Zero-shot Classification of News Title

YenTing Lin
Department of Information Management
National Taiwan University
Taipei, Taiwan
b04705026@ntu.edu.tw

YenJung Hsu
Department of Information Management
National Taiwan University
Taipei, Taiwan
b04702077@ntu.edu.tw

ChiYu Lin
Department of Information Management
National Taiwan University
Taipei, Taiwan
b04705023@ntu.edu.tw

YuLun Li
Department of Information Management
National Taiwan University
Taipei, Taiwan
b04705025@ntu.edu.tw

ABSTRACT

This work introduce a multi-label zero-shot learning model that can classify sentences into classes even if the class is absent from training data. We transform multi-label problem into multiple binary classification to expand the possibility of classifying into unseen label. We combine machine reading comprehension techniques to capture semantic relationship to aid the model learning mapping between classes and sentences in embedding space.

KEYWORDS

zero shot learning, text categorization, attention, news

ACM Reference Format:
YenTing Lin, YenJung Hsu, ChiYu Lin, and YuLun Li. 2019. Zero-shot Classification of News Title. In *Proceedings of NTU IM IRTM Workshop.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Zero-shot learning refers to a task that establishes a model able to classify instances of an unseen class. Zero-shot learning increases the capacity of a classifier in dealing with a situation in which unseen classes are unavailable.

We remodel the problem of multi-class multi-lable classification problem in to multiple binary classification problem, where the model learn the "`relatedness`" between a sentence and single tag.

2 MODEL

In this section, we first formulate the classification problem and define our neural network model with three major components: embedding layer, attention layer and encoder block.

NTU IM IRTM Workshop, Jan 15, 2019,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.1 Problem Formulation

The sentence classification in our work is defined as follows. Given a sentence with n words $S = \{s_1, ..., s_n\}$ and a tag (or class, interchangeably) with m words $T = \{t_1, ..., t_m\}$, our model outputs $P(r = 1|T, S)$, the probability of $r = 1$ where r is the binary random variable denoting whether the given sentence and tag is related or not.

Compared with our formulation, multi-class problem and multi-label problem have the following drawbacks. Multi-class problem $P(T|S)$ is given a sentence and outputs a probability distribution over multiple classes. In multi-lable problem, the classifier output $P(r_{T_i}|S)\ \forall i \in \{1, ..., unique\ classes\}$. There are two obstacles. First, the number of classes in this work is enormous, making the model impossible to learning anything useful. Secondly, the unseen class would never be in the distribution, and the model cannot achieve zero-shot learning.

2.2 Architecture Overview

Our architecture design is inspired from the neural machine reading comprehension field and question answer task. We use bi-directional attention in sentence-tag attention layer to attraction contextual-aware representation and encoder block consist of convolution operation, self-attention mechanisms, and fully connect layer to capture higher level global information. Encoder block is designed specifically to replace auto-regressive network (like GRU, LSTM) with similar effects but much lower training time. At the top of our model is the relatedness probability of the sentence and tag.

Baseline model only haves fully-connected layers after embedding layer. Comparison of performance is at 4.3

2.3 Embedding Layer

Chinese words can be composed of multiple characters but with no space appearing between words. Chinese word segmentation is standard way for pre-processing. However, Chinese word segmentation will increase the data sparseness in terms of word occurrence. When inference, any unseen classes after Chinese word segmentation would be mapped to Out-of-Vocabulary (OOV), and all unseen classes would be the same in embedding space, which is impossible for our zero-shot learning model to learning the relatedness. In order to generalize well to unseen classes, we do not adopt Chinese

word segmentation method and treat each character as a single word after filter out punctuation.

The text embedding features are extracted from the state-of-the-art text representation model BERT[3]. The contextual embedding, generated from the hidden layers of the last four Transformer of the pre-trained "`BERT-Base, Chinese`" https://github.com/google-research/bert/blob/master/multilingual.md. To utilize the next sentence prediction task in BERT, we consider the sentence(, title) and class(, tag) as first and second sentence, respectively. We take arithmetic mean across four hidden state of each word to avoid curse of dimension and the number of parameters in downstream network. In our experiment, concatenation of feature across four hidden state does not increase model performance but devouring precious GPU memory. In training time, we did not back-propagate into BERT due to GPU memory constraints.

2.4 Sentence-Tag Attention Layer

The bi-directional attention mechanism to introduced in [10], and became a common practice in machine reading comprehension task[2][11][12]. Sentence-Tag Attention layer is fusing information from the sentence and tag.

We use $S \in \mathbb{R}^{d \times n}$ and $T \in \mathbb{R}^{d \times m}$ to denote the sentence and tag embedding, where d is the embedding dimension. Both direction of attention is derived from the similarity matrix $H \in \mathbb{R}^{n \times m}$.

$$H_{ij} = \mathbf{w}^T [S_{:i}; T_{:j}; S_{:i} \odot T_{:j}] \in \mathbb{R}$$

is the trilinear function[10] to capture the similarity of corresponding word in sentence and tag word, where $\mathbf{w}$ is trainable vector, **;** is concatenation along row, and $\odot$ is element-wise multiplication. $A = \overline{H}\ T^T \in \mathbb{R}^{n \times d}$ is the sentence-to-tag attention and $B = \overline{H}\ \overline{\overline{H}}\ S^T \in \mathbb{R}^{n \times d}$ is the tag-to-sentence attention, where $\overline{H}$ is row normalized by softmax from H to represent the attention weight on each tag word to one sentence word, and $\overline{\overline{H}}$ is column normalized by softmax from H to represent the attention weight on each sentence word to one tag word.

Outputs of this layer are $A\ B\ S$.

2.5 Encoder Block

The input of this layer at each time step is $[c; a; c \odot a; c \odot b]$, where a and b are row in attention weight matrix A and B [12]. Encoder block is composed of three component: [convolutional layer, self-attention layer, fully-connect layer].

Convolution layer is to extract local feature, and depthwise separable convolution [5] is preferred than vanilla convolution since more efficient and with better generalizability [6]. Multi-head self-attention layer is adopted in order to (, theoretically) fuse information and output summarization of upstream tensor to downstream network. The number of heads is 4 in this work purely because GPU memory limitation. Fully-connected layer learns the non-linear combination of upstream features.

Each input to one of three component in encoder block is normalized by layernorm[1]. The output of encoder block is added with skip-connection (aka residual)[4], which is widely used and is known for stability of gradient when training deep network[8]. Our model pass through encoder block twice with the shared weights in consideration of number of parameters.

3 DATA

Our data is crawled from two news website: UDN https://udn.com/news/index, ETTODAY https://www.ettoday.net/. For the purpose of Search Engine Optimized (SEO) many news website add several (averaged around 2 to 7) tags for each news. We download news titles and corresponding tags from 2011 to 2018. Our training data are from 2011 to 2017. Validation data is randomly selected from training data. Testing data is all in 2018 to test the generalizability of our model. 1 shows the dataset distribution. 2 contains some samples in dataset.

Table 1: News Tags Dataset

Year	# News	Proportion (%)
2012	38,978	7.09
2013	53,293	9.69
2014	32,943	5.99
2015	52,703	9.58
2016	105,780	19.24
2017	120,643	21.94
2018	145,606	26.48

Table 2: Data Examples

News Title	Tags
2017普悠瑪花蓮出軌報告曝光，主因：枕木嚴重腐朽、軌道鬆動浮起	普悠瑪 新馬車站 台鐵 宜蘭 出軌
靜脈注射太快害昏迷喪命，榮總屏東分院判賠297萬	醫師 手術 膝關節

4 EXPERIMENTS

4.1 Training

When training, for each news title, the positive tags are the tags downloaded while negative tags are randomly chosen from tags in other news excluding overlaps with positive tags. Concerning data imbalance issue, the number of negative tags and that of positive tags are purposely made the same. There are more than 4 million sentence-tag pair in training data. On each epoch end, negative tags will be re-generated.

We use PyTorch[9] to implement the model. The model is trained with Adam optimizer[7] minimizing binary cross entropy loss. The learning rate is scheduled to scale down 70% each 3 epochs from initial learning rate equal 0.001. It took 40 to 50 hrs to train a single model on NVIDIA TITAN V until converge.

4.2 Test

Testing data is all news in 2018, roughly 26% in all data we have. Negative tags are generated the same as training but fixed at the beginning of this work for equity of comparison.

For classifying a sentence into a list of tags, we separately calculate the probability of each tag and the sentence, and classify the tag is related to the sentence if the probability of relatedness is above a threshold (, 0.5 in this word). The threshold is a hyper-parameter, and can be adjusted in different circumstances.

4.3 Result

We compare our models on news, including positive sentence-tag pairs and negative ones. We chose accuracy as our metric and results is list at 3. There are several conclusion based on experiments we ran. First, thanks to huge amount to data, high dropout rate is of little effect, so we chose 0.3 as default. Secondly, higher number of times passing encoder block would not collapse model training due to shared weights and residual connection. Finally, 4 self-attention head is preferred than 8, since double parameters led to the curse of dimensionality.

Table 3: Performance Result

Model	Accuracy(%)	Note
Baseline	79.2398	
Attention + Encoder	84.7498	Enc*2, 4 Att Head, Dropout 0.3
Attention + Encoder	83.8474	Enc*4, 8 Att Head, Dropout 0.3
Attention + Encoder	81.2437	Enc*4, 8 Att Head, Dropout 0.5

5 CONCLUSION

We extended multi-lable multi-class text classification problem into zero-shot text classification problem, combined state-of-the-art embedding model BERT as feature extractor, and several machine reading comprehension techniques to boost our performance.

Our work, different from other classification model, can be fully customized to any users with their own interest of topics without re-training or fine-tuning models, and can be a valuable generalized sentence, news, article classifier.

Further work would be extending zero-shot learning problem to full text classification (instead of news title) with focus on dealing with semantic understanding and reasoning in full text.

6 MEMBER CONTRIBUTION

Name	Student ID	Contribution
YenTing Lin	B04705026	Model, Report, Slide
YenJung Hsu	B04702077	Report, Slide
ChiYu Lin	B04705023	Data collecting
YuLun Li	B04705025	Demo page

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1870–1879. https://doi.org/10.18653/v1/P17-1171

[3] 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* (2018).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[6] Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. 2017. Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059* (2017).

[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[8] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*. 6391–6401.

[9] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[10] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional Attention Flow for Machine Comprehension. *CoRR* abs/1611.01603 (2016). arXiv:1611.01603 http://arxiv.org/abs/1611.01603

[11] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making Neural QA as Simple as Possible but not Simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics, 271–280. https://doi.org/10.18653/v1/K17-1028

[12] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *CoRR* abs/1804.09541 (2018). arXiv:1804.09541 http://arxiv.org/abs/1804.09541