXVAE, performs comparably well with MUSE, and better than all other embedding models including its parallel counterpart PEMB.

4.2 Multi-lingual BERT Encoder

Experimental Setup. We use the cased mBERT, a 12 layer Transformer (Vaswani et al., 2017) trained on Wikipedia of 104 languages with 100k shared WordPiece vocabulary. The training corpus is larger than Europarl by orders of magnitude, and high-resource languages account for most of the corpus. We use the best SDGM setup (M1+M2+BOW §4.1), on top of mBERT encoder against the mBERT supervised model with a linear layer as classifier (SUP-h) in 5 representative languages (EN, DE, FR, RU, ZH). We report the results over 5 runs due to the training instability of BERT (Dodge et al., 2020; Mosbach et al., 2020).

Classification Results. Figure 1 demonstrates that M1+M2+BOW outperforms the SOTA supervised mBERT (SUP-h) on average across all languages. This corroborates the effectiveness of our

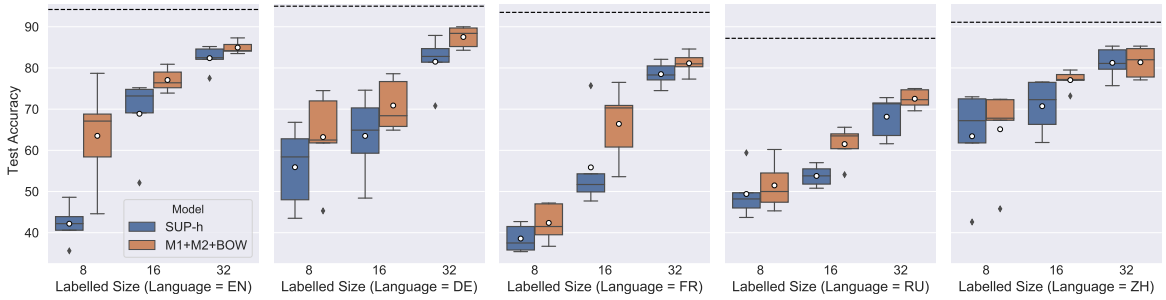

Figure 1: Boxplot of test accuracy scores for SUP-h and M1+M2+BOW over 5 runs. The mean is shown as white dot. The dashed line is the test mean accuracy of SUP-h trained on 1k labelled data of the corresponding language.

SDGM in leveraging unlabelled data within smaller labelled data regime, as well as its independence from encoding architecture.[8] As expected, the gap is generally larger with 8 and 16 labelled data, but reduces as the data size grows to 32. The variance shows similar pattern, but with relatively large values because of the instability of mBERT. Interestingly, the performance difference seems to be more notable in high-resource languages with more pretrained data, whereas in languages with fewer pretrained texts or vocabulary overlaps such as RU and ZH, the two models achieve closer results.

5 Conclusion

We bridged between multi-lingual pretraining and deep generative models to form a semi-supervised learning framework for document classification. While outperforming SOTA supervised models in several settings, we showed that the benefits of SDGMs are orthogonal to the encoding architecture or pretraining procedure. It opens up a new avenue for SDGMs in low-resource NLP by incorporating unlabelled data potentially from different domains and languages. Our preliminary results in cross-lingual zero-shot setting with SDGMs+NXVAE are promising, and we will continue the exploration in this direction as future work.

Acknowledgments

This work is supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (648909). The first author would like to thank Victor Prokhorov and Xiaoyu Shen for their comments on this work. The authors would like to thank the three anonymous reviewers for their helpful suggestions.

[8] Compared to smaller pretraining corpus (§4.1), we found that the representations pretrained on large corpus are less prune to overfit to the training instances of the task. We observe that training without the KL regularisation yields better performance for SDMGs+mBERT.

References

Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. 2018. Fixing a broken ELBO. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 159–168. PMLR.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998*, pages 92–100. ACM.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Jihun Choi, Taeuk Kim, and Sang-goo Lee. 2019. A cross-sentence latent variable model for semi-supervised text sequence matching. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4747–4761. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Chunyuan Li, Xiang Gao, Yuan Li, Xiujun Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. *CoRR*, abs/2004.04092.

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. 2016. Auxiliary deep generative models. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1445–1453. JMLR.org.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1727–1736. JMLR.org.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. *CoRR*, abs/2006.04884.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Liang-Chen Wei and Zhi-Hong Deng. 2017. A variational autoencoding approach for inducing cross-lingual word embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4165–4171. ijcai.org.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 833–844. Association for Computational Linguistics.

Zhongbin Xie and Shuai Ma. 2019. Dual-view variational autoencoders for semi-supervised text matching. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5306–5312. ijcai.org.

Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. Variational autoencoder for semi-supervised text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3358–3364. AAAI Press.

Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. 2018. Structvae: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 754–765. Association for Computational Linguistics.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, October 31 - November 1, 2018*, pages 1–21. Association for Computational Linguistics.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.*, 17(11):1529–1541.

Yi Zhu, Benjamin Heinzerling, Ivan Vulic, Michael Strube, Roi Reichart, and Anna Korhonen. 2019. On the importance of subword information for morphological tasks in truly low-resource languages. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 216–226. Association for Computational Linguistics.

A Derivations of semi-supervised ELBOs

We derive the full ELBOs of both labelled and unlabelled data for M1+M2 and Auxiliary Skip Deep Generative Model (AUX; Maaløe et al. (2016)).[9] We first use $(\cdot)$ to represent different conditional variables for the two models so that the derivations can be unified, then we will realise it with the model-specific conditions in the end.

As written in the paper, the labelled ELBO for both models is:

$$\begin{aligned}&\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},y)}\left[\log \frac{p_\theta(\mathbf{x},y,\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},y)}\right]\\&=\mathcal{L}(\mathbf{x},y)\leq \log p(\mathbf{x},y)\end{aligned}$$

Expanding the ELBO, we will have:

$$\begin{aligned}&\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},y)}[\log \frac{p_\theta(\mathbf{x},y,\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},y)}]\\=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\cdot)}[\\&\log p(\mathbf{z}_2)+\log p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)+\log p_\theta(\mathbf{x}|\cdot)+\log p(y)-\\&\log q_\phi(\mathbf{z}_2|\cdot)-\log q_\phi(\mathbf{z}_1|\mathbf{x})]\\=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\cdot)}[\log p_\theta(\mathbf{x}|\cdot)]-\\&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\cdot)}[\\&\log q_\phi(\mathbf{z}_2|\cdot)+\log q_\phi(\mathbf{z}_1|\mathbf{x})-\\&\log p(\mathbf{z}_2)-\log p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)-\log p(y)]\\=&\mathbb{E}_{q_\phi(\mathbf{z}\mathbf{1}|\mathbf{x})q_\phi(\mathbf{z}_2|\cdot)}[\log p_\theta(\mathbf{x}|\cdot)]-\\&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\cdot)}[\\&\log \frac{q_\phi(\mathbf{z}_2|\cdot)}{p(\mathbf{z}_2)}+\log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)}-\log p(y)]\end{aligned}$$

After realising $(\cdot)$, we can then obtain the labelled ELBO for M1+M2 and AUX in the original paper:

$$\begin{aligned}&\mathcal{L}_{\text{M1+M2}}(\mathbf{x},y)\\=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1)]-\\&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,y)}[\\&\log \frac{q_\phi(\mathbf{z}_2|\mathbf{z}_1,y)}{p(\mathbf{z}_2)}+\log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)}-\log p(y)]\end{aligned}$$

$$\begin{aligned}&\mathcal{L}_{\text{AUX}}(\mathbf{x},y)\\=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y)}[\log p_\theta(\mathbf{x}|\mathbf{z}_1,\mathbf{z}_2,y)]-\\&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y)}[\\&\log \frac{q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y)}{p(\mathbf{z}_2)}+\log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)}-\log p(y)]\end{aligned}$$

[9] As mentioned in the footnote of original paper, we compare M1+M2 with AUX in LSTM encoder with VAE pretraining, but found that the simpler M1+M2 performs better. Results on AUX can be found in §D.

For the unlabelled ELBO, y is unobservable:

$$\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},y,\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|\mathbf{x})}\right] = \mathcal{U}(\mathbf{x}) \leq \log p(\mathbf{x})$$

After expansion:

$$\begin{aligned}
&\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|\mathbf{x})}[\log \frac{p_\theta(\mathbf{x},y,\mathbf{z}_1,\mathbf{z}_2)}{q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|\mathbf{x})}] \\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\cdot)q_\phi(\mathbf{z}_2|\cdot)}[\\
&\log p(\mathbf{z}_2) + \log p_\theta(\mathbf{z}_1|\mathbf{z}_2,y) + \log p_\theta(\mathbf{x}|\cdot) + \log p(y) - \\
&\log q_\phi(\mathbf{z}_2|\cdot) - \log q_\phi(\mathbf{z}_1|\mathbf{x}) - \log q_\phi(y|\cdot)] \\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\cdot)q_\phi(\mathbf{z}_2|\cdot)}[\log p_\theta(\mathbf{x}|\cdot)] - \\
&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\cdot)q_\phi(\mathbf{z}_2|\cdot)}[\\
&\log q_\phi(\mathbf{z}_2|\cdot) + \log q_\phi(\mathbf{z}_1|\mathbf{x}) + \log q_\phi(y|\cdot) - \\
&\log p(\mathbf{z}_2) - \log p_\theta(\mathbf{z}_1|\mathbf{z}_2,y) - \log p(y)] \\
=&\mathbb{E}_{q_\phi(\mathbf{z}1|\mathbf{x})q_\phi(y|\cdot)q_\phi(\mathbf{z}_2|\cdot)}[\log p_\theta(\mathbf{x}|\cdot)] - \\
&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\cdot)q_\phi(\mathbf{z}_2|\cdot)}[\\
&\log \frac{q_\phi(\mathbf{z}_2|\cdot)}{p(\mathbf{z}_2)} + \log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)} + \log \frac{q_\phi(y|\cdot)}{p(y)}]
\end{aligned}$$

Similarly, we will get unlabeled ELBO of M1+M2 and AUX:

$$\begin{aligned}
&\mathcal{U}_{\text{M1+M2}}(\mathbf{x}) \\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1)] - \\
&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\mathbf{z}_1)q_\phi(\mathbf{z}_2|\mathbf{z}_1,y)}[\\
&\log \frac{q_\phi(\mathbf{z}_2|\mathbf{z}_1,y)}{p(\mathbf{z}_2)} + \log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)} + \log \frac{q_\phi(y|\mathbf{z}_1)}{p(y)}]
\end{aligned}$$

$$\begin{aligned}
&\mathcal{U}_{\text{AUX}}(\mathbf{x}) \\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\mathbf{z}_1,\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y)}[\log p_\theta(\mathbf{x}|\mathbf{z}_1,\mathbf{z}_2,y)] - \\
&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\mathbf{z}_1,\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y)}[\\
&\log \frac{q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y)}{p(\mathbf{z}_2)} + \log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)} + \log \frac{q_\phi(y|\mathbf{z}_1,\mathbf{x})}{p(y)}]
\end{aligned}$$

In our experiments, we sample $\mathbf{z}_1$ and $\mathbf{z}_2$ once during inference, so both labeled and unlabeled ELBOs can be approximated by:

$$\begin{aligned}
&\mathcal{L}(\mathbf{x},y) \\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\cdot)}[\log p_\theta(\mathbf{x}|\cdot)] - \\
&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\cdot)}[\\
&\log \frac{q_\phi(\mathbf{z}_2|\cdot)}{p(\mathbf{z}_2)} + \log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)} - \log p(y)] \\
\approx&\log p_\theta(\mathbf{x}|\cdot) + \log p(y) - \\
&\text{KL}(q_\phi(\mathbf{z}_2|\cdot)\|p(\mathbf{z}_2)) - \text{KL}(q_\phi(\mathbf{z}_1|\mathbf{x})\|p_\theta(\mathbf{z}_1|\mathbf{z}_2,y))
\end{aligned}$$

$$\begin{aligned}
&\mathcal{U}(\mathbf{x}) \\
=&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\cdot)q_\phi(\mathbf{z}_2|\cdot)}[\log p_\theta(\mathbf{x}|\cdot)] - \\
&\mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\cdot)q_\phi(\mathbf{z}_2|\cdot)}[\\
&\log \frac{q_\phi(\mathbf{z}_2|\cdot)}{p(\mathbf{z}_2)} + \log \frac{q_\phi(\mathbf{z}_1|\mathbf{x})}{p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)} + \log \frac{q_\phi(y|\cdot)}{p(y)}] \\
\approx&\log p_\theta(\mathbf{x}|\cdot) - \text{KL}(q_\phi(y|\cdot)\|p(y)) - \\
&\mathbb{E}_{q_\phi(y|\cdot)}[\text{KL}(q_\phi(\mathbf{z}_2|\cdot)\|p(\mathbf{z}_2))] - \\
&\mathbb{E}_{q_\phi(y|\cdot)}[\text{KL}(q_\phi(\mathbf{z}_1|\mathbf{x})\|p_\theta(\mathbf{z}_1|\mathbf{z}_2,y))]
\end{aligned}$$

B Factorisation of M1+M2 and AUX

The two models have different factorisations, with M1+M2 being written as:

$$\begin{aligned}
q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},y) &= q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,y) \\
q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|\mathbf{x}) &= q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\mathbf{z}_1)q_\phi(\mathbf{z}_2|\mathbf{z}_1,y) \\
p_\theta(\mathbf{x},y,\mathbf{z}_1,\mathbf{z}_2) &= p(y)p(\mathbf{z}_2)p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)p_\theta(\mathbf{x}|\mathbf{z}_1) \\
\mathcal{J}_{cls}(\mathbf{x},y) &= \mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})}[q_\phi(y|\mathbf{z}_1)]
\end{aligned}$$

and AUX is factorised as follows:

$$\begin{aligned}
q_\phi(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x},y) &= q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y) \\
q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|\mathbf{x}) &= q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(y|\mathbf{z}_1,\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1,\mathbf{x},y) \\
p_\theta(\mathbf{x},y,\mathbf{z}_1,\mathbf{z}_2) &= p(y)p(\mathbf{z}_2)p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)p_\theta(\mathbf{x}|\mathbf{z}_1,\mathbf{z}_2,y) \\
\mathcal{J}_{cls}(\mathbf{x},y) &= \mathbb{E}_{q_\phi(\mathbf{z}_1|\mathbf{x})}[q_\phi(y|\mathbf{z}_1,\mathbf{x})]
\end{aligned}$$

where $q_\phi(\mathbf{z}_1|\mathbf{x})$, $q_\phi(\mathbf{z}_2|\cdot)$, and $p_\theta(\mathbf{z}_1|\mathbf{z}_2,y)$ are parameterised as diagonal Gaussians, and other distributions are defined as:

$$\begin{aligned}
q_\phi(y|\cdot) &= \text{Cat}(y|\pi_\phi(\cdot)) & p(y) &= \text{Cat}(y|\pi) \\
p(\mathbf{z}_2) &= \mathcal{N}(\mathbf{z}_2|\mathbf{0},\mathbf{I}) & p_\theta(\mathbf{x}|\cdot) &= f(\mathbf{x},\cdot;\theta)
\end{aligned}$$

where $\text{Cat}(\cdot)$ is a multinomial distribution and y is treated as latent variables if it is unobserved in unlabelled case. $f(\mathbf{x},\cdot;\theta)$ serves as the decoder and calculates the likelihood of the input sequence $\mathbf{x}$.

C Details on LSTM Encoder with VAE Pretraining

C.1 Data preprocessing and statistics

We use two pairs of data from Europarl v7 (Koehn, 2005):[10] EN-DE and EN-FR, which consist of four datasets in total: $\text{EN}_{\text{EN-DE}}$, $\text{DE}_{\text{EN-DE}}$, $\text{EN}_{\text{EN-FR}}$, and $\text{FR}_{\text{EN-FR}}$. Regarding DE-FR data, we take the datasets of $\text{DE}_{\text{EN-DE}}$ and $\text{FR}_{\text{EN-FR}}$.

For each language pair, the sentences in the same line of both datasets are a pair of parallel sentences. We do the following preprocessing to each dataset: tokenization; lower case; substitute digits with 0; remove all punctuations; remove redundant spaces and empty lines. Then we trim all four datasets into exactly the same sentence size. We randomly split a small part of parallel sentences to build a dev. set, which leads to 189m lines of training set and 13995 lines of dev. set for each language. Then we shuffle each dataset so that each language pair is not parallel anymore (for both train and dev. sets).

Our goal is to merge the two datasets of each pair and scramble them to form a single dataset. In practice, we keep each dataset separate, and sample

[10] https://www.statmt.org/europarl/.

a batch randomly from one language alternatively during pretraining, so that the data from both languages are mixed.

C.2 Model and training details

Instead of optimising the standard VAE, we optimise the following objective for NXVAE (Higgins et al., 2017; Alemi et al., 2018):

$$\mathcal{J}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \alpha \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \quad (3)$$

where we manually tune the fixed hyperparameter α on EN-DE data to reach a good balance between the reconstruction and the KL empirically. We select $\alpha = 0.1$ and apply it for the pretraining of other language pairs as well. The model and training details of XNVAE are shown in Table 5 (left).

C.3 Pretraining other models

For MLDoc supervised document classification, we also pretrain other baseline models to compare with ONLY for **EN-DE** pair:

Cross-lingual VAE with parallel input (PEMB; Wei and Deng (2017)): For the model of Wei and Deng (2017), we run the original code directly on the same EN-DE Europarl data without changing any of the model architecture. Since the model requires parallel input, we take the preprocessed and split EN-DE data. However, we do not shuffle each dataset, but rather feed them as parallel input to the model, so that the model and our corresponding NXVAE use the same amount and content of the data.

Subword-based non-parallel cross-lingual VAE SNXVAE: Instead of having separate vocabulary and decoders for each language, we use a single vocabulary and decoder for SNXVAE. We build the vocabulary with SentencePiece[11] of size 1e4. All other settings are the same as NXVAE. Its model and training details can be found in Table 5 (right).

Word and subword-based BERT model BERTW/BERTSW : For BERTW, we change the vocabulary and model size to be comparable with NXVAE. Note that the vocabulary size of BERTW is the same as the intersected vocabulary size of the two languages in NXVAE. We only use the masked language model objective during pretraining, and discard the objective of next sentence prediction.[12] For BERTSW, we use the same vocabulary as SNXVAE and set the model to similar parameter size as SNXVAE. The model and training details of BERTW and BERTSW are shown in Table 6.

D More Results on Document Classification

D.1 LSTM Encoder with VAE Pretraining

Supervised Learning. Our base model is NXVAE-z_1, which adds an MLP classifier $q_\phi(y|\mathbf{z}_1)$ on top of the encoder with the same architecture of the NXVAE. The similar applies to the subword-based models SNXVAE-z_1. NXVAE-h takes the deterministic $\mathbf{h}$ as the input to $q_\phi(y|\mathbf{x})$. All our baseline models with pretrained embeddings use the architecture of NXVAE-z_1. For fastText (FT), we train the embeddings of both languages with the same data of $\mathrm{EN}_{\text{EN-DE}}$ and $\mathrm{DE}_{\text{EN-DE}}$. For MUSE, we align on the pretrained FT embeddings. For BERTW and BERTSW, we use the library Transformers[13] for classification, and initialise the models with the corresponding pretrained parameters. All model and training details can be found in Table 7. The comparison results of word-based and subword-based models are shown in Table 8.

Semi-supervised learning with SDGMs. The main model (NXVAE) and training details are the same as in supervised learning. Besides M1+M2, we also compare with AUX (Maaløe et al., 2016) with the two decoder types. The training details are shown in Table 9. Regarding the decoding of GRU, all conditional latent variables of $p_\theta(\mathbf{x}|\cdot)$ are fed as extra input at each decoding step (Xu et al., 2017).

We tune all semi-supervised models on $\mathrm{EN}_{\text{EN-DE}}$ with 32 labels in semi-supervised settings, and then apply it to all other languages and data sizes. We tune only one hyperparameter: the scaling factor β in the weight for the classification loss α in the original SDGM paper (Maaløe et al., 2016):

$$\alpha = \beta \frac{N_l + N_u}{N_l}$$

where N_l and N_u are labelled and unlabelled data point numbers. We tune β from $\{0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0\}$. We pick the β with the best dev. performance for each

[11] https://github.com/google/sentencepiece.

[12] Both word and subword-based models are trained with: https://github.com/google-research/bert.

[13] https://github.com/huggingface/transformers.

model, and randomly select one when there is a tie. Then we use such fixed β for all other experiments across different training data sizes and languages.

The results of AUX can be seen in Table 10 along with M1+M2 results from the original paper. The parameter size of each model is shown in Table 11.

D.2 mBERT Encoder

The supervised model (SUP-h) adds a single linear transformation layer on the pooled `[CLS]` representation of mBERT, and M1+M2+BOW adds the corresponding SDGM framework on the same mBERT output. Like BERT, as mBERT uses a shared WordPiece vocabulary across languages, the parameter size of the same model will be the same for each language. All model and training details along with parameter size can be found in Table 12.

For tuning the hyperparameter of M1+M2+BOW, different from LSTM encoder with VAE pretraining, we set α fixed to $\alpha = \beta$. We tune β on EN with 8 labels in semi-supervised settings with 5 trials from $\{0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0, 50.0\}$, pick the β with the best **average** dev. performance, and then apply it to all other languages and data sizes. We report the mean and variance over 5 trials, and the full results for both models can be seen in Table 13.

E Conditional document generation

Semi-supervised deep generative models can not only explore the complex data distributions, but are also equipped with the ability to generate documents conditioned on latent codes, which is another advantage over other semi-supervised models. We follow Kingma et al. (2014) by varying latent variable y for generation, and fixing $\mathbf{z}_2$ either sampled from the prior (Table 14) or obtained from the input through the inference model (Table 15), and generate sequence samples from the trained semi-supervised models M1+M2+BOW and M1+M2+GRU.[14]

Overall, all models generate words or utterances directly related to the class, with the class labels among top nouns generated by BOW models, and subjects/objects in sentences from GRU are also pertaining to corresponding classes. However, we also observe that the utterances in GRU are not fluent with many repetitions. We argue that it is caused by the high proportion of UNK in the training corpus that makes the sequence generation harder, supported by the fact that the most probable word in all BOW decoders is always UNK.

[14] All models are treined on $\text{EN}_{\text{en-fr}}$ with 128 labelled data.

Hyperparameter	NXVAE	SNXVAE
vocabulary size	4e4 (EN), 5e4 (DE, FR)	1e4
embedding size	300	300
embedding dropout	0.2	0.2
encoder	BiLSTM	BiLSTM
encoder input dimension	300	300
encoder hidden dimension	600 for each direction	600 for each direction
encoder layer number	2	2
encoder dropout	0.2	0.2
discriminator configuration	[2 $\times$ 600, 1024, leakyrelu, 1024, 1]	[2 $\times$ 600, 1024, leakyrelu, 1024, 1]
inferer ($\mathbf{h}$ to μ or $\log\sigma$) configuration	[2 $\times$ 600, 300, batchNorm, relu, 300]	[2 $\times$ 600, 300, batchNorm, relu, 300]
$\mathbf{z}$ dimension	300	300
parameter size	41.8M (EN-DE and EN-FR)/ 44.9M (DE-FR)	17.8M
running time	$\sim$ 1 day	less than 1 day
tie embeddings of encoder and decoder	True	True
sentence length threshold	median length of training data	median length of training data
α in Equation 3	{0.01, 0.02, 0.05, **0.1**, 0.2, 0.5, 1.0}	0.1
training epoch	500	500
early stopping	5 epochs on dev. negative likelihood	5 epochs
batch size	128	128
validate on dev.	every 4000 iterations	every 4000 iterations
optimiser	Adam	Adam
learning rate	5e-4	5e-4

Table 5: Model and training details of NXVAE.

Hyperparameter	BERTW	BERTSW
vocabulary size	84101	10005
hidden size	300	300
max position embeddings	512	512
hidden dropout prob	0.1	0.1
hidden activation	gelu	gelu
intermediate size	2100	1800
num attention heads	12	12
attention probs dropout prob	0.1	0.1
num hidden layers	12	11
parameter size	45.0M	19.1M
running time	$\sim$ 5 days	$\sim$ 3 days
max seq length	200	
max predictions per seq	30	
masked lm prob	0.15	
batch size	32	
optimiser	Adam	
learning rate	1e-4	
weight decay	0.01	
num train steps	1e6	

Table 6: Model and training details of BERTW and BERTSW.

Hyperparameter	BERTW/BERTSW	VAE-based
vocabulary	same as pretrained model	same as pretrained model
training epoch	5000	5000
early stopping	1000 epochs on dev. accuracy	1000
batch size	16	16
running time	$\sim$5.5h	$\sim$2.5h
sentence length	200	200
optimiser	Adam	Adam
learning rate	2e-5	5e-4
classifier	[input_dim, class_num]	[input_dim, 1024, leakyrelu, 1024, class_num]

Table 7: LSTM encoder with VAE pretraining: model and training details of MLDoc supervised document classification. The running time is calculated on $\text{EN}_{\text{EN-DE}}$ with 32 labelled data for all models.

EN-DE	EN				DE			
	32	64	128	FULL	32	64	128	FULL
BERTW	67.7	72.7	**84.6**	**91.8**	58.1	77.5	**89.2**	94.0
BERTSW	54.4	69.0	83.0	91.4	62.2	80.1	84.3	**94.1**
NXVAE-z_1	63.9	71.4	77.0	91.6	65.0	73.8	82.7	93.0
SNXVAE-z_1	68.9	**76.5**	79.2	90.3	69.0	79.4	85.5	91.7

Table 8: LSTM encoder with VAE pretraining: comparisons of word-based models and subword-based models for BERT and NXVAE in MLDoc supervised document classification. Word-based results are directly from the original paper.

Hyperparameter	M1+M2	M1+M2+BOW	M1+M2+GRU	AUX+BOW	AUX+GRU
training epoch	5000	5000	5000	5000	5000
early stopping	1000	1000	1000	1000	1000
best β	0.1	0.2	10.0	20.0	5.0
$\mathbf{z}_1$ dim	300	300	300	300	300
$\mathbf{z}_2$ dim	300	300	300	300	300
tie embedding	-	False	False	False	False
running time	$\sim$2h	$\sim$12h	$\sim$14h	$\sim$13h	$\sim$ 14.5h
GRU input dim	-	-	100	-	100
GRU hidden dim	-	-	50	-	50
GRU layers	-	-	1	-	1
GRU dropout prob	-	-	0.5	-	0.5

Table 9: LSTM encoder with VAE pretraining: model and training details of MLDoc semi-supervised document classification. The running time is calculated on $\text{EN}_{\text{EN-DE}}$ with 32 labelled data for all models.

EN-DE	EN				DE			
	32	64	128	FULL	32	64	128	FULL
M1+M2	56.6	67.1	70.3	-	52.6	67.2	76.8	-
M1+M2+BOW	**79.8**	**81.7**	87.2	-	70.5	79.6	**89.7**	-
M1+M2+GRU	75.3	79.4	84.9	-	75.1	80.0	87.1	-
AUX+BOW	78.8	**81.7**	**87.7**	-	**75.2**	**86.2**	89.3	-
AUX+GRU	74.8	80.0	85.1	-	72.2	76.5	87.6	-
EN-FR	EN				FR			
	32	64	128	FULL	32	64	128	FULL
M1+M2	71.8	73.5	76.5	-	66.2	78.7	79.7	-
M1+M2+BOW	**81.0**	**85.5**	**88.2**	-	80.3	**86.0**	**88.8**	-
M1+M2+GRU	75.3	81.4	83.8	-	**80.7**	82.3	87.4	-
AUX+BOW	79.8	83.4	87.1	-	80.4	85.7	88.1	-
AUX+GRU	78.3	81.3	86.6	-	**80.7**	83.2	85.4	-
DE-FR	DE				FR			
	32	64	128	FULL	32	64	128	FULL
M1+M2	59.1	70.6	75.4	-	48.5	57.4	60.7	-
M1+M2+BOW	**78.0**	**83.2**	88.3	-	**81.4**	**84.5**	**88.4**	-
M1+M2+GRU	74.6	80.5	86.2	-	66.2	77.2	81.9	-
AUX+BOW	74.6	82.9	**89.0**	-	73.9	79.5	82.1	-
AUX+GRU	70.7	79.5	80.3	-	67.3	81.0	83.6	-

Table 10: LSTM encoder with VAE pretraining: test accuracy of AUX models. The header numbers denote number of labelled training data instances. The best results are in bold. Other results related to M1+M2 are directly from the original paper.

	EN	DE	FR
EMBEDDING MODELS	25.8M	28.8M	28.8M
NXVAE-h	26.8M	29.8M	29.8M
NXVAE-z_1	25.8M	28.8M	28.8M
SNXVZE-z_1	16.8M	16.8M	16.8M
BERTW	45.0M	45.0M	45.0M
BERTSW	19.1M	19.1M	19.1M
M1+M2	0.9M	0.9M	0.9M
M1+M2+BOW	38.5M	44.5M	44.5M
M1+M2+GRU	43.2M	49.2M	49.2M
AUX+BOW	43.8M	49.8M	49.8M
AUX+GRU	48.5M	54.5M	54.5M

Table 11: LSTM encoder with VAE pretraining: parameter size of all supervised and semi-supervised models. The difference between NXVAE-based models and BERTW is caused by language specific vocabulary of NXVAE, where only one vocabulary is used for **mono-lingual** document classification.

Hyperparameter	SUP-h	M1+M2+BOW
vocabulary size	1e5	1e5
$\mathbf{z}_1$ dim	768	768
$\mathbf{z}_2$ dim	768	768
tie embedding	True	True
best β	-	10.0
training epoch	500	500
early stopping	100 epochs on dev. accuracy	100
batch size	4	4
running time	$\sim$1h	$\sim$5h
sentence length	200	200
optimiser	Adam	Adam
learning rate	2e-5	2e-5
classifier	[768, class_num]	[768, class_num]
parameter size	178M	185M

Table 12: mBERT encoder: model and training details of MLDoc document classification. The running time is calculated on $\text{EN}_{\text{EN-DE}}$ with 8 labelled data for both models.

	Model	8	16	32	1K
EN	SUP-h	42.2 (4.7)	68.9 (9.7)	82.4 (3.0)	94.2 (0.8)
	M1+M2+BOW	**63.5** (12.8)	**77.1** (2.8)	**85.0** (1.5)	-
DE	SUP-h	55.9 (9.9)	63.5 (10.2)	81.5 (6.5)	95.0 (0.3)
	M1+M2+BOW	**63.2** (11.5)	**70.9** (6.3)	**87.5** (2.6)	-
FR	SUP-h	38.6 (3.3)	55.9 (11.4)	78.5 (3.0)	93.5 (0.7)
	M1+M2+BOW	**42.4** (4.6)	**66.4** (9.1)	**81.1** (2.7)	-
RU	SUP-h	49.4 (6.0)	53.8 (2.6)	68.2 (5.2)	87.2 (0.4)
	M1+M2+BOW	**51.5** (6.0)	**61.5** (4.6)	**72.6** (2.3)	-
ZH	SUP-h	63.4 (12.5)	70.7 (6.5)	81.2 (3.9)	91.1 (0.1)
	M1+M2+BOW	**65.1** (11.1)	**77.1** (2.4)	**81.4** (3.8)	-

Table 13: mBERT Encoder: MLDoc average test accuracy for both SUP-h and M1+M2+BOW models. The variance is in the bracket after the mean score. The first row denotes the number of labelled instances. The best results are in bold.

Class	M1+M2+BOW	M1+M2+GRU
C	1: UNK, industry, credibility, agreement, ticket, co, decision, concept, ltd, people, sale, government, market, president, designations, minister, firm, plans, partner, deal	1: the bank said it lump of the united ... the new girls ltd said the concept ... the new extraordinary and the concept ... said the statement ...
	2: UNK, ticket, year, shares, days, results, age, net, demand, securities, period, stock, concept, construction, bank, programme, procedure, statement, value, commission	2: the bank of organisation said on thursday that it had revoked by the first girls ... first year to ...
E	1: UNK, finance, market, loophole, budget, surprise, bank, basis, issue, government, system, exchanges, committee, municipal, world, securities, holding, net, confidence, minister	1: the international basic fund said on acknowledged that it said on publish to vote on publish to a bank said on publish ...
	2: UNK, ticket, city, escalation, finance, bank, budget, concept, revenue, net, price, sale, trade, tax, prices, markets, series, rate, fund, pack	2: the bank of submitting on publish florence said on acknowledged that ... it said on publish that ... to the new coherent said on acknowledged to bumping the bank said the bank ...
G	1: UNK, government, state, minister, delay, pension, work, president, plans, summit, ticket, people, procedure, conference, ambassador, country, talks, opposition, nations, house	1: the president remarkable said on thursday it surprise of ethnocide arrival the infidels of the islamic of the waterway the bank was ...
	2: UNK, state, president, war, police, office, authorities, problem, information, result, country, rights, committee, city, people, biodiversity, justice, health, securities, issue	2: the summit in the authors and a virtual geological and the first time of the first party of the first time of ...
M	1: UNK, ticket, phase, market, government, minister, markets, banks, bank, budget, floor, points, rate, traders, procedure, strength, economy, finance, prices, loophole	1: the database distinctions the market closed sharply entire on thursday on acknowledged ...
	2: UNK, markets, market, stock, loophole, points, trade, shares, ticket, corporate, speaker, issues, fund, bank, group, exchanges, results, anticipation, companies, surprise	2: the following of the the the ries and not have embargo costs unveiling on publish pleading a impact of the japanese ... market and a bank was to be of the bank ...

Table 14: Generated samples from M1+M2+GRU (BOW) for class C (*Corporate/Industrial*), E (*Economics*), G (*Government/Social*), and M (*Markets*). We randomly sample $\mathbf{z}_2$ from the prior while varying y.

1: Fiat shares lost nearly two percent on Wednesday, slipping below the psychologically important 4,000 lire level in thin trading on a generally easier Milan Bourse, traders said. "The stock has gradually lost ground but without any major sell orders. At the moment there just isn't any interest in Fiat," one trader said. At 1439 GMT, Fiat was quoted 1.99 percent off at 3,980 lire, after touching a day's low of 3,970 lire, in volume of just under four million shares. The all-share Mibtel index posted a 0.47 percent fall. – Milan newsroom +392 66129589 (E)

1: fiat shares lost nearly two percent on UNK slipping below the psychologically important UNK lire level in thin trading on a generally easier milan UNK traders UNK UNK stock has gradually lost ground but without any major sell UNK at the moment there just UNK any interest in UNK one trader UNK at UNK UNK fiat was quoted UNK percent off at UNK UNK after touching a UNK low of UNK UNK in volume of just under four million UNK the UNK UNK index posted a UNK percent UNK UNK milan UNK UNK UNK

2: The top prosecutor of Honduras said on Wednesday that his country is a haven for money laundering. "In Honduras it's easy to launder money, the system allows it," Edmundo Orellana told reporters. "It's permitted because there is no law in Honduras that obligates a Honduran to explain the origin of his wealth." Honduran authorities estimate that $300 million in illegal drug profits is laundered through the country each year. Money laundering is not classified as an offence in Honduras, although legislators have been working on a bill to outlaw it since last year. (G)

2: the top prosecutor of honduras said on wednesday that his country is a haven for money UNK UNK honduras UNK easy to launder UNK the system allows UNK UNK UNK told UNK UNK permitted because there is no law in honduras that UNK a honduran to explain the origin of his UNK honduran authorities estimate that UNK million in illegal drug profits is laundered through the country each UNK money laundering is not classified as an offence in UNK although legislators have been working on a bill to outlaw it since last UNK

Class	M1+M2+BOW	M1+M2+GRU
C	1: UNK, ticket, profit, concept, net, market, escalation, share, results, shares, delay, group, revision, profits, period, misery, statement, bank, key, procedure	1: the bank said on fourthly it has inject requirement of the first group of ...
	2: UNK, concept, ticket, group, market, shares, delay, president, stock, companies, bank, statement, government, stake, price, co, state, girls, meeting, ltd	2: the bank of organisation said on acknowledged that it had a meeting ...
E	1: UNK, ticket, escalation, inflation, key, revision, delay, period, floor, consumer, bank, contexts, result, instance, show, market, level, government, gross, price	1: the bank of submitting on publish florence said on acknowledged that it said on publish that ... the new coherent ... to the bank ...
	2: UNK, ticket, bank, government, finance, market, state, budget, tax, minister, rate, delay, debt, issue, trade, investment, surprise, policy, sale, procedure	2: the international basic fund said on acknowledged that it said on publish ... to vote on acknowledged to a bank ...
G	1: UNK, world, ticket, policies, time, surprise, procedure, demand, campaigns, group, team, president, match, communities, place, minister, bank, government, number, relief	1: the ana police said acknowledged it had a tackling ...
	2: UNK, president, government, people, state, minister, pension, police, designations, meeting, talks, opposition, leaders, country, security, result, statement, authorities, peace, summit	2: the president remarkable said on thursday that it surprise of ethnocide arrival infidels of her wines of her recall and the white house of ...
M	1: UNK, shares, ticket, contexts, touch, market, stock, points, escalation, share, traders, phase, immigrants, procedure, price, pledges, revision, agriculture, group , level	1: the bank of the settlement following the following vocational meda of the deal was delay ... and the market ...
	2: UNK, market, ticket, bank, traders, anticipation, delay, procedure, trade, prices, immigrants, rate, government, money, meda, escalation, demands, exchange, points, reallocation	2: the bank of the settlement following the following vocational value of the relative gains of ...

Table 15: Generated samples from M1+M2+GRU (BOW) by varying class label y. We take $\mathbf{z}_2$ from the input examples shown above. For each example, the first is the original document with the class label in the end, and the second is the real input to the system.