Combining rich features and deep learning for finding similar sentences in electronic medical records

Qingyu Chen
National Institutes of Health

Jingcheng du
University of Texas Health Science Center at Houston

Sun Kim
National Institutes of Health

W. John Wilbur
U.S. Department of Health and Human Services

Combining rich features and deep learning for finding similar sentences in electronic medical records

Qingyu Chen, Jingcheng Du, Sun Kim, W. John Wilbur and Zhiyong Lu
National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health
8600 Rockville Pike, Bethesda, MD, USA
{qingyu.chen, jingcheng.du, sun.kim, john.wilbur, zhiyong.lu}@nih.gov

***Abstract*—We here describe our participation in the BioCreative/OHNLP Clinical Semantic Textual Similarity challenge task. To obtain similarity scores between clinical notes, we combine traditional machine learning and latest deep learning approaches. The machine learning model takes 63 features derived from different perspectives of textual evidence, e.g. lexical patterns, word semantics and named entities. We also use two deep learning models: sentence encoders and dense neural networks. Sentence encoders are particularly used for transfer learning. Dense neural networks are used to combine the manually chosen features and other deep learning outputs. The 5-fold cross-validation on the training set shows that our approach provides 8% better correlation coefficient than the baseline. We submitted four runs for the official test set, and the best run achieved 0.8328 correlation coefficient.**

Keywords—natural language processing; machine learning; deep learning; sentence similarity; clinical text mining

I. Introduction

The semantics between key sentences or text snippets play a vital role in text mining applications [1]. In biomedical and clinical domains, sentence semantics have direct applications, such as evidence sentence retrieval in biocuration [2, 3]. The sentence-level features also have indirect applications, such as biomedical relation extraction [4] and biomedical document summarization [5].

In the general domain, many efforts have been made to develop semantic textual similarity datasets and related models [6]. For example, the SemEval Semantic Textual Similarity (SemEval STS) challenge has been organized for over five years and the dataset collectively has close to 10,000 annotated sentence pairs. In contrast, while related work exists in biomedical and clinical domains [7], available datasets are much smaller and existing models are mostly not sufficient for specific biomedical or clinical use cases [8]. The BioCreative/OHNLP organizers have made the first attempt to annotate 1,068 sentence pairs from clinical notes and made a call for community effort to tackle the Semantic Textual Similarity (BioCreative/OHNLP STS) challenge [9].

In this paper, we report our participation in the BioCreative/OHNLP STS task. Our approaches employ random forest and deep learning models to measure sentence similarities. In our preliminary runs, we explored various features based on lexical patterns, semantics and named entities, and chose 63 features for input. Random forest and dense neural networks are used for learning the best combination of the 63 features. In addition, we also utilized sentence encoders via transfer learning, i.e. the sentence encoders trained on general domain text were tuned for the BioCreative/OHNLP dataset.

We optimized our approaches based on 5-fold cross-validation on the training set, and the results showed that our proposed methods were 8% higher than the baseline method (Q-gram string similarity) on average. We submitted four runs for the official test set, and obtained a best correlation of 0.8328.

II. Data Collection

The BioCreative/OHNLP STS dataset consists of 1,068 pairs of sentences derived from clinical notes. 750 pairs are used for the training set; 318 pairs are used for the test set. Each pair in the set was annotated by two medical experts on a scale of 0–5, from completely dissimilar to semantically equivalent. The detailed collection and curation procedure can be found in the task overview paper [9].

III. Machine learning models

Machine learning has been used for decades in diverse biomedical and clinical applications, such as biomedical information retrieval [10], natural language processing [11], and data curation [12]. According to the overview on the most recent SemEval STS task, such traditional approaches still form the essence of top performing systems [6]. Therefore, we designed a machine learning model with 63 features, described as follows.

A. Data Preprocessing

We examined the sentence pairs in the training set and developed three preprocessing steps. First, the *pre-tokenization* step separates words joined by punctuations including “/” (e.g., “PROVENTIL/VENTOLIN”), “=” (e.g., “0=”), “-” (e.g., “63-76”) and “.” (e.g., “content.Caller”). We do not split “-” and “.” if surrounding words are numbers, e.g., “twenty-four” and “100.4”. Second, the *tokenization step* converts a sentence to a list of tokens. We use the TreeBank tokenizer implemented in the NLTK toolkit [13]. The final *post-tokenization* step removes the tokens if they are punctuations, stopwords (from https://www.ncbi.nlm.nih.gov/IRET/DATASET) or single letters, and corrects spelling mistakes by checking whether a token is in word embeddings (we will detail the embeddings later), e.g., “refil” is changed to “refill”. We also transform digits to text, e.g., “24” to “twenty-four”.

The features are similarity measures generated based on the preprocessed text, and they can be categorized into four broad categories: string similarity metrics, entity similarity metrics, number similarity and similarity scores from other models.

B. String Similarity Features (60 features)

String similarity metrics measure the similarity between a pair of text snippets, which are used as features for machine learning and deep learning models. Zobel and Moffat [14] analyzed a range of similarity measures in information retrieval and found that there was no one-size-fits-all metric – no metric consistently worked better than others. We hypothesized that aggregating similarity metrics from different perspectives could better capture the similarity between sentences. In this task, we adopted metrics of four types.

Token-based measures (12 features). These measures see strings as an unordered list of tokens and evaluate the similarity between two lists of tokens. We used 10 token based similarity measures with their variations: the Jaccard similarity [15] and its generalized version (which considers two tokens are the same if the Jaro similarity is above 0.7), the Block similarity [16], the Q-gram similarity (q = 2, 3 and 4) [17], the Cosine similarity [18], the Dice similarity [19], the Overlap coefficient similarity [20], the Tversky index similarity [21], the Monge Elkan similarity [22] and the tf-idf similarity.

Sequence-based features (7 features). One limitation of token-based measures is that they ignore the order of tokens. Sequence-based metrics focus on the edit distance (i.e. insertions, deletions and substitutions) between two strings. They have been found effective in textual entailment [23], plagiarism detection [24] and duplicate records detection [25]. We used 7 sequence based metrics: Affine gap penalty score [26], Bag similarity [27], Editex similarity [28], Jaro similarity [29], Levenshtein similarity [30], Needleman Wunsch similarity [31] and Smith Waterman similarity [32].

Embedding-based features (11 features). Token-based and sequence-based metrics measure similarity based on exact match between words. However, in natural language, distinct words may have similar meanings. Word embeddings can address such cases to complement other metrics. For this task, we trained word embeddings using word2vec [33] on PubMed abstracts. PubMed abstracts were tokenized using NLTK and the parameters of word2vec were set up following Kim et al. [34]. Averaged or maxed embeddings are often used to compute sentence similarity (e.g., for a pair of sentences, transforming a sentence into a vector by averaging or maxing word vectors for all the words in that sentence and computing the Cosine similarity between the two vectors) [6]. In our approach, only words that are nouns, verbs or adjectives identified by the NLTK pos-tagger were averaged or maxed. This is because each word in a sentence does not contribute to the semantics equally; nouns, verbs and adjectives are arguably more important than others. We used the Cosine similarity [18], the Euclidean similarity [35] and its variation Squared Euclidean similarity, the Block similarity [16], the Correlation similarity [36] for a pair of average or max embeddings. We also used the Word Mover's Distance (WMD) to measure the similarity between entire sentences [37].

Domain-based features (30 features). Sentences or longer text snippets from different domains are described in a different manner. By manually examining the sentence pairs, we found that some sentence pairs shared a same substring in the beginning. For example, "Patient is here for the following immunization(s): Inactivated Influenza Virus Vaccination; Tetanus and Diphtheria and Acellular Pertussis (Tdap) Vaccine" and "Patient is here for the following immunization(s): Meningococcal conjugate Vaccine; Tetanus Toxoid, Reduced Diphtheria and Acellular Pertussis (Tdap) Vaccine." have the same substring, "Patient is here for the following immunization(s):". Such long identical prefix could bias our string similarity measures. Hence, we applied the measures again on the pairs where same prefixes were removed.

C. Entity Similarity Features (1 feature)

The clinical concepts embedded in text may provide important clues for computing similarity. We leveraged CLAMP (Clinical Language Annotation, Modeling, and Processing Toolkit) [38] to perform Named Entity Recognition (NER) and extract clinical concepts (e.g. medication, treatment, problem) from sentence pairs. The extracted concepts were then mapped to UMLS Concept Unique Identifiers (CUI). To measure the overlap of clinical concepts of sentence pair, we define the following overlap score:

$$\frac{len(concepts_{sent1} \cap \quad concepts_{sent2} \ )}{MAX\ (len(concepts_{sent1}), \quad len(concepts_{sent2}))}$$

In biomedical and clinical domains, the same entities or concepts often have different expressions, e.g., "vaccination" vs "immunization", "cancer" vs "tumor". In order to capture such synonymous concepts, we first map the tokens in the concepts to the embeddings, and then WMD was computed between concepts. If the WMD was equal or less than a threshold (we set it as 0.4 empirically), we considered the entity pair as a match.

D. Number Similarity Features (1 feature)

We also observed clinical sentences often contain numbers and these numbers play an important role for sentence similarity scores. Our preprocessing step normalizes numbers to words. Thus, the number similarity is the similarity between two normalized texts. We compute WMD between normalized numbers if both sentences in a pair include numbers. If neither has a number, the similarity score is 1; if only one of them has a number, the score is 0.

E. Deep Learning Features (1 feature)

In addition to the features above, we also used the output score of the Encoder-MLP with the *universal sentence encoder* (explained below) as a feature.

We used these features and tested a variety of models implemented in the Scikit-learn toolkit [39]. Random Forest gave the best results and was used for our submitted runs.

IV. Deep learning models

In contrast to traditional machine learning approaches with feature engineering, deep learning models aim to extract features and learn representations automatically, requiring minimal human effort. To date, deep learning has produced the state-of-the-art performance in many biomedical and clinical applications [40], such as medical image classification [41], mental health text mining [42], and disease progression prediction [43]. For the STS challenge, we developed two deep learning models, sentence encoders and dense neural networks (DNN), to complement our traditional machine learning models described earlier:

A. Sentence Encoders

A sentence encoder transforms the text into a high dimensional vector space, which can be further used to capture the semantic similarity. We converted the regression problem (i.e., the similarity score is continuous) to a classification problem (i.e., the similarity score is categorical, from 0, 1 to 5) and designed an encoder based multiple layer perceptron (Encoder-MLP) for the similarity classification task.

Encoder-MLP first transforms the two sentences into two high dimensional vectors, respectively. Then, the element-wise absolute difference and the element-wise multiplication of two vectors are concatenated together as the representation of the sentence pair. Three fully connected dense layers are added (number of nodes: 512, 256, 64). The output layer is a softmax layer with 6 units. Cross-entropy was set as the loss function. We also added dropout to avoid overfitting.

Considering the limited size of the clinical training corpus, we leveraged transfer learning to improve the performance. We first trained the Encoder-MLP on the STS dataset and then fine-tuned the model on the training set. We chose Adam optimizer [44] and set the learning rate at 0.001. We evaluated two pre-trained sentence encoders, including *universal sentence encoder* [45] and *inferSent* [46].

B. Dense Neural Networks

We also implemented a DNN that combines human engineered features and deep learning features. It used the traditional machine learning model features mentioned above and a 1024-dimensional vector from the universal sentence encoder (a 512-dimensional vector for each sentence). The DNN is a three-layer MLP, where the number of hidden units is 144, 32 and 1 respectively. We followed the popular weight initialization approach proposed by He et al. [47], applied the L2 regularization, added dropout, and used the ReLu activation function [48]. For training, we selected Adam optimizer at a learning rate of 0.0001, set the cost function to mean squared error function, and adopted the early stop to reduce overfitting.

V. Results and discussion

A. Feature Analysis

We split the training set into 5 folds; each contains 450 for training, 150 for validation and 150 for testing. We trained the model on the 450-pair training set, performed parameter tuning on the 150-pair validation set and tested on the 150-pair test set. Table I shows the feature analysis results for the Random Forest model. We compared individual string similarity metrics (token and sequence based) and found Q-gram (q = 3) gives the highest correlation, thus making it as a baseline. Leveraging multiple token-based metrics gave a 3.5% increase; adding sequence-based metrics increased further by 1%. Embedding-based metrics also have positive impacts: improving the correlation by 2.1%. Number-based feature, entity feature and other model scores are single features. These three features collectively provided a further 1% improvement. The Random Forest model, after feature selection (recursive feature elimination) and parameter optimization (number of trees and tree depth), has an 8% higher correlation and a 4% lower variance than the baseline.

TABLE I. Feature analysis results on 5-fold cross-validation. Average correlation, standard deviation and the percentage improved are reported.

Features	**Average correlation ± standard deviation (improved percentage)**
Baseline (Q-gram)	0.7850 ± 0.063
Token-based	0.8199 ± 0.042 (+3.5%)
+ Sequence-based	0.8288 ± 0.041 (+0.9%)
+ Embedding-based	0.8498 ± 0.029 (+2.1%)
+ Number-based	0.8503 ± 0.030 (+0.1%)
+ Entity features	0.8539 ± 0.029 (+0.4%)
+ Deep learning features	0.8552 ± 0.029 (+0.1%)
+ Optimization	0.8569 ± 0.029 (+0.2%)

TABLE II. Deep learning model results on 5-fold cross-validation. Universal: the universal sentence encoder; InferSent: the inferSent sentence encoder; No transfer: Directly training on the task training set. ML: machine learning features.

Model	**Average correlation ± std**
Encoder-MLP (*universal*)	0.6847 ± 0.097
Encoder-MLP (*inferSent*)	0.7642 ± 0.073
Encoder-MLP (*universal*, no transfer)	0.6367 ± 0.126
Encoder-MLP (*inferSent*, no transfer)	0.7121 ± 0.077
DNN (ML + *universal*)	0.8562 ± 0.027

B. Deep Learning Model Results

The results of deep learning models are summarized in Table II. The DNN model had similar performance to the

Random Forest model; the performance of Encoder-MLPs (with transfer learning) ranged from 0.68 to 0.76. The *inferSent* encoder gave the highest correlation of 0.76. It shows that the end-to-end deep learning models still have room to improve. The performance may increase further with continuing development of the training set. It also demonstrates transfer learning is the best approach in clinical domain given the scale of the current datasets.

TABLE III. SUBMISSION TRAINING SET AND OFFICIAL TEST RESULTS. THE MEDIAN AND AVERAGE RESULTS OF ALL THE TEAM SUBMISSIONS ARE REPORTED.

Submission	Training set correlation	Test set correlation
Random Forest	0.8569	0.8106
Random Forest + DNN	0.8648	0.8246
Regression	**0.8662**	**0.8328**
Random Forest + Encoder-MLP (*inferSent*)	0.8337	0.8258
All team median	-	0.8016
All team average	-	0.7820

C. Submissions and Test Set Results

We submitted four runs for the test set. The first run was the Random Forest model (the number of trees was 1,500 and the max tree depth was 6). The predictions were the average scores of the 5-fold cross-validation. (using training and validation set for training the model). The second submission was the average score between the Random Forest model and the DNN using the same 5-fold cross-validation.

For the third submission, we applied stacking techniques. We trained 8 models on the training set only: the Random Forest model, the Bayesian Ridge regression model, the Lasso regression model, the linear regression model, the Extra Tree model, the DNN using the Universal Sentence Encoder, the DNN using the *inferSent* encoder, and the Encoder-MLP using the *inferSent* encoder. We then used the prediction scores of the models on the validation set to train a Ridge regression model that accumulates these 8 model scores. We set a large L2 regularization penalty (alpha=10) to ensure the regression model does not rely on a single model score. The prediction scores of the third submission were the output of the accumulated regression model.

The fourth submission was similar to the second: the average score between the Random Forest and the Encoder-MLP using the *inferSent* encoder.

Table III presents the official results for our submissions. The Random Forest model achieved 0.8106 correlation. The second and the fourth submissions achieved a similar correlation of ~0.825, showing that deep learning models can complement traditional machine learning models for sentence similarity. The third model achieved a correlation of 0.8328. While the fourth submission had the lowest performance in the training set, its performance on the testing set only decreased slightly and had the second highest correlation. This suggests that end-to-end deep learning may have potential to make a more generalizable model. Overall, the result of our best submission was more than 2% and 5% higher than the median and average result amongst all the submissions respectively.

VI. CONCLUSIONS

In this paper, we report our efforts on the BioCreative/OHNLP STS task. The proposed approaches utilize traditional machine learning, deep learning and the ensemble between them. Our best system shows 0.8662 and 0.8328 coefficient correlation scores for the training and test sets, respectively. Future work includes the development of deep learning models using Convolutional Neural Networks and Long Short-Term Memory Networks. These two architectures have shown promise in semantic similarity related applications. We also plan to incorporate negation-based features into the current models. It is expected that they will bring new perspectives to complement our current models.

ACKNOWLEDGMENT

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine, and UTHealth Innovation for Cancer Prevention Research Training Program Pre-doctoral Fellowship (Cancer Prevention and Research Institute of Texas grant # RP160015). We also thank Yifan Peng, Aili Shen and Yuan Li for various discussions. In addition, we thank Yaoyun Zhang and Hua Xu for word embedding related input.

REFERENCES

[1] Aggarwal CC, Zhai C. Mining text data. Springer Science & Business Media, 2012.
[2] Rastegar-Mojarad M, Komandur Elayavilli R, Liu H. BELTracker: evidence sentence retrieval for BEL statements. Database 2016;2016.
[3] Mao Y, Van Auken K, Li D, Arighi CN, McQuilton P, Hayman GT, et al. Overview of the gene ontology task at BioCreative IV. Database 2014;2014.
[4] Chen Q, Panyam NC, Elangovan A, Davis M, Verspoor K. Document Triage and Relation Extraction for Protein-Protein Interactions affected by Mutations. Proceedings of the BioCreative VI Workshop 2017;6:52.1.
[5] Chandu K, Naik A, Chandrasekar A, Yang Z, Gupta N, Nyberg E. Tackling Biomedical Text Summarization: OAQA at BioASQ 5B. BioNLP 2017 2017:58-66.
[6] Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L. SemEval-2017 Task 1: Semantic Textual Similarity-Multilingual and Cross-lingual Focused Evaluation. arXiv preprint arXiv:1708.00055 2017.
[7] Soğancıoğlu G, Öztürk H, Özgür A. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics 2017;33:i49-i58.
[8] Chen Q, Kim S, Wilbur J, Lu Z. Sentence similarity measures revisited: ranking sentences in PubMed documents. To appear in ACMBCB'18: 9th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics 2018.
[9] Wang Y, Afzal N, Liu S, Rastegar-Mojarad M, Wang L, Shen F, et al. Overview of the BioCreative/OHNLP Challenge 2018 Task 2: Clinical Semantic Textual Similarity. . To appear in Proceedings of the BioCreative/OHNLP Challenge. 2018. 2018.
[10] Fiorini N CK, Starchenko G, Kireev E, Kim W, Miller V, Osipov M, Kholodov M, Ismagilov R, Mohan S, Ostell J, Lu Z. Best Match: new relevance search for PubMed. PLoS Biology 2018.
[11] Wei C-H, Phan L, Feltz J, Maiti R, Hefferon T, Lu Z. tmVar 2.0: integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine. Bioinformatics 2017;34:80-7.
[12] Chen Q, Zobel J, Zhang X, Verspoor K. Supervised Learning for Detection of Duplicates in Genomic Sequence Databases. PLoS One 2016;11:e0159644.
[13] Bird S, Klein E, Loper E. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.", 2009.
[14] Zobel J, Moffat A. Exploring the similarity space. ACM SIGIR Forum 1998;32:18-34.
[15] Jaccard P. The distribution of the flora in the alpine zone. 1. New phytologist 1912;11:37-50.
[16] Krause EF. Taxicab geometry: An adventure in non-Euclidean geometry. Courier Corporation, 1986.
[17] Ukkonen E. Approximate string-matching with q-grams and maximal matches. Theoretical computer science 1992;92:191-211.
[18] Singhal A. Modern information retrieval: A brief overview. IEEE Data Eng. Bull. 2001;24:35-43.
[19] Dice LR. Measures of the amount of ecologic association between species. Ecology 1945;26:297-302.
[20] Pianka ER. The structure of lizard communities. Annual review of ecology and systematics 1973;4:53-74.
[21] Tversky A. Features of similarity. Psychological review 1977;84:327.
[22] Monge AE, Elkan C. The Field Matching Problem: Algorithms and Applications. KDD 1996:267-70.
[23] Kouylekov M, Magnini B. Recognizing textual entailment with tree edit distance algorithms. Proceedings of the First Challenge Workshop Recognising Textual Entailment 2005:17-20.
[24] Su Z, Ahn B-R, Eom K-Y, Kang M-K, Kim J-P, Kim M-K. Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm. Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on 2008:569-.
[25] Chen Q, Zobel J, Verspoor K. Duplicates, redundancies and inconsistencies in the primary nucleotide databases: a descriptive study. Database (Oxford) 2017;2017.
[26] Myers EW, Miller W. Optimal alignments in linear space. Bioinformatics 1988;4:11-7.
[27] Bartolini I, Ciaccia P, Patella M. String matching with metric trees using an approximate distance. International Symposium on String Processing and Information Retrieval 2002:271-83.
[28] Zobel J, Dart P. Phonetic string matching: Lessons from information retrieval. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval 1996:166-72.
[29] Jaro MA. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association 1989;84:414-20.
[30] Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady 1966;10:707-10.
[31] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology 1970;48:443-53.
[32] Smith TF, Waterman MS. Comparison of biosequences. Advances in applied mathematics 1981;2:482-9.
[33] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 2013:3111-9.
[34] Kim S, Fiorini N, Wilbur WJ, Lu Z. Bridging the gap: incorporating a semantic similarity measure for effectively mapping PubMed queries to documents. Journal of biomedical informatics 2017;75:122-7.
[35] Danielsson P-E. Euclidean distance mapping. Computer Graphics and image processing 1980;14:227-48.

[36] Allemang RJ, Brown DL. A correlation coefficient for modal vector analysis. Proceedings of the 1st international modal analysis conference 1982;1:110-6.
[37] Kusner M, Sun Y, Kolkin N, Weinberger K. From word embeddings to document distances. International Conference on Machine Learning 2015:957-66.
[38] Soysal E, Wang J, Jiang M, Wu Y, Pakhomov S, Liu H, et al. CLAMP–a toolkit for efficiently building customized clinical natural language processing pipelines. Journal of the American Medical Informatics Association 2017;25:331-6.
[39] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of machine learning research 2011;12:2825-30.
[40] Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, et al. Opportunities and obstacles for deep learning in biology and medicine. Journal of The Royal Society Interface 2018;15:20170387.
[41] Wang X, Peng Y, Lu L, Lu Z, Summers RM. Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018:9049-58.
[42] Du J, Zhang Y, Luo J, Jia Y, Wei Q, Tao C, et al. Extracting psychiatric stressors for suicide from social media using deep learning. BMC medical informatics and decision making 2018;18:43.
[43] Razavian N, Marcus J, Sontag D. Multi-task prediction of disease onsets from longitudinal laboratory tests. Machine Learning for Healthcare Conference 2016:73-100.
[44] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 2014.
[45] Cer D, Yang Y, Kong S-y, Hua N, Limtiaco N, John RS, et al. Universal sentence encoder. arXiv preprint arXiv:1803.11175 2018.
[46] Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A. Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364 2017.
[47] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision 2015:1026-34.
[48] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th international conference on machine learning (ICML-10) 2010:807-14.