Machine Intelligence Research
www.mi-research.net

21(2), April 2024, 283-293
DOI: 10.1007/s11633-023-1424-x

Text Difficulty Study: Do Machines Behave the Same as Humans Regarding Text Difficulty?

Bowen Chen [1] Xiao Ding [1] Yi Zhao [1] Bo Fu [2]
Tingmao Lin [2] Bing Qin [1] Ting Liu [1]

[1] Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin 150001, China

[2] Fundamental Technology Center, China Construction Bank Financial Technology Co., Ltd., Beijing 100033, China

Abstract: With the emergence of pre-trained models, current neural networks are able to give task performance that is comparable to humans. However, we know little about the fundamental working mechanism of pre-trained models in which we do not know how they approach such performance and how the task is solved by the model. For example, given a task, human learns from easy to hard, whereas the model learns randomly. Undeniably, difficulty-insensitive learning leads to great success in natural language processing (NLP), but little attention has been paid to the effect of text difficulty in NLP. We propose a human learning matching index (HLM Index) to investigate the effect of text difficulty. Experiment results show: 1) LSTM gives more human-like learning behavior than BERT. Additionally, UID-SuperLinear gives the best evaluation of text difficulty among four text difficulty criteria. Among nine tasks, some tasks′ performance is related to text difficulty, whereas others are not. 2) Model trained on easy data performs best in both easy and medium test data, whereas trained on hard data only performs well on hard test data. 3) Train the model from easy to hard, leading to quicker convergence.

Keywords: Cognition inspired natural language processing, psycholinguistics, explainability, text difficulty, curriculum learning.

Citation: B. Chen, X. Ding, Y. Zhao, B. Fu, T. Lin, B. Qin, T. Liu. Text difficulty study: Do machines behave the same as humans regarding text difficulty? *Machine Intelligence Research*, vol.21, no.2, pp.283–293, 2024. http://doi.org/10.1007/s11633-023-1424-x

1 Introduction

Recently, there has been large progress in the field of natural language processing (NLP). The emergence of pre-trained models has achieved SOTA performance on various tasks, which gives comparable or even outperforms human performance on certain tasks showing that current models are capable of showing human-level task performance on a broad range of tasks. While such progress is made, we still know little about the fundamental mechanism of current neural networks. The performance of such models is mainly due to the large-scale pre-training, and specific fine-tuning is needed. One thing we have to notice is that it seems that those models are driving far away from the human mechanisms that we understand. In the development of neural networks, the neuron targets mimic human nerves, the convolutional neural network (CNN) is inspired by the cat′s vision signal processing[1] while the recurrent neural network (RNN) family is inspired by human sequence processing which usually works in a recurrent way. However, it seems hard to tell the connection between current transformer-based pre-trained models and human intelligence, in a way we could say that it is not inspired by human mechanisms. Certainly, being inspired by humans is not necessary for machine learning, but we expect the model to have human-level intelligence. Especially, for natural language processing, we expect the model to have human-like reading performance and behavior. One of the most obvious phenomena in language study is the effect of text difficulty, which is that when human learns a language, human prefers to start with the easy text. However, in the training of current models, the effect of text difficulty is a less focused area. In this manner, we investigate whether the model behaves like humans regarding text difficulty.

Previous research in psycholinguistics shows language learners perform better on language tests when they start from easy sentences[2–4]. Additionally, making sentences easier is also an important education method to teach language in real-world[1,2]. This leads to a natural question that whether the machine behaves like a human with regard to the text difficulty? Undeniably, current AI models are insensitive to text difficulty. However, little

Research Article
Special Issue on Commonsense Knowledge and Reasoning: Representation, Acquisition and Applications
Manuscript received on November 4, 2022; accepted on February 7, 2023; published online on January 12, 2024
Recommended by Associate Editor Zhi-Yuan Liu
Colored figures are available in the online version at https://link.springer.com/journal/11633
© Institute of Automation, Chinese Academy of Sciences and Springer-Verlag GmbH Germany, part of Springer Nature 2024

[1] http://www.weeklyreader.com

[2] http://www.corestandards.org

attention has been paid to investigating the effect of text difficulty on models.

In this paper, we investigate how text difficulty effects models. Specifically, we aim to answer the following questions:

1) In which criteria of text difficulty, neural-based or feature-based, or information-theory-based way, the model could give better human learning matching (HLM) performance?

2) In which model type, a transformer[5]-based or RNN[6] based model, the model could give better HLM performance?

3) In which kind of task, classification or regression task, etc, the model could give better HLM performance?

To answer the above questions, in this paper, we propose the human learning matching index (HLM Index) which considers the model′s performance when tested on different text difficulties of each dataset. We examed a broad spectrum of NLP tasks covering most NLP task types. Additionally, we also discussed how the text difficulty affects the model′s training process. We further explored how difficulty transfer between datasets affects model performance.

2 Related works

The paper′s research relates to the following topics:

2.1 Curriculum learning

Since we investigated the effect of text difficulty in this research, it is also important to notice the related research on curriculum learning[7]. Curriculum learning is an area that tries to inject human learning behavior into the training of model which is that the human learns with easy data to quickly understand the concept of the task and then gradually learns the hard data based on the learned general concept to improve its expertise on certain tasks.

The curriculum learning tries to bring such human learning behavior into the training of the model. In curriculum learning, there are several major problems in this area. First is the design of difficulty criterion which aims to accurately classify the data into different difficulties, the following research with the classification of data difficulty would be the curriculum selection method which decides how to mix different training examples with different difficulties. Then research on curriculum learning schedulers is also important in which judges when to stop the training and when to switch between training examples and when to stop. In our research, since we study the text difficulty, the research is related to the design of the difficulty criterion. Basically, there are two major ways to decide the difficulty of a model. One is based on the inherent learning statistics given certain training cases. For example, the loss of a certain training case can be thought of as a difficulty measurement from the model perspective, which means a higher loss caused by a case means that this case is harder. Another evaluation criterion is based on the training examples themselves in which the difficulty is based on certain features of training examples like the length of sentences, etc.[8]

2.2 Cognition-inspired NLP

Cognition-inspired NLP is a long and established research area in NLP[9]. For example, the research on eye movement and human reading is a long-discussed topic in the field of language processing.

Those researches diverge in two ways. One way is to utilize the cognition data like eye-tracking or EEG signals to improve the model′s performance on various tasks[9–11]. The research utilizes those data with the target to inject certain human behavior into the model to make the model more human-like. Research on those areas is proved to be effective for both NLP and cognition tasks. Ranging from the utilization of human eye-tracking data to improve performance on the named entity recognition (NER) task to sentiment classification. The usage of those works proved that the cognition data could be helpful for a broad range of NLP tasks.

In another direction, there are also lots of works using the cognition data to do an analysis of current NLP models to see whether they could naturally give human-like behavior[12–15]. Those researches are mainly based on the explanation of NLP models with a focus on how humans and models are the same or different. For example, Hollenstein and Beinborn[16] discuss the relative importance of words in both NLP models and human attention. Additionally, Merkx and Frank[17] discuss whether human sentence processing is based on either recurrent or attention mechanism based on the study of sentence perplexity of both the long short-term memory (LSTM) model and transformer model and conclude that human sentence processing is working in a way more like the attention mechanism rather than recurrent processing way like LSTM which might help to explain why transformer model is more effective than LSTM model since human sentence processing also works in attention mechanism.

We define our work as both related to curriculum learning and cognition-inspired NLP methods. In the former one, since the proposed research exams which criteria could lead to human-like behavior, which might help to research in the curriculum learning to find more suitable difficulty criterion. In the latter area, the proposed research discusses how the model matches humans regarding text difficulty, which in a sense is the exploration of text study.

3 Text difficulty

Since we investigate the effect of text difficulty on the performance and other various factors for the training of the neural networks. It is also important to introduce the difficulty criterion that we use in the research. The difficulty criterion used are:

1) Flesch-Kincaid score[18]: This criterion is one of the most common and easy methods to compute the readability of a text which reflects how easy a text is to understand for human readers. Given a document d, the number of sentences, words, and syllables are d_s, d_w and d_l, then the Flesch-Kincaid score is computed using the equation below:

$$Flesch(d) = 206.835 - \frac{1.015 d_w}{d_s} - \frac{84.6 d_l}{d_w} \tag{1}$$

where 206.835, 1.015, and 84.6 are empirical values from the original paper. Those hyperparameters of the Flesch score are empirical values taken from actual reading ability assessment and are largely accepted as a common value for the Flesch-Kincaid score which is used as a readability assessment for technical manuals and soon developed in general fields to decide difficulty of a text. A text with more difficulties will have a lower score whereas an easy text receives a higher score, which means that a text with a higher score is easier to understand.

2) Neural evaluation: For the neural criteria, the model is trained on datasets like Weebit[19], one-stop corpus[20], in which human experts judge the text difficulty, and manually classify them into different difficulty level like elementary, intermediate and advanced level, reflecting the amount of work that is required to understand the sentence. Those datasets are originally used in the education area that helps language learners to decide their learning and reading schedule. Taking advantage of those labelled text difficulty data, we fine-tune BERT[21] on the one-stop corpus to rank the text difficulty, hoping the model might learn the human expert's judgment of text difficulty which might be more accurate and reflect the mental processing load while reading this sentence.

3) Uniform information density (UID[22–25]) hypothesis: UID is based on the information theory[26], which means the cognitive processing load of words is proportional to its log-probability and the ideal distribution of information should be uniform. A sentence that follows UID will not be cognitively taxing for the reader, so a sentence with uniformly distributed information is easier to understand and read. However, the actual implementation of UID varies based on different interpretations of such theory. We test the two most popular UID hypothesis implementations: UID super-linear (UID-SL) and UID variance (UID-Var).

i) UID super-linear:

$$UID(\boldsymbol{u})^{-1} = \frac{1}{N}\sum_{n=1}^{n} s(u_n)^k. \tag{2}$$

where k controls the strength of super-linearity. UID-SL suggests the text difficulty increases regarding the exponential sum of sentence surprisal. The k means that with higher k, the UID-SL will increase and grow exponentially more quickly. The basic idea is that the text difficulty is close to linearly increasing at the beginning and grows exponentially with the increase in sentence length.

ii) UID language-variance:

$$UID(\boldsymbol{u})^{-1} = \frac{1}{N}\sum_{n=1}^{n} (s(u_n) - \mu_{lang})^2 \tag{3}$$

where $s(u_n) \overset{def}{=} -log\ p(u_n|u_{<n})$ means the log-probability conditioned on its prior context in both equations. UID-Var suggests the text difficulty is decided by the variance between the sentence surprisal and the mean language-level surprisal u_{lang}. We follow the implementation of [27], in which the k is 1.25 and u_{lang} is 3.8845. The hyperparameter of UID-SL and UID-Var is taken from [27], as shown in Figs. 1 and 2. The hyperparameter of these two criteria is found to be the most expressive regarding explanation of human reading behavior like the eye-tracking data and linguistic acceptability data, which proves that these two hyperparameters reflect the actual human reading performance. Therefore, it is the best hyperparameter setting for UID-based difficulty evaluation.

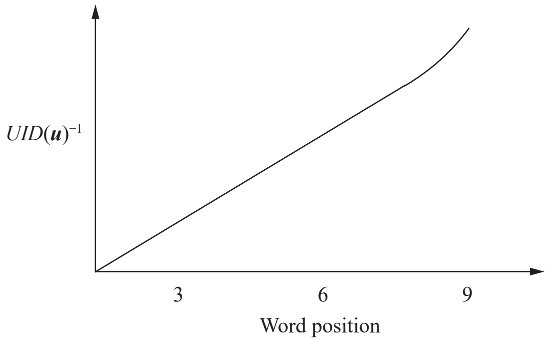


Fig. 1 Super-linear UID example

Undeniably, changing the hyperparameters in the Flesch score or UID leads to a different experiment result. But a different setting of hyperparameters is not supported by any previous research, thus evaluation of text difficulty based on different hyperparameters is less accurate, which leads to unsupported and unconvincing results.

4 Human learning matching index

We propose human learning matching index (**HLM Index**) to answer the above questions raised in the introduction section, which has 3 sub-indexes I_{task}, I_{model} and $I_{criteria}$.

We split each dataset into 3 text difficulty levels computed by different criteria, which corresponds to easy, medium and hard level. Given tasks $T = \{t_1, \cdots, t_j\}$, models $M = \{m_1, \cdots, m_k\}$ and criteria $C = \{c_1, \cdots, c_l\}$. Under task, t_o, criterion c_p and model m_q, a model

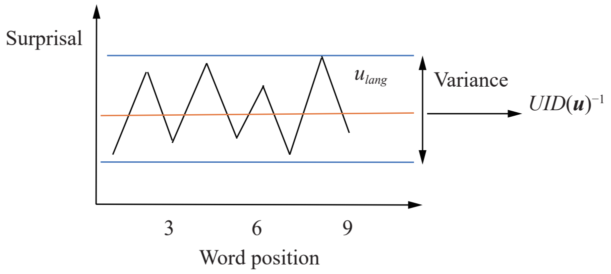


Fig. 2 Variance UID example

trained on each difficulty level has test performance on each difficulty level of the test set $P(t_o, c_p, m_q) = \{p_e^{opq}, p_m^{opq}, p_h^{opq}\}$. Then, we define a logical score function s:

$$s(p_e, p_i, p_s) = \begin{cases} 0.75, & \text{if } p_e \geq p_m \geq p_h \\ 0.375, & \text{if } p_e \geq p_h \geq p_m \\ 0, & \text{if } p_m \geq p_h \geq p_e \\ 0, & \text{if } p_m \geq p_e \geq p_h \\ -0.375, & \text{if } p_h \geq p_e \geq p_m \\ -0.75, & \text{if } p_h \geq p_m \geq p_e. \end{cases} \quad (4)$$

Then we compute the I_{model} index as following:

$$I_{model}(m_k) = \frac{1}{JL}\sum_{j=1}^{j}\sum_{l=1}^{l}(s(P(t_j, c_l, m_k))+ \\ 0.25\text{sgn}(s(P(t_j, c_l, m_k))f(STD(P(t_j, c_l, m_k)))). \quad (5)$$

Function (4) considers the logical relations between task performance on different difficulties in various datasets. We give the highest score if the p_e is higher than p_m and also the p_m is higher than p_h which corresponds to the task performance trained on the easy, medium and hard parts of the dataset. We value the task trained on the easy part as the most important and give a positive score when p_e is the highest. If the p_m is highest which means the performance trained on the medium dataset is the best, we give a neutral score in which the HLM index is neither positive nor negative. Additionally, if the p_h which is task performance trained on the hard dataset is the highest, we assign a negative score since it contradicts that the human gives the best performance when learned from easy data. In conclusion, the design of the HLM Index is based on the comparison of task performance in which the performance trained on easy text difficulty should be higher than others and followed by the model trained on medium difficulty. Then the model trained on hard difficulty should give the lowest performance.

STD means standard deviation, f is a sigmoid function. The input of both STD and s is the performance triplet in easy, medium and hard difficulty levels p_e, p_m and p_h, and sgn is the sign function that is to decide the sign of STD. The reason to include STD is to consider the dispersion of performance. As s only considers logical relation, the performance gap between different difficulty levels is ignored. However, the f is to prevent STD from dominating the HLM Index. By replacing m_k to t_j or c_l, we have I_{task} and $I_{criteria}$. The reason to choose 0.75, 0.375 and 0.25 is to make the HLM Index have a maximum and minimum value of ±1, which is straightforward to illustrate the results. Changing these parameters only affects the maximum value and does not affect the results.

To notice that the I_{task}, $I_{criteria}$, and I_{model} share the same computing method, we can obtain them through setting different conditions in (5). The reason is that we treat tasks, models, and criteria at the same level and consider them as 3 different dimensions of the HLM Index. Additionally, the HLM Index is based on the task performance on different dataset difficulties. By fixing any two variables of t, c, m, we could have the HLM Index of the remaining one.

5 Experiments

5.1 Datasets and models

In Table 1, to cover the spectrum of NLP tasks as much as possible, we select SST2 (single sentence binary classification), MRPC (pair-wise sentence binary classification), QNLI, RTE (multiple classification), and STS-B (regression) from GLUE benchmark[28]. In addition to these task types, we also include ROC story (multiple choice)[29], WIKITEXT2 (language modelling)[30], SQUAD 2.0 (QA)[31], CoNLL2003 NER (tagging)[32].[3] Choosing those tasks covers most task types of NLP which could help to examine the effect of text difficulty on a broad research area that explores the actual influence of text difficulty. The statistics of the dataset used in this paper are shown in Table 1.

Models in this paper are BERT and LSTM[33], which represent parallel and recurrent NLP models. BERT is the pre-trained language model using transformer architecture that achieves SOTA performance in various NLP tasks. The LSTM model is another popular sequence processing model in which the model processes sequences in a recurrent way that enables the correlation of different components of the sequence using a neural network.

5.2 Data processing and experiment implementation

5.2.1 Experiment setting

All the results are trained using five runs with different random seeds. To support the fully reproducible results of this research, we used [7 800, 8 321, 7 084, 8 147, 15 000] as the random seeds, and no special parameter initializa-

[3] We use WT2, SQUAD, CoNLL2003, ROC to denote WIKITEXT2, SQUAD 2.0, CoNLL2003 NER and ROC Story.

Table 1 Statistics of the datasets. The table includes the dataset name, train/validation/test size, and the dataset's task type.

Dataset	Train size	Validation size	Test size	Task type
SST2	4959	993	891	Sentiment classification
MRPC	56668	7374	7368	Paraphrase classification
QNLI	1500	469	1039	Natural language inference (classification)
RTE	11677	3934	9890	Textual entailment (classification)
STS-B	19808	3733	3739	Sentence similarity (regression)
ROC story	121200	13499	NA	Script prediction (multiple choice)
WIKITEXT2	7000	1034	NA	Language modelling
SQUAD 2.0	12059	1625	NA	Question answer (extraction)
CoNLL 2003	9502	1300	NA	Named entity recognition (tagging)

tion trick is implemented while initializing the LSTM, through which we want to keep the model as simple as possible so any trick would not affect the model's performance to obtain the most general performance on different tasks.

The results of LSTM for GLUE tasks are using 20 epochs with AdamW optimizer with a learning rate of [3×10^{-5}, 1×10^{-4}]. The LSTM trains on single sentence classification like SST2, which is based on a 2-layer LSTM and GloVe 300 dimension embedding, and we use spaCy tokenizer to tokenize sentences for LSTM. We use 2 LSTMs to encode sentences respectively for training on classification datasets between two sentences like MRPC, and the output is concatenated to pass the softmax function. For the language modelling task training, the sentence is tokenized and encoded using word embedding, and the output is asked to predict the next sentence.

We use a standard fine-tuned procedure for the BERT implementation with different fine-tuned heads. We also use AdamW optimizer with a learning rate of [1×10^{-5}, 3×10^{-5}, 5×10^{-5}] and report the best results on the development set.

5.2.2 Estimation of surprisal

To estimate the log-probability of the sentence while using UID criteria, we use the GPT2 as the language model, which is one of the current SOTA models of language modelling that is pre-trained on a large corpus using causal language modeling. Using the GPT2 could give an accurate estimation of log probability given the previous context.

6 Split of the text difficulty

Since we have mentioned the evaluation of text difficulty, it is also essential to notice how we split each dataset into three difficulty levels that correspond to easy, medium and hard level text difficulty.

For each criterion, we first input each sentence into the evaluation module introduced in Section 2. Each criterion will output a text difficulty for each sentence in the dataset. We then sort and reorder the text difficulty for each sentence, split it into three parts based on the sorted dataset. For example, data difficulty in the top 1/3 will be collected as the hard difficulty. However, as most datasets do not have a clear difficulty boundary between the different splits, we drop 2.5% data that is at the boundary of hard and medium difficulty levels in both hard and easy difficulty level data to make sure a clear difficulty boundary exists between each dataset's difficulty level. Additionally, we drop 1.25% data in the medium-level data on its boundary of easy and hard levels so that the data size of each difficulty level is the same. In this way, we manually create three different text difficulty level subsets for each dataset that shares the same data size. An example of the difficulty split of the SST2 task under the Flesch-Kincaid score is given in Fig. 3. We can see a manually created blank between each difficulty level to ensure a clear boundary exists among different difficulty levels.

6.1 Main results

For the experiment procedure, we test each model on each task on its different text difficulty level and then we collect their corresponding task performance. Those collected performances will be computed using the method introduced in Section 4 to produce the I_m, I_t and I_c, which is shown in Fig. 4. Due to limited page size, in Table 2, we use the Flesch, UID-SL, UID-Var and neural to represent Flesch-Kincaid, UID-Super linear, UID-variance and neural evaluation text difficulty criteria, respectively. We will analyze the results in Section 6.1.

6.1.1 Models HLM Index

Fig. 4(a) shows the I_m of LSTM and BERT on different tasks divided by different criteria.

From the results, the LSTM has a higher average I_m than BERT, which means that LSTM has a more human-like learning behavior. This contradicts Merkx and Frank[17], who find the transformer is more human-like with regard to learning human eye-tracking data, whereas we investigate it from learning behavior on multiple tasks. The results also contradict the idea that a model with higher task performance is more intelligent, as the BERT model is known for the competitive perform-

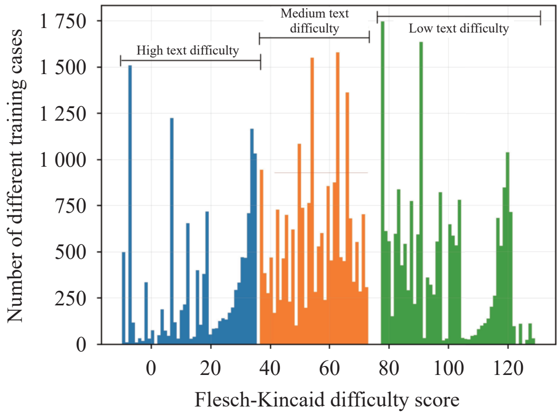


Fig. 3 Example difficulty split of Flesch-Kincaid score on SST2 task

ance with humans on several NLP tasks. The results suggest that a higher model performance does not guarantee that the model could give better HLM performance. Another point to notice is that both models could achieve a high or low score on some tasks, which indicates there are naturally some tasks that could make the model have a human-like behavior.

6.1.2 Tasks HLM Index

Fig. 4 (b) shows I_t of different tasks on different splits. From the results, SQUAD and WT2 have the highest average, which means the performance of the LSTM and BERT on these tasks are highly related to the text difficulty even under different criteria. Especially in WT2, the I_t reaches maximum 1 in both LSTM and BERT in UID-SL, UID-Var and neural criteria. However, the model gives a contrary score in the Flesch-Kincaid criterion, which means the sentence length and syllables are not a good criterion to split WT2 due to a strong tendency to give low difficulty to short sentences, which does not guarantee easiness. A high score in SQUAD indicates that the model′s performance in QA is highly related to the difficulty of the context and question.

Moreover, CoNLL2003 and QNLI have a close zero score, which shows that those two tasks are not highly related to the text difficulty. For CoNLL2003, text difficulty is not an obstacle. For NER tasks, entities are mainly represented as uppercase words in English, so the text difficulty does not influence the model or human to tag the entity since uppercase characters are a very strong indicator to tell whether a word is an entity or not. For QNLI, the model infers answers from context and question, in which the difficulty is based on the difficulty of inferring the answer based on the context rather than solely based on the difficulty of the text.

For other tasks, the I_t is positive, which means the model performance on these tasks is related to text difficulty but also intertwines with difficulty at a higher level like inference difficulty. Therefore, the text difficulty does affect the model performance in many tasks, but the definition of different tasks brings an inherent task difficulty to the dataset, so the difficulty is a mixed combination of both text and the task difficulty.

6.1.3 Criteria HLM Index

Fig. 4(c) shows the I_c of different criteria.

From the result, the UID-SL gives the highest match, whereas the neural-based method gives the lowest match. The UID-SL gives the highest score, indicating that the text difficulty is better evaluated by the super-linear function, which is close to linear at the beginning and increases exponentially with the sentence surprisal.

Surprisingly, the neural criteria yield the lowest match, which we originally expected the model to learn more sophisticated criteria from human experts as they could judge text difficulty from a higher level which may not be completely based on information or text features and reflects the true mental processing load. However, the results presented do not support this argument that the neural criteria give the lowest match score. Such a result may be due to insufficient training data, and the model is not effectively learning human judgment from human experts, and only learns surface features that cannot generalize well on unseen texts since the unseen text

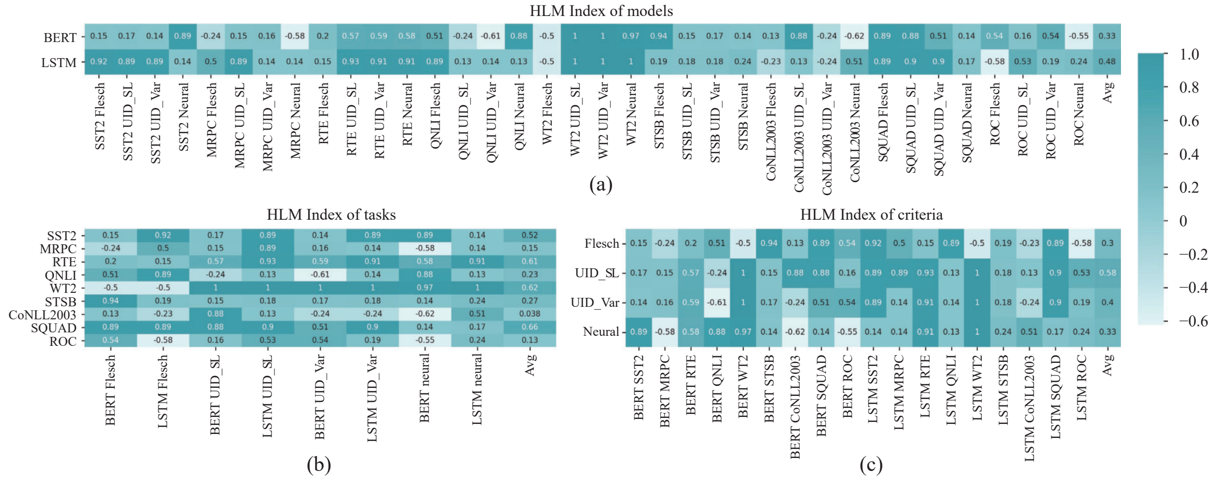


Fig. 4 HLM Index heatmap for models, criteria, and tasks. SL and Var represent super-linear and variance.

Table 2 Results of models on different difficulties evaluated by four criteria. Accuracy for SST2, RTE, QNLI, and ROC. F1 for MRPC, SQUAD and CoNLL2003. Perplexity for WT2. Pearson correlation for STS-B. The HLM Index is computed based on this table. CNLL is the abbreviation for CoNLL2003.

Criterion	Model	Train	SST2	MRPC	RTE	QNLI	WT2	STS-B	CoNLL	SQUAD	ROC
Flesch	BERT	Hard	92.17	**84.24**	59.56	89.62	**68.83**	83.61	89.92	81.53	85.27
		Medium	**92.66**	83.69	**63.18**	89.54	83.15	85.69	**90.03**	81.57	84.55
		Easy	91.63	84.01	61.15	**89.89**	89.44	**85.96**	89.89	**82.05**	**86.08**
	LSTM	Hard	84.69	81.23	54.24	60.25	**156.24**	76.79	87.78	53.52	67.22
		Medium	85.48	81.22	**55.30**	60.59	206.19	**78.31**	**87.91**	53.69	**66.90**
		Easy	**86.32**	**81.37**	54.94	**60.77**	475.91	75.43	87.48	**54.12**	65.36
UID-SL	BERT	Hard	91.17	**84.26**	60.57	**89.96**	92.88	85.98	90.15	81.49	84.24
		Medium	84.39	84.24	59.35	89.62	86.36	**86.86**	90.21	81.66	**84.80**
		Easy	**92.57**	83.46	**62.38**	89.70	**67.90**	85.87	**90.32**	**81.69**	83.60
	LSTM	Hard	85.18	81.34	54.31	60.42	522.19	73.86	87.48	53.20	66.96
		Medium	85.66	81.54	55.65	**60.67**	199.31	**76.15**	88.42	53.60	65.98
		Easy	**85.85**	**82.02**	**56.48**	60.44	**166.33**	74.99	**88.79**	**54.36**	**67.27**
UID-Var	BERT	Hard	91.76	84.24	60.28	89.95	93.78	**87.72**	**89.98**	81.83	85.24
		Medium	**92.52**	**84.92**	58.92	**89.99**	86.19	86.95	89.75	81.55	84.76
		Easy	92.20	83.37	**63.39**	89.53	**74.84**	86.18	89.91	**81.95**	**86.24**
	LSTM	Hard	85.25	81.58	54.08	61.72	407.09	75.51	**88.21**	53.34	65.30
		Medium	85.66	**81.71**	55.16	**62.06**	196.23	**76.39**	87.71	53.76	**67.82**
		Easy	**85.85**	81.28	**55.67**	61.69	**166.41**	74.04	87.91	**54.43**	65.22
Neural	BERT	Hard	92.36	**86.54**	61.22	89.76	90.79	85.72	**90.32**	81.62	**87.90**
		Medium	92.50	85.62	60.94	**89.84**	87.38	**85.75**	90.11	**82.10**	86.94
		Easy	**92.80**	84.74	**64.18**	86.53	**78.63**	85.12	90.07	81.81	84.77
	LSTM	Hard	85.36	81.27	55.38	**60.94**	304.66	75.40	87.62	53.14	69.18
		Medium	**85.82**	**81.73**	56.34	60.72	200.26	**78.13**	**88.79**	**54.98**	**70.88**
		Easy	85.52	81.61	**56.91**	60.61	**198.69**	76.43	88.41	54.43	62.76

is much larger than the text that is trained.

Additionally, the UID-Var and Flesch-Kincaid give similar results, showing that these two criteria are relatively less expressive than UID-SL. The reason for UID-Var may be that it did not consider that the text difficulty increases with the sentence length, and the reason for the Flesch-Kincaid score is that simple syllables and sentence length cannot fully characterize the difficulty of the text. Though they are not very accurate compared to the UID-SL method, they could still reflect the difficulty of a text from different aspects. The UID-Var introduces the idea of the language-level average surprisal which shows that even a sentence with the same meaning may lead to different difficulties based on language-level average surprisal. The Flesch-Kincaid score considers the phonology factor and shows that pronunciation also affects the difficulty of a text.

6.2 Effect of training order

In this part, we train the model in a human-like schedule that trains from easy to hard and another schedule that trains reversely, then we report the convergence step divided by the total steps and the performance on the test set. We select RTE, SQUAD, MRPC, and SST2 to perform the experiments. The results are in Table 3.[4]

From the results, training the model in a human-like schedule leads to a quicker convergence, whereas a reverse schedule shows the slowest convergence. The random schedule is in between. This means we could train the model more efficiently in a human-like schedule.

Additionally, different training schedules give a close best performance, indicating the final performance is not sensitive to the training schedule. But the convergence is sensitive to it, which helps explain why a difficulty-insensitive schedule is successful in NLP. That is the performance which does not obviously relate to the order of training example, but the time to converge relates to it. Therefore, even the best performance is insensitive to difficulty, we could still reach comparable performance with less data in a human-like training schedule.

To show the result more clearly, we present a training curve on the SST2 task in Fig. 5. In Fig. 5, the left ax-

[4] Default criterion to split the data is UID-SL.

Table 3 Performance and convergence of different tasks. Rand means a randomly shuffled training set. P means performance. C means the convergent step divided by the total steps, which is the lower, the better.

	LSTM						BERT					
Task	E → H		H → E		Rand		E → H		H → E		Rand	
	P	C(%)	P	C(%)	P	C(%)	P	C(%)	P	C(%)	P	C(%)
SST2	81.65	**15.07**	81.67	36.29	**81.77**	18.50	**93.00**	**22.49**	92.88	44.23	92.78	28.44
MRPC	82.06	**20.04**	81.91	53.22	**82.26**	40.16	**87.73**	**19.04**	87.43	43.24	86.59	31.09
RTE	**58.18**	**16.75**	58.19	75.38	58.12	42.33	64.62	**31.22**	63.89	54.23	**65.34**	44.17
SQUAD	56.32	**39.22**	56.21	68.63	**56.43**	47.21	82.34	**30.21**	82.03	53.21	**83.21**	43.46

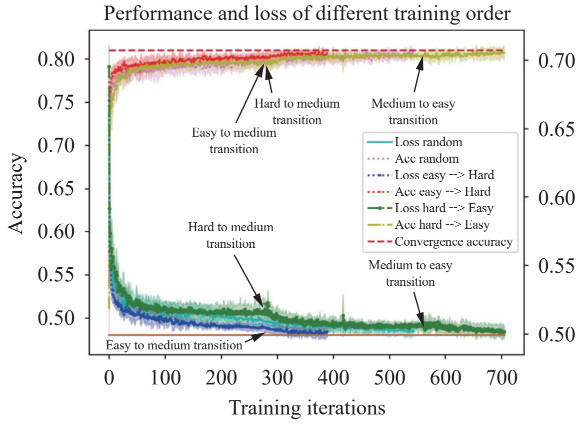


Fig. 5 LSTM learning curve in SST2 task. Transition points are annotated using arrow symbols.

is means the accuracy of this task, the right axis means the loss during training and the bottom axis is the training iterations. We also annotated the transition point that the data example shifts to another difficulty. We present the loss and accuracy curves of different training schedules using different colors. From Fig. 5, we could see a human-like training order leads to an evident quicker convergence, in which we could see both the loss and accuracy curve of easy-to-hard training stops with obviously lesser iteration steps. However, the reverse schedule leads to the slowest convergence. Based on the observation of the transition point annotated using the arrow symbol, we could tell that the LSTM uses the easy part and a subset of the medium part of train data in the human-like training schedule to achieve the best performance. In contrast, the reverse schedule needs to use whole data to achieve the same performance. Moreover, the loss also decreases slowly when we train on the hard data, which implies that the model did not effectively learn the hard data or the hard data is not the same informative as the easy data.

6.3 Transfer between text difficulty

This part investigates the difficulty transfer on the test set. For example, suppose the model trained on easy data performs better than the model trained on hard data. In that case, we ask the question: In which difficulty level of the test set, the performance improves or decreases? To be more specific, we investigate the effect of transfer between text difficulty. For humans, if the human learns in easy text, he mostly would fail in hard text, but if a human could perform well on hard text, he mostly would also succeed in easy text. We investigate whether the machine has such a learning phenomenon or not.

To answer the above question, we split the test set of SST2, WT2, MRPC, RTE, QNLI, and SQUAD into 3 difficulty levels using all criteria and then collect the model performances on each difficulty level of the test set. We sort the results and give 3, 2 and 1 scores to the best, mediocre and lowest performances. The average scores are in Tables 4 and 5.

Table 4 Transfer scores between different text difficulties. Med means medium difficulty level.

Train	LSTM		
	Evaluation set		
Set	Easy	Med	Hard
Easy	**2.54**	**2.33**	1.29
Med	2.16	2.21	2.17
Hard	1.29	1.46	**2.54**

Table 5 Transfer scores between different text difficulties. Med means medium difficulty level.

Train	BERT		
	Evaluation set		
Set	Easy	Med	Hard
Easy	**2.67**	**2.37**	1.67
Med	1.95	2.24	1.83
Hard	1.38	1.46	**2.50**

From the results, the model trained on the easy level has the best performance in both easy and mediocre levels datasets, whereas the model trained on the hard level only performs well on the hard level and fails to give the same results in other levels. Additionally, the model trained on a medium level gives a stable performance in all difficulty levels of the test set. The model behaves in a way like humans when trained on an easy or mediocre set while behaving in a totally contradicting way while trained on the hard set. Such phenomenon happens to both LSTM and BERT, which shows that this is a more general problem beyond the model itself, which means the model does not have such learning behavior as humans.

The reason may be hard text difficulty dataset follows a different distribution from the easy and medium levels. In the hard dataset, as pointed out by results in Section 6.1.2, the difficulty not only relates to the textual level but also relates to a higher concept like the difficulty of inference or understanding that text, and for such hard text, textual difficulty is only one feature and mixed with other different kinds of difficulties. For example, the easy or medium question usually relates to a very general concept that could be easily understood without much difficulty. Therefore, an easy textual difficulty can be thought of as the most important feature of the general difficulty. However, for the hard example, it might require much more knowledge than just general concepts, which drives the hard data to shift from the distribution of easy and medium data. Moreover, for a human, if the human can finish the hard task, this means that he already learned the easy one since humans cannot directly learn hard problems at the first step, but the model could directly learn the hard data which is not aligned with humans. Therefore, a model trained on the distribution of hard data cannot perform well on easy or medium-level data.

7 Conclusions

In this work, we investigate how and in what way the text difficulty affects and exists in NLP tasks and models. We analyze experiments on nine tasks using HLM Index. Results show that LSTM gives more human-like behavior. UID-SL gives the best text difficulty evaluation. Some tasks are related to text difficulty, whereas some are not and there are other kinds of difficulties also affecting the model's performance. Moreover, the transfer experiment shows that the training that begins with easy data leads to a more general and better performance than hard data. Additionally, training with a human-like schedule is more efficient than other schedules and leads to a quicker convergence.

8 Future work

Though this research discussed how textual difficulty affects model performance and how the model is aligned with humans in this aspect, the research is still limited in several ways. Even though the model explored in this research covers two paradigms of NLP, there are still many kinds of models that are not discussed like the CNN model, GPT2, etc., which represents convolution neural network and generation-based pre-trained model. Those models could give different behaviors. Additionally, this paper only discussed the textual difficulty but there are lots of factors when we talk about the difficulty given a sentence like inference difficulty, and understanding difficulty. However, those difficulties are hard to quantify and evaluate by model. Therefore, it is beyond the focus of this research. We hope we are able to calculate the difficulty of inference and understanding to give a better estimation of textual difficulty.

Acknowledgements

We would like to gratefully acknowledge the support of the National Natural Science Foundation of China (Nos. U22B2059, 62176079), National Natural Science Foundation of Heilongjiang Province, China (No. YQ 2022F005) and the Industry-University-Research Innovation Foundation of China University (No. 2021ITA05009).

Declarations of conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

[1] D. H. Hubel, T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959. DOI: 10.1113/jphysiol.1959.sp006308.

[2] S. J. Amendum, K. Conradi, E. Hiebert. Does text complexity matter in the elementary grades? A research synthesis of text difficulty and elementary students' reading fluency and comprehension. *Educational Psychology Review*, vol. 30, no. 1, pp. 121–151, 2018. DOI: 10.1007/s10648-017-9398-2.

[3] H. J. Faulkner, B. A. Levy. How text difficulty and reader skill interact to produce differential reliance on word and content overlap in reading transfer. *Journal of Experimental Child Psychology*, vol. 58, no. 1, pp. 1–24, 1994. DOI: 10.1006/jecp.1994.1023.

[4] S. A. Crossley, H. S. Yang, D. S. McNamara. What's so simple about simplified texts? A computational and psycholinguistic investigation of text comprehension and text processing. *Reading in a Foreign Language*, vol. 26, no. 1, pp. 92–113, 2014.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 6000–6010, 2017.

[6] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning representations by back-propagating errors. *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: 10.1038/323533a0.

[7] X. Wang, Y. D. Chen, W. W. Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4555–4576, 2022. DOI: 10.1109/TPAMI.2021.3069908.

[8] E. A. Platanios, O. Stretcu, G. Neubig, B. Poczos, T. Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of the Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, USA, pp. 1162–1172, 2019. DOI: 10.18653/v1/N19-1119.

[9] N. Hollenstein, M. Barrett, M. Troendle, F. Bigiolli, N. Langer, C. Zhang. Advancing NLP with cognitive language processing signals, [Online], Available: https://arxiv.org/abs/1904.02682, 2019.

[10] M. Barrett, J. Bingel, N. Hollenstein, M. Rei, A. Søgaard. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, Brussels, Belgium, pp. 302–312, 2018. DOI: 10.18653/v1/K18-1030.

[11] N. Hollenstein, C. Zhang. Entity recognition at first sight: Improving NER with eye movement information. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, USA, pp. 1–10, 2019. DOI: 10.18653/v1/N19-1001.

[12] N. Hollenstein, F. Pirovano, C. Zhang, L. Jäger, L. Beinborn. Multilingual language models predict human reading behavior. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 106–123, 2021. DOI: 10.18653/v1/2021.naacl-main.10.

[13] N. Hollenstein, A. de la Torre, N. Langer, C. Zhang. CogniVal: A framework for cognitive word embedding evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, Hong Kong, China, pp. 538–549, 2019. DOI: 10.18653/v1/K19-1050.

[14] C. Pfeiffer, N. Hollenstein, C. Zhang, N. Langer. Neural dynamics of sentiment processing during naturalistic sentence reading. *NeuroImage*, vol. 218, Article number 116934, 2020. DOI: 10.1016/j.neuroimage.2020.116934.

[15] N. Hollenstein, C. Renggli, B. Glaus, M. Barrett, M. Troendle, N. Langer, C. Zhang. Decoding EEG brain activity for multi-modal natural language processing. *Frontiers in Human Neuroscience*, vol. 15, Article number 659410, 2021. DOI: 10.3389/fnhum.2021.659410.

[16] N. Hollenstein, L. Beinborn. Relative importance in sentence processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 141–150, 2021. DOI: 10.18653/v1/2021.acl-short.19.

[17] D. Merkx, S. L. Frank. Human sentence processing: Recurrence or attention? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pp. 12–22, 2021. DOI: 10.18653/v1/2021.cmcl-1.2.

[18] J. P. Kincaid, R. P. Jr. Fishburne, R. L. Rogers, B. S. Chissom. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel, Research Branch Report 8–75, Institute for Simulation and Training, University of Central Florida, USA, 1975.

[19] S. Vajjala, D. Meurers. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*, Montreal, Canada, pp. 163–173, 2012.

[20] S. Vajjala, I. Lučić. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, USA, pp. 297–304, 2018. DOI: 10.18653/v1/W18-0535.

[21] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, USA, pp. 4171–4186, 2019. DOI: 10.18653/v1/N19-1423.

[22] A. Fenk, G. Fenk-Oczlon. Konstanz im kurzzeitgedächtniskonstanz im sprachlichen informationsfluss. *Zeitschrift für Experimentelle und Angewandte Psychologie*, vol. 27, no. 3, pp. 400–414, 1980.

[23] D. Genzel, E. Charniak. Entropy rate constancy in text. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, USA, pp. 199–206, 2002. DOI: 10.3115/1073083.1073117.

[24] M. Aylett, A. Turk. The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and Speech*, vol. 47, no. 1, pp. 31–56, 2004. DOI: 10.1177/00238309040470010201.

[25] R. Levy, T. F. Jaeger. Speakers optimize information density through syntactic reduction. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, Vancouver, Canada, pp. 849–856, 2006.

[26] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[27] C. Meister, T. Pimentel, P. Haller, L. Jäger, R. Cotterell, R. Levy. Revisiting the uniform information density hypothesis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Punta Cana, Dominican Republic, pp. 963–980, 2021. DOI: 10.18653/v1/2021.emnlp-main.74.

[28] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, pp. 353–355, 2018. DOI: 10.18653/v1/W18-5446.

[29] M. Bugert, Y. Puzikov, A. Rücklé, J. Eckle-Kohler, T. Martin, E. Martínez-Cámara, D. Sorokin, M. Peyrard, I. Gurevych. LSDSem 2017: Exploring data generation methods for the story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, Valencia, Spain, pp. 56–61, 2017. DOI: 10.18653/v1/W17-0908.

[30] S. Merity, C. M. Xiong, J. Bradbury, R. Socher. Pointer sentinel mixture models. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[31] P. Rajpurkar, R. Jia, P. Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, pp. 784–789, 2018. DOI: 10.18653/v1/P18-2124.

[32] E. F. T. K. Sang, F. De Meulder. Introduction to the CoN-

LL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, Edmonton, Canada, pp. 142–147, 2003.

[33] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

Bowen Chen received the M. Sc. degree in cyberspace security from Harbin Institute of Technology, China in 2022. He used to study at the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China. He is currently a Ph. D. degree candidate at Graduate School of Information Science and Technology, University of Tokyo, Japan.

His research interests include cognition-inspired natural language processing, temporal commonsense inference and dialogue system.

E-mail: hitbwchen@gmail.com

ORCID ID: 0000-0003-1477-2776

Xiao Ding received the Ph. D. degree in computer science from the School of Computer Science and Technology, Harbin Institute of Technology, China in 2016 where he is currently a professor.

His research interests include natural language processing, text mining, social computing, and common-sense inference.

E-mail: xding@ir.hit.edu.cn (Corresponding author)

ORCID ID: 0000-0002-5838-0320

Yi Zhao is a master student of the Harbin Institute of Technology, China. He is now studying at the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China.

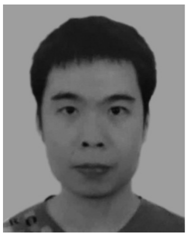
His research interest is cognition-inspired natural language processing.

E-mail: yzhao@ir.hit.edu.cn

Bo Fu received the Ph. D. degree in computer science from Harbin Institute of Technology, China in 2015. She is an NLP algorithm engineer at foundation technology center of CCB Fintech co. ltd., China.

Her research interests include pre-trained language models and dialogue systems.

E-mail: fubo.zb@ccbft.com



Tingmao Lin received the B. Sc. degree in computer science from Peking University, China in 2009. Currently, he is a machine learning engineer at foundation technology center of CCB Fintech co. ltd., China.

His research interests include stock market prediction, natural language processing and representation learning.

E-mail: lintingmao.zb@ccbft.com

Bing Qin received the Ph. D. degree in computer science from the Department of Computer Science, Harbin Institute of Technology, China in 2005. She is currently a full professor in the Department of Computer Science, and the director of Research Center for Social Computing and Information Retrieval (HIT-SCIR), Harbin Institute of Technology, China.

Her research interests include natural language processing, information extraction, document-level discourse analysis, and sentiment analysis.

E-mail: qinb@ir.hit.edu.cn

Ting Liu received the Ph. D. degree in computer science from the Department of Computer Science, Harbin Institute of Technology, China in 1998. He is currently a full professor in the Department of Computer Science, and the director of the Research Center for Social Computing and Information Retrieval (HIT-SCIR), Harbin Institute of Technology, China.

His research interests include information retrieval, natural language processing, and social media analysis.

E-mail: tliu@ir.hit.edu.cn