Pre-trained Language Model Based Active Learning for Sentence Matching

Guirong Bai[1,2], **Shizhu He**[1,2], **Kang Liu**[1,2], **Jun Zhao**[1,2], **Zaiqing Nie**[3]
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Alibaba AI Labs
{guirong.bai, shizhu.he, kliu, jzhao}@nlpr.ia.ac.cn
zaiqing.nzq@alibaba-inc.com

Abstract

Active learning is able to significantly reduce the annotation cost for data-driven techniques. However, previous active learning approaches for natural language processing mainly depend on the entropy-based uncertainty criterion, and ignore the characteristics of natural language. In this paper, we propose a pre-trained language model based active learning approach for sentence matching. Differing from previous active learning, it can provide linguistic criteria from the pre-trained language model to measure instances and help select more effective instances for annotation. Experiments demonstrate our approach can achieve greater accuracy with fewer labeled training instances.

1 Introduction

Sentence matching is a fundamental technology in natural language processing. Over the past few years, deep learning as a data-driven technique has yielded state-of-the-art results on sentence matching (Wang et al., 2017; Chen et al., 2016; Gong et al., 2017; Yang et al., 2016; Parikh et al., 2016; Gong et al., 2017; Kim et al., 2019). However, this data-driven technique typically requires large amounts of manual annotation and brings much cost. If large labeled data can't be obtained, the advantages of deep learning will significantly diminish.

To alleviate this problem, active learning is proposed to achieve better performance with fewer labeled training instances (Settles, 2009). Instead of randomly selecting instances, active learning can measure the whole candidate instances according to some criteria, and then select more efficient instances for annotation (Zhang et al., 2017; Shen et al., 2017; Erdmann et al., ; Kasai et al., 2019; Xu et al., 2018). However, previous active learning approaches in natural language processing mainly depend on the entropy-based uncertainty criterion (Settles, 2009), and ignore the characteristics of natural language. To be more specific, if we ignore the linguistic similarity, we may select redundant instances and waste many annotation resources. Thus, how to devise linguistic criteria to measure candidate instances is an important challenge.

Recently, pre-trained language models (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018; Yang et al., 2019) have been shown to be powerful for learning language representation. Accordingly, pre-trained language models may provide a reliable way to help capture language characteristics. In this paper, we devise linguistic criteria from a pre-trained language model to capture language characteristics, and then utilize these extra linguistic criteria (noise, coverage and diversity) to enhance active learning. It is shown in Figure 1. Experiments on both English and Chinese sentence matching datasets demonstrate the pre-trained language model can enhance active learning.

2 Methodology

In a general active learning scenario, there is a small set of labeled training data P and a large pool of available unlabeled data Q. Active learning is to select instances in Q according to some criteria, and then

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 28th International Conference on Computational Linguistics, pages 1495–1504
Barcelona, Spain (Online), December 8-13, 2020

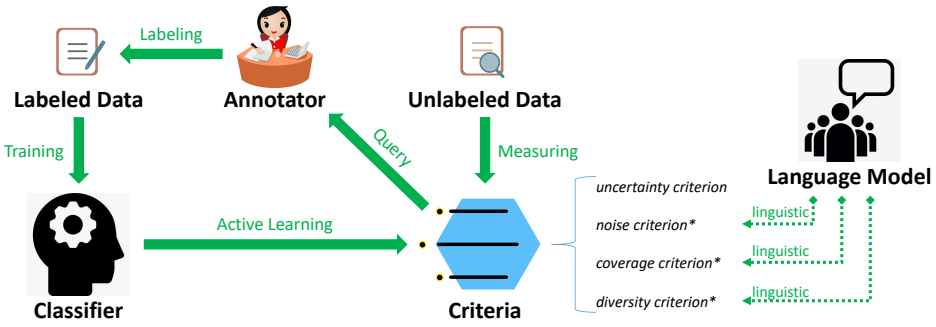

Figure 1: Pipeline of our pre-trained language model based active learning. Noise, coverage and diversity are proposed linguistic criteria from a pre-trained language model.

label them and add them into P, so as to maximize classifier M performance and minimize annotation cost. More details of preliminaries about sentence matching and active learning are in the Appendix.

2.1 Pre-trained Language Model

We choose the widely used language model BERT (Devlin et al., 2018) as the pre-trained language model. From BERT, we can obtain two kinds of information to provide linguistic criteria. One is the cross entropy loss s_{a_i} of reconstructing of the i-th word a_i in sentence A (the same with another B) by masking only a_i and predicting a_i again. The other is word embeddings (contextual representations of the last layer) **a**=[**e**(a_1),**e**(a_2),...,**e**(a_{l_A})] in the sentence, where l_A is the length of sentence A.

2.2 Criteria for Instance Selection

(1) Uncertainty: The uncertainty criterion indicates classification uncertainty of an instance and is the standard criterion in active learning. Instances with high uncertainty are more helpful to optimize the classifier and thus are worthier to be selected. The uncertainty is computed as the entropy, and we can obtain uncertainty rank $rank_{uncer}(x_i)$ for the i-th instance in Q based on the entropy. Formally,

$$rank_{uncer}(x_i) \propto -Ent(x_i) \tag{1}$$

where $Ent(x_i) = -\sum_k P(y_i = k|x_i) \log P(y_i = k|x_i)$.

(2) Noise: The noise criterion indicates how much potential noise there is in an instance. Intuitively, instances with noise may degrade the labeled data P, and we want to select noiseless instances. Noisy instances usually have rare expression with low generating probability. Thus, tokens in noisy instances may be hard to be reconstructed with context by the pre-trained language model. Based on this assumption, noise criterion is formulated about losses of reconstructing masked tokens:

$$rank_{noise}(x_i) \propto -P(A) - P(B) \tag{2}$$

where $P(A) = P(a_1 a_2 \dots a_{l_A}) \propto \frac{l_A}{\sum_{i \in l_A} s_{a_i}}$. $P(B)$ is similar. $rank_{noise}(x_i)$ denotes noise rank of the i-th instance in Q, s_{a_i}/s_{b_i} is the reconstruction loss of the i-th word a_i/b_i in sentence A/B from the pre-trained language model.

(3) Coverage: The coverage criterion indicates whether the language expression of the current instance can enrich representation learning. On the one hand, some tokens like stop words are meaningless and easy to model (high coverage). On the other hand, the classifier needs fresh instances (low coverage) to enrich representation learning. These fresh instances like relatively low-frequency professional expressions usually have lower generating probabilities than common ones. Thus, we can employ reconstruction losses to capture the low coverage ones as follows:

$$rank_{cover}(x_i) \propto -\frac{\sum_{j \in l_A} c_{a_j} s_{a_j}}{\sum_{j \in l_A} c_{a_j}} - \frac{\sum_{j \in l_B} c_{b_j} s_{b_j}}{\sum_{j \in l_B} c_{b_j}} \tag{3}$$

$$c_{a_j} = \begin{cases} 0 & if\ s_{a_j} > \beta \\ 1 & others \end{cases}, c_{b_j} = \begin{cases} 0 & if\ s_{b_j} > \beta \\ 1 & others \end{cases} \tag{4}$$

where β denotes a hyperparameter to distinguish noise and is set as 10.0.

(4) Diversity: The diversity criterion indicates the diversity of instances. Redundant instances are inefficient and waste annotation resources. In contrast, diverse ones can help learn more various language expressions and matching patterns.

First, we use a vector $\mathbf{v}_i$ for instance representation of a sentence pair instance x_i. To model the difference between two sentences, we employ the subtraction of word embeddings between "Delete Sequence" L_D and "Insert Sequence" L_I from Levenshtein Distance (when we transform sentence A to sentence B by deleting and inserting tokens, these tokens are added into L_D and L_I respectively). It is illustrated in the Appendix. Besides, the word embeddings in the subtraction are weighted by reconstruction losses. Intuitively, meaningless tokens such as preposition should have less weight, and they are usually easier to predict with lower reconstruction losses. Formally,

$$\mathbf{v}_i = \sum_{j \in L_I} w_{b_j} \mathbf{e}(b_j) - \sum_{j \in L_D} w_{a_j} \mathbf{e}(a_j) \tag{5}$$

$$w_{a_j} = \frac{s_{a_j}}{\sum_{k \in l_A} s_{a_k}}, w_{b_j} = \frac{s_{b_j}}{\sum_{k \in l_B} s_{b_k}} \tag{6}$$

where s_{a_i}/s_{b_j} is the reconstruction loss of the i/j-th word of sentence A/B. $\mathbf{e}(a_j)$/$\mathbf{e}(b_j)$ denotes word embdeddings. w_{a_i}/w_{b_j} denotes the weight for tokens.

With instance representation, we want to select diverse ones that are representative and different from each other. Specifically, we employ k-means clustering algorithm for diversity rank as follows:

$$rank_{diver}(x_i) = \begin{cases} 0 & if\ \mathbf{v}_i \circ \mathbf{v}_i \in O_{diver} \\ n & others \end{cases} \tag{7}$$

where O_{diver} are the centers of n clusters of $\{\mathbf{v}_i \circ \mathbf{v}_i\}$. $\circ$ denotes multiplication on element.

2.3 Instance Selection

In practice, according to different effectiveness of criteria, we combine ranks of criteria and select the top n candidate instances in unlabeled data Q. Specifically, we sequentially use $rank_{uncer}$, $rank_{diver}$, $rank_{cover}$, $rank_{noise}$ to select top $8n$, $4n$, $2n$, n candidate instances, and add the final n instances into labeled data P for training at every round.

3 Experiments

3.1 Settings and Comparisons

We conduct experiments on Both English and Chinese datasets, including **SNLI** (Bowman et al., 2015), **MultiNLI** (Williams et al., 2017), **Quora** (Iyer et al., 2017), **LCQMC** (Liu et al., 2018), **BQ** (Chen et al., 2018). The number of instances to select at every round is $n = 100$. We choose (Devlin et al., 2018) as classifier M and perform 25 rounds of active learning. There is a held-out test set for evaluation after all rounds. We compare the following active learning approaches:

(1)Random sampling (Random) randomly selects instances for annotation and training at each round.

(2)Uncertainty sampling (Entropy) is the standard entropy criterion (Tong and Koller, 2001; Zhu et al., 2008).

(3)Expected Gradient Length (EGL) aims to select instances expected to result in the greatest change to the gradients of tokens (Settles and Craven, 2008; Zhang et al., 2017).

(4)Pre-trained language model (LM) is our proposed active learning approach.

3.2 Results

Table 1 and Figure 2 (1-5) report accuracy and learning curves of each approach on the five datasets. Overall, our approach obtains better performance on both English and Chinese datasets. We can know that extra linguistic criteria are effective, demonstrating that a pre-trained language model can substantially capture language characteristics and provide more efficient instances for training. Besides, active learning

	SNLI	MultiNLI	Quora	LCQMC	BQ
Random	77.90	67.83	79.01	82.04	71.44
Entropy	79.80	70.27	80.21	83.25	73.60
EGL	77.86	66.80	77.91	80.35	71.59
LM	**80.99**	**71.79**	**81.79**	**84.29**	**74.73**
	Ent	E+Cov	E+Noi	E+Div	E+All
Ablation	79.80	80.99	81.11	**81.45**	80.99

Table 1: The upper part lists accuracy of different approaches on five datasets. The low part lists accuracy of combining different linguistic criterion with uncertainty on SNLI dataset for ablation.

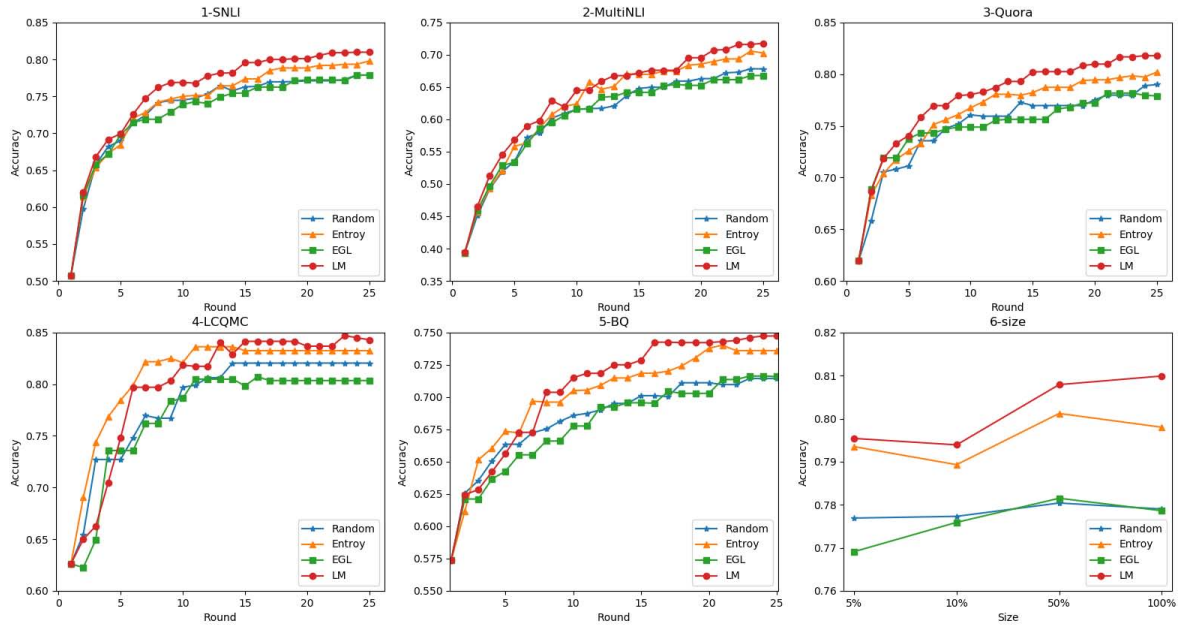

Figure 2: The figures 1-5 are learning curves of comparisons on the five datasets. The 6-th figure illustrates learning curves on four SNLI subsets to show the relation between data size and accuracy.

approaches always obtain better performance than random sampling. It demonstrates that the amount of labeled data for sentence matching can be substantially reduced by active learning. And EGL performs worse than the standard approach active learning, maybe gradient based active learning is not suitable for sentence matching. In fact, sentence matching needs to capture the difference between sentences and gradients of a single token can't reflect the relation. Moreover, we show the relation between the size of unlabeled data and accuracy in Figure 2 (6), we can see the superiority of the pre-trained model based approach is more significant for larger data size.

3.3 Ablation Study

To validate the effectiveness of extra linguistic criteria, we separately combining them with standard uncertainty criterion. "Ent" denotes the standard uncertainty criterion, "E+Noi/E+Cov/E+Div/E+All" denotes combining uncertainty with noise/coverage/diversity/all criteria. Table 1 reports the accuracy. Curves are also illustrated in the Appendix.

We can see each combined criterion performs better than a single uncertainty criterion. It demonstrates that each linguistic criterion from a pre-trained language model helps capture language characteristics and enhances selection of instances. More ablation discussions are shown in the Appendix.

4 Conclusion

In this paper, we combine active learning with a pre-trained language model. We devise extra linguistic criteria from a pre-trained language model, which can capture language characteristics and enhance active learning. Experiments show that our proposed active learning approach obtains better performance.

Acknowledgements

The work is supported by the National Natural Science Foundation of China under Grant Nos.61533018, U1936207, 61976211, and 61702512. This research work was also supported by the independent research project of National Laboratory of Pattern Recognition and the Youth Innovation Promotion Association CAS.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv*.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In *EMNLP*.

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

Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Béatrice Joyeux-Prunel, and Marie-Catherine de Marneffe. Practical, efficient, and customizable active learning for named entity recognition in the digital humanities. In *NAACL*.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv*.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. *data. quora. com*.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. In *ACL*.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *AAAI*.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *COLING*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*.

Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. *arXiv*.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *JMLR*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv*.

Yang Xu, Yu Hong, Huibin Ruan, Jianmin Yao, Min Zhang, and Guodong Zhou. 2018. Using active learning to expand training data for implicit discourse relation recognition. In *EMNLP*.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *ACM international on conference on information and knowledge management*. ACM.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv*.

Ye Zhang, Matthew Lease, and Byron C Wallace. 2017. Active discriminative text representation learning. In *AAAI*.

Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. *COLING*.

Appendix A: More Details and Discussions

Sentence Matching Task: Given a pair of sentences as input, the goal of the task is to judge the relation between them, such as whether they express the same meaning. In formal, we have two sentences A=[a_1,a_2,...,a_{l_A}] and B=[b_1,b_2,...,b_{l_B}], where a_i and b_j denote the i-th and j-th word respectively in corresponding sentences, and l_A and l_B denote the length of corresponding sentences.

Through a shared word embedding matrix $\mathbf{W}_e \in \mathbb{R}^{n_e \times d}$, we can obtain word embeddings of input sentences **a**=[**e**(a_1),**e**(a_2),...,**e**(a_{l_A})] and **b**=[**e**(b_1),**e**(b_2),...,**e**(b_{l_B})], where n_e denotes the vocabulary size, d denotes the embedding size and **e**(a_i) and **e**(b_j) denote the word embedding of the i-th and j-th word respectively in corresponding sentences. And there is a sentence matching model M to predict a label $\hat{y}$ based on **a** and **b**. When testing, we choose the label with the highest probability in prediction distribution $P(y_i|\mathbf{a}, \mathbf{b}; \theta_M)$ as output, where θ_M denotes parameters of the model M and y_i denotes a possible label. When training, the model M is optimized by minimizing cross entropy:

$$Loss = -P(y|\mathbf{a}, \mathbf{b}; \theta_M) \log P(y|\mathbf{a}, \mathbf{b}; \theta_M) \tag{8}$$

where y denotes the golden label.

Standard Active Learning: In a general active learning scenario, there exists a small set of labeled data P and a large pool of available unlabeled data Q. P is for training a classifier and can absorb new instances from Q. The task for the active learning is to select instances in Q based on some criteria, and then label them and add them into P, so as to maximize classifier performance and minimize annotation cost. In the selection criteria, a measure is used to score all candidate instances in Q, and instances maximizing this measure are selected into P.

The process is illustrated in Algorithm 1. The instance selection process is iterative, and the process will repeat until a fixed annotation budget is reached. At every round, there are n instances to be selected and labeled.

Algorithm 1 Active learning algorithm flow.

Input:

labeled data set P=$\{\varnothing\}$, unlabeled data set Q=$\{q_i\}$, the classifier M, criteria of instance selection C, the number of instances for annotation at every round n

Output:

labeled data set P=$\{p_i\}$, the classifier M

1: **repeat**
2: Sort Q based on M and C
3: Select top n instances from Q to label, update Q
4: Add labeled n instances into P, update P
5: Train and update classifier M based on P
6: **until** The annotation budget is exhausted

With the same amount of labeled data P, criteria for instance selection in active learning determine the classifier performance. Commonly, the criteria is mainly based on uncertainty criterion ($uncertainty\ sampling$), in which ones near decision boundaries have priority to be selected. A general uncertainty criterion uses entropy, which is defined as follows:

$$Ent(x_i) = -\sum_k P(y_i = k|x_i) \log P(y_i = k|x_i) \tag{9}$$

where k indexes all possible labels, x_i denotes a candidate instance that is made up of a pair of sentences A and B in available unlabeled data Q.

Visualization of Delete Sequence and Insert Sequence: To model the difference between two sentences, we employ the subtraction of word embeddings between "Delete Sequence" and "Insert Sequence" from Levenshtein Distance (when we transform sentence A to sentence B by deleting and inserting tokens, these tokens are added into "Delete Sequence" and "Insert Sequence" espectively). We illustrate it in

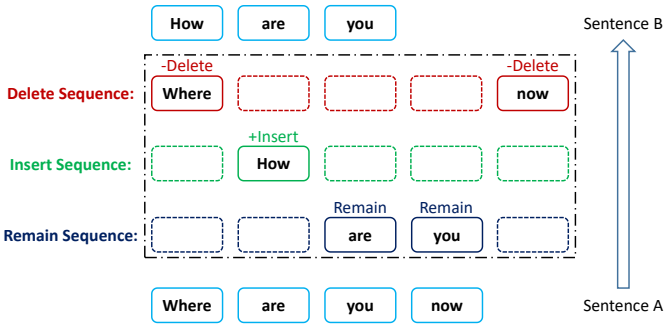

Figure 3: "Delete Sequence" and "Insert Sequence".

Figure 3.

Datasets: We conduct experiments on three English datasets and two Chinese dataset. Table 2 provides statistics of these datasets.

(1)SNLI: an English natural language inference corpus based on image captioning.

(2)MultiNLI: an English natural language inference corpus with greater linguistic difficulty and diversity.

(3)Quora: an English question matching corpus from the online question answering forum Quora.

(4)LCQMC: an open-domain Chinese question matching corpus from the community question answering website Baidu Knows.

(5)BQ: an in-domain Chinese corpus question matching corpus from online bank custom service logs.

	training	validation	test
SNLI	549,367	9,842	9,824
MultiNLI	392,702	9,815	9,832
Quora	384,348	10,000	10,000
LCQMC	238,766	8,802	12,500
BQ	100,000	1,000	1,000

Table 2: Statistics of sentence matching datasets.

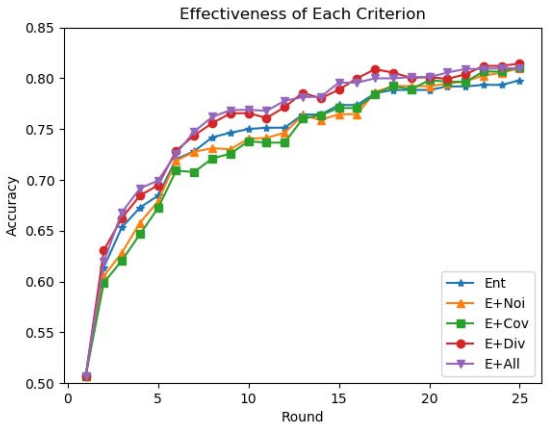

Figure 4: Learning curves of combining each proposed linguistic criterion with uncertainty on SNLI dataset.

Configuration: The number of instances to select n is 100 at every round and we perform 25 rounds of active learning, that is there are total of 2500 labeled instances for training in the end. Batch size is 16 for English and 32 for Chinese, Adam is used for optimization. We evaluated performance by calculating accuracy and learning curves on a held-out test set (classes are fairly balanced in datasets) after all rounds.

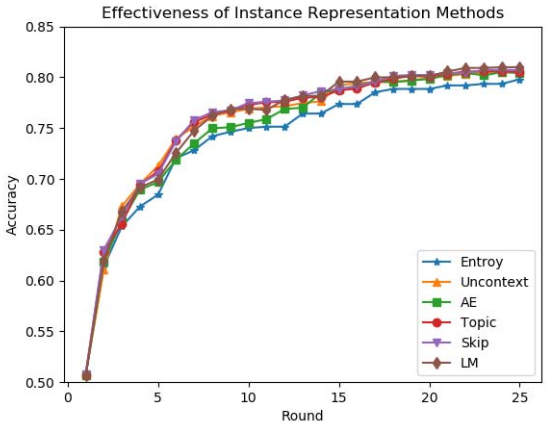

Figure 5: Learning curves of different instance representation methods.

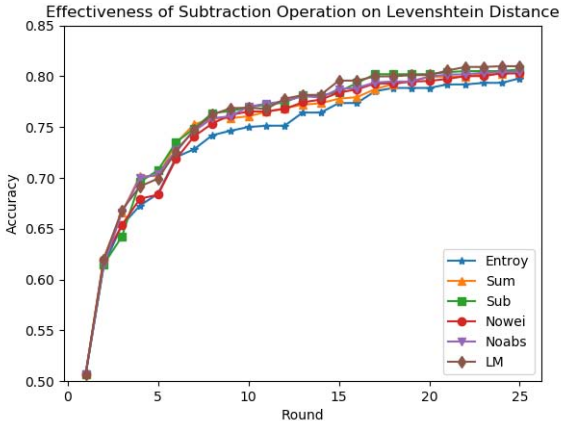

Figure 6: Learning curves of subtraction operation on Levenshtein Distance.

Curves of Ablation Study: Figure 4 shows learning curves of combining each proposed linguistic criterion with uncertainty on SNLI dataset.

Discussion:

(1)Effectiveness of different instance representation methods: We validate the effectiveness of different instance representation methods in diversity criterion on SNLI dataset. We compare our method with 4 baselines: (a) using the first word embedding layer in BERT as context-dependent representations (Uncontext); (b) using the subtraction between sentence vectors from auto-encoding (AE); (c) using the subtraction between sentence vectors from topic model (Topic); (d) using the subtraction between sentence vectors from Skip-Thoughts (Skip).

Entroy	Uncontext	AE	Topic	Skip	LM
79.80	80.63	80.42	80.54	80.71	**80.99**

Table 3: Accuracy of different instance representation methods.

Table 3 and Figure 5 report accuracy and learning curves respectively. We can see contextual representations are better than context-dependent representations. In intuition, contextual representations are more exact especially when dealing with polysemy. Next, we find our proposed method outperforms sentence vector based methods (Topic, AE, and Skip). It is possibly because BERT used more data to learn language representations.

(2)Effectiveness of subtraction operation on Levenshtein Distance: Here we validate the effectiveness of the operation that uses the subtraction of word embeddings between "Delete Sequence" and "Insert Sequence" in diversity criterion on SNLI dataset. We compare it with 4 baselines: (a) using the sum of word embeddings of the two sentences (Sum); (b) directly using the subtraction of word embeddings of

the two sentences without "Delete Sequence" and "Insert Sequence" (Sub); (c) without weight for word embeddings (Nowei); (d) without absolute value operation for symmetry (Noabs).

Entroy	Sum	Sub	Nowei	Noabs	LM
79.80	80.35	80.67	80.29	80.44	**80.99**

Table 4: Accuracy of subtraction operation on Levenshtein Distance.

Table 4 and Figure 6 report accuracy and learning curves respectively. We can see subtraction operation is better than sum operation. It demonstrates that subtraction has better ability to capture the difference between two sentences, and provides better instance representation for diversity rank. We can see the results without "Delete Sequence" and "Insert Sequence" performs a little worse, proving its necessity. And the results without weight operation for word embeddings perform worse. We can know weight for meaningless tokens is effective. Besides, we can see the results without absolute value operation for symmetry is worse, demonstrating absolute value operation is necessary.