A Dynamic Word Representation Model Based on Deep Context

四川大學 SICHUAN UNIVERSITY

Xiao Yuan, Xi Xiong*, Shenggen Ju, Zhengwen Xie and Jiawei Wang
flyxiongxi@gmail.com

望江人工智库（K-TF-P）
https://yuanxiaosc.github.io/

Introduction

The currently used word embedding techniques use fixed vectors to represent words without the concept of context and dynamics. This paper proposes a deep neural network CoDyWor to model the context of words so that words in different contexts have different vector representations of words. CoDyWor is a deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy).

Aim

One-hot encoding

	cat	mat	on	sat	the
the =>	0	0	0	0	1
cat =>	1	0	0	0	0
sat =>	0	0	0	1	0
…	…				

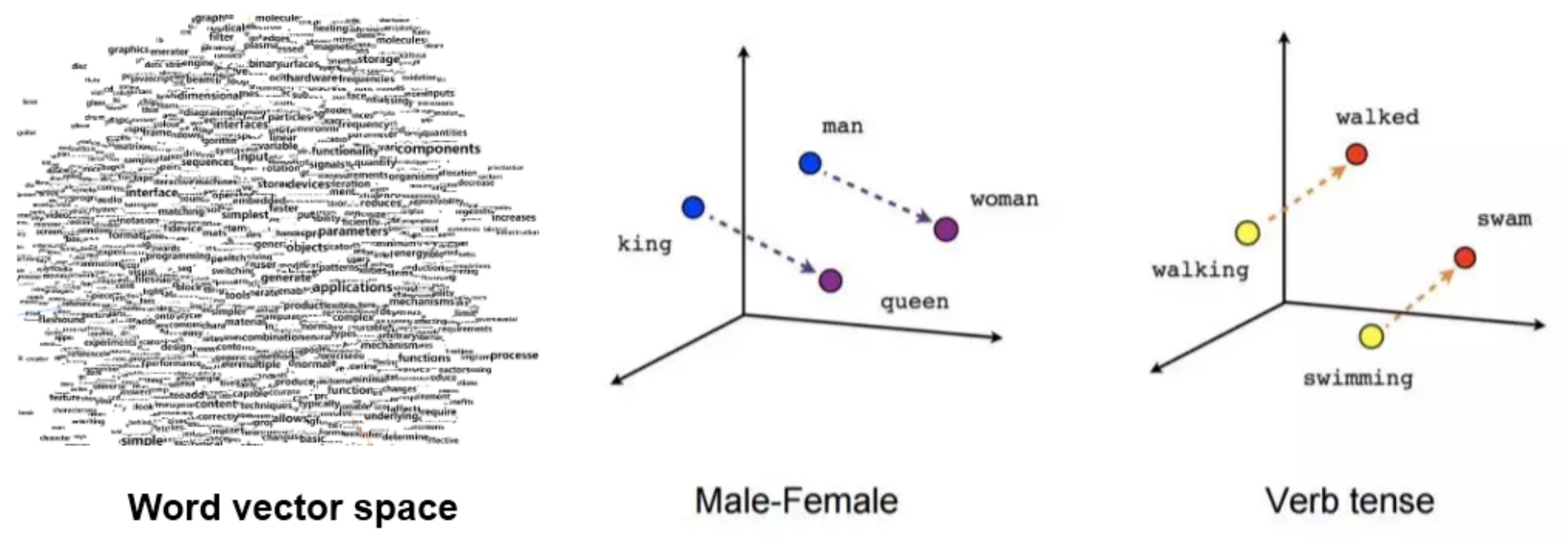
A 4-dimensional embedding

cat =>	1.2	-0.1	4.3	3.2
mat =>	0.4	2.5	-0.9	0.5
on =>	2.1	0.3	0.1	0.4
…	…			

The NLP system available based on deep learning usually first converts the text input into a vectorized word representation, i.e. the word embedding vector, and then proceeds to be processed next. We need vectorized representations of words with more a priori information. These a priori information includes: word meaning, syntax, common sense, and so on.

Technological Evolution

2013 word2vec -> 2014 Glove -> 2015 Skip-thoughts -> 2016 fastText

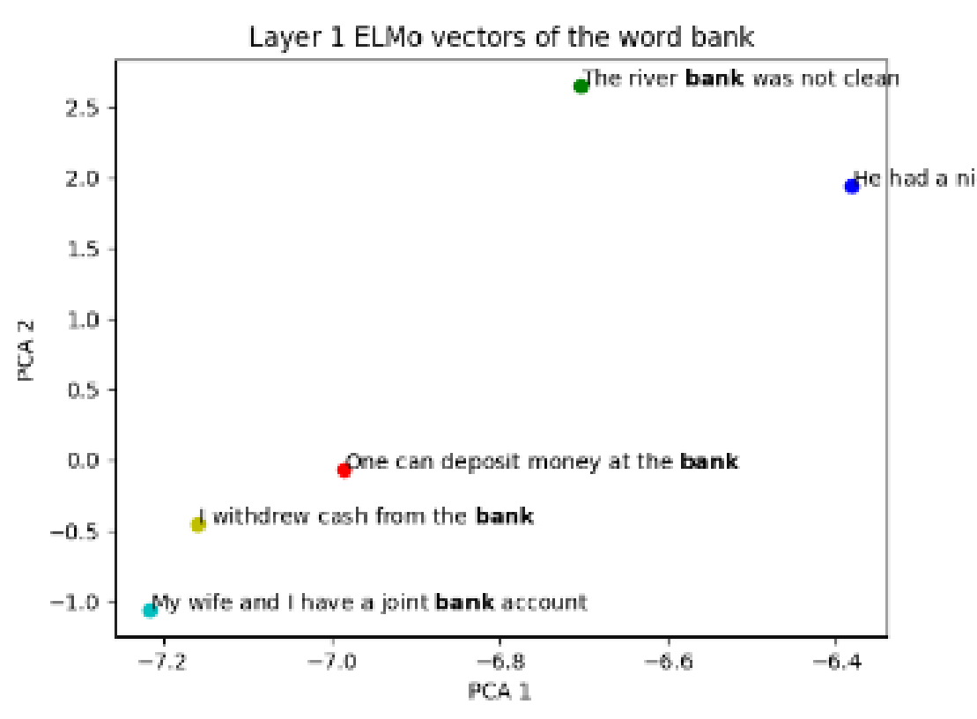


Word vector space　Male-Female　Verb tense

The word embedding technique described above can convert words into vectors, and can ensure that vectors corresponding to similar words are closer in vector space.

Please note: A word corresponds to a fixed vector.

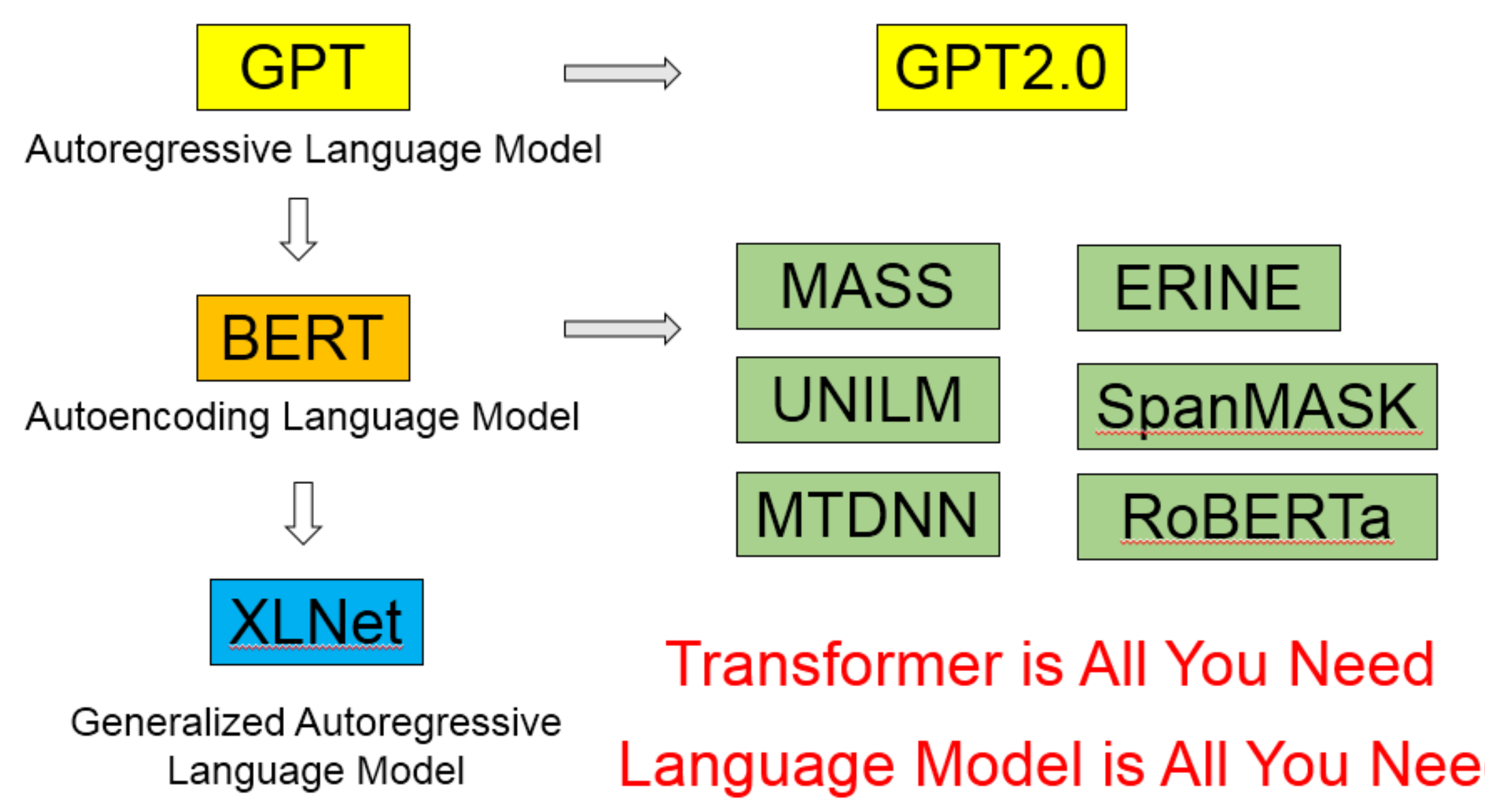
2018 ELMo



We can cluster the above contextual vectors into 2 groups. The word vectors of bank in the upper right cluster mean a slope land besides water, while the bottom left cluster has the meaning of a financial institution.

A word can correspond to a different word vector depending on the context.

2018 GPT -> BERT -> 2019 BERT improved version



GPT ⟹ GPT2.0

Autoregressive Language Model

⇩

BERT ⟹ MASS, ERINE, UNILM, SpanMASK, MTDNN, RoBERTa

Autoencoding Language Model

⇩

XLNet

Generalized Autoregressive Language Model

Transformer is All You Need

Language Model is All You Need

Method

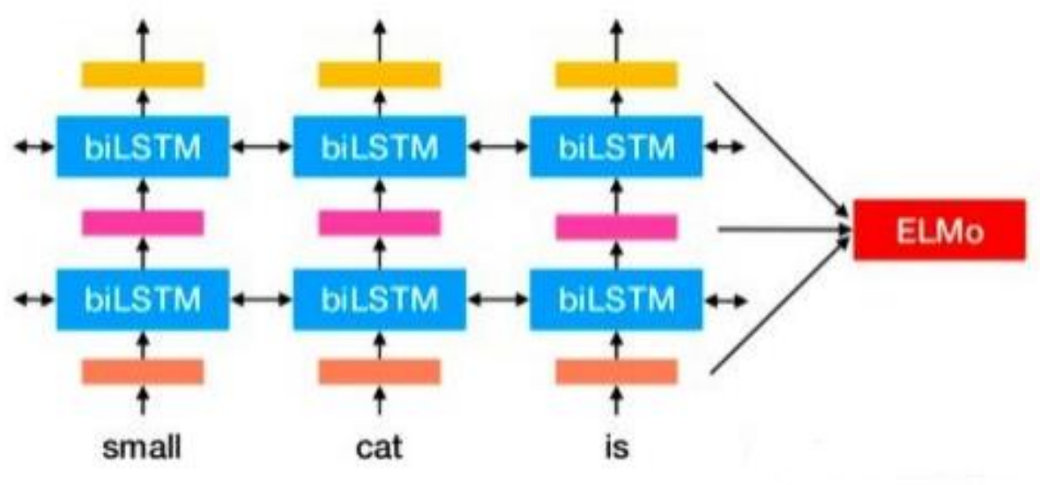
The method in this paper adopts the depth Neural Network Language Model (NNLM), meanwhile, it absorbs the idea of generating the word vector from the internal state of NNLM in ELMo, and replaces the BiLSTM encoder in the model with the Transformer encoder with concurrent computing and contextual coding capability, and introduces a multi-layer attention mechanism, blending word representation information of different levels in neural network, and generating the word vector with contextual meanings.

Approach = ELMo Framework + Transformer Feature Extractor

ELMo Framework

$$\mathbf{ELMo}_k^{task} = E(R_k;\Theta^{task}) = \gamma^{task}\sum_{j=0}^{L} s_j^{task}\mathbf{h}_{k,j}^{LM}$$

$\mathbf{s}^{task}$ are softmax-normalized weights and the scalar parameter γ^{task} allows the task model to scale the entire ELMo vector. At each position k, LSTM layer outputs a context-dependent representation $\mathbf{h}_{k,j}^{LM}$ where $j = 1,\ldots,L$.



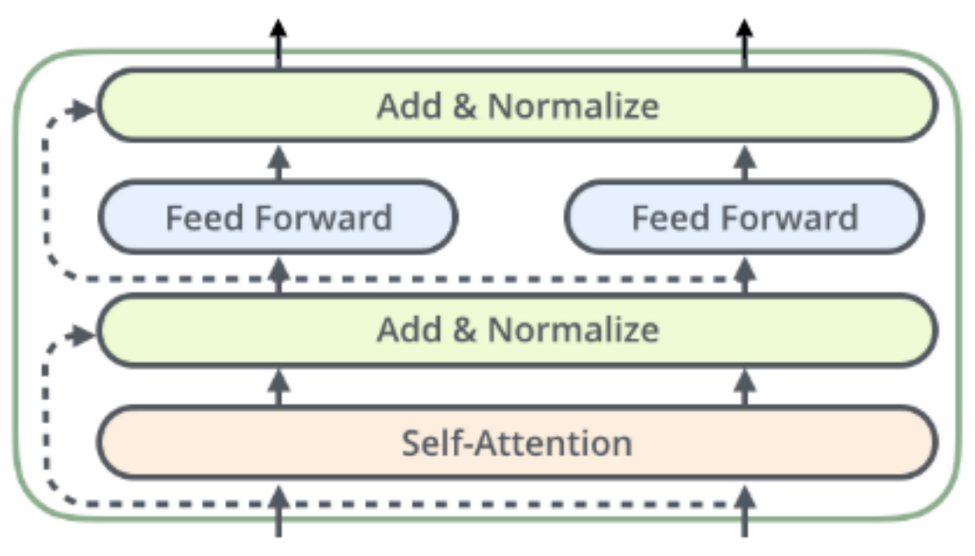
Transformer Feature Extractor

Scaled Dot-Product Attention

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Multi-Head Attention

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1},...,\mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

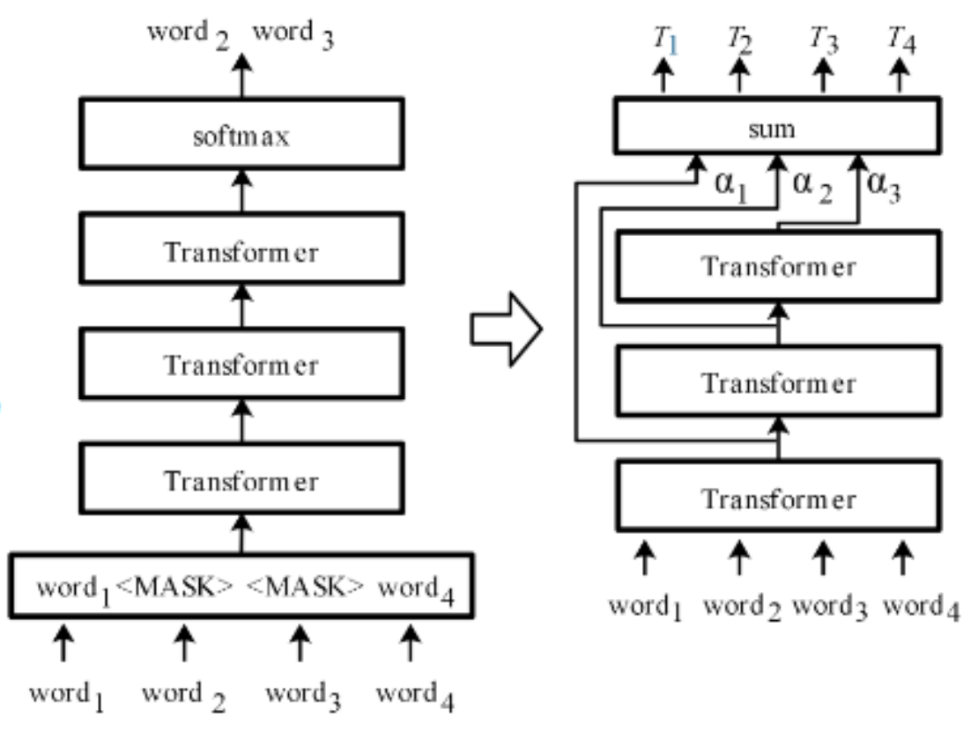


Transformer encoder layer

Proposed Approach

$$h_i = Transformer(h_{h-1}), \forall i \in [1, L]$$

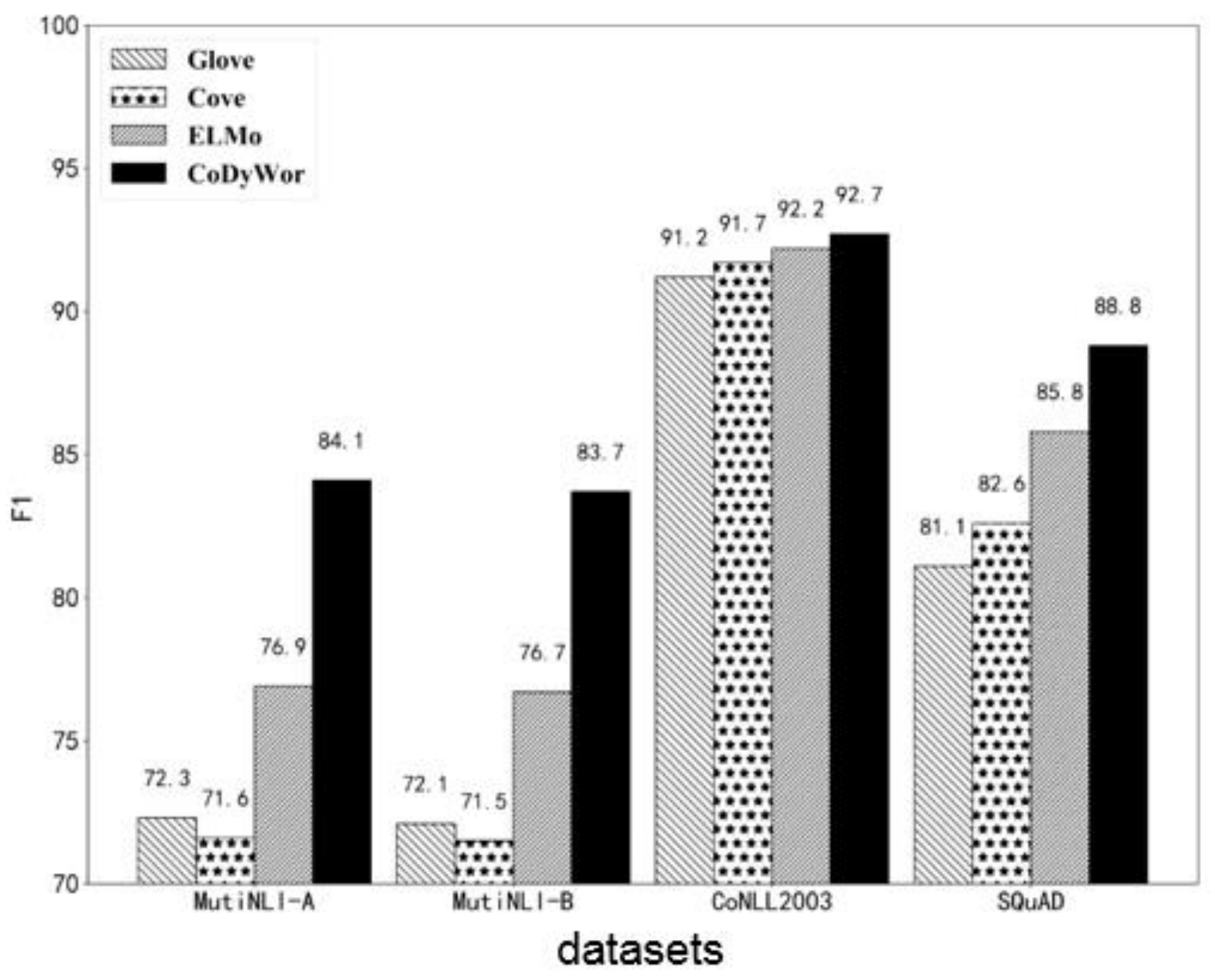
$$CoDyWor(word) = \beta\sum_{j=1}^{L}\alpha_j h_j, where \begin{cases} \alpha_1+\alpha_2+\ldots+\alpha_L=1 \\ \alpha_j \geq 0 \end{cases}, \beta \neq 0$$



The overall framework of the dynamic context representation model of deep context proposed in this paper is illustrated in above Fig. It consists of two main processes: 1) the masked language model on the left of Fig.; 2) The Transformer layer in the pre-trained masked language model is extracted and a new output layer is added to form the model of this paper—the deep context dynamic word representation model (on the right side of Fig.), which blends multiple outputs of Transformer layer through a multi-layered attention mechanism, generating a deep dynamic word representation vector.

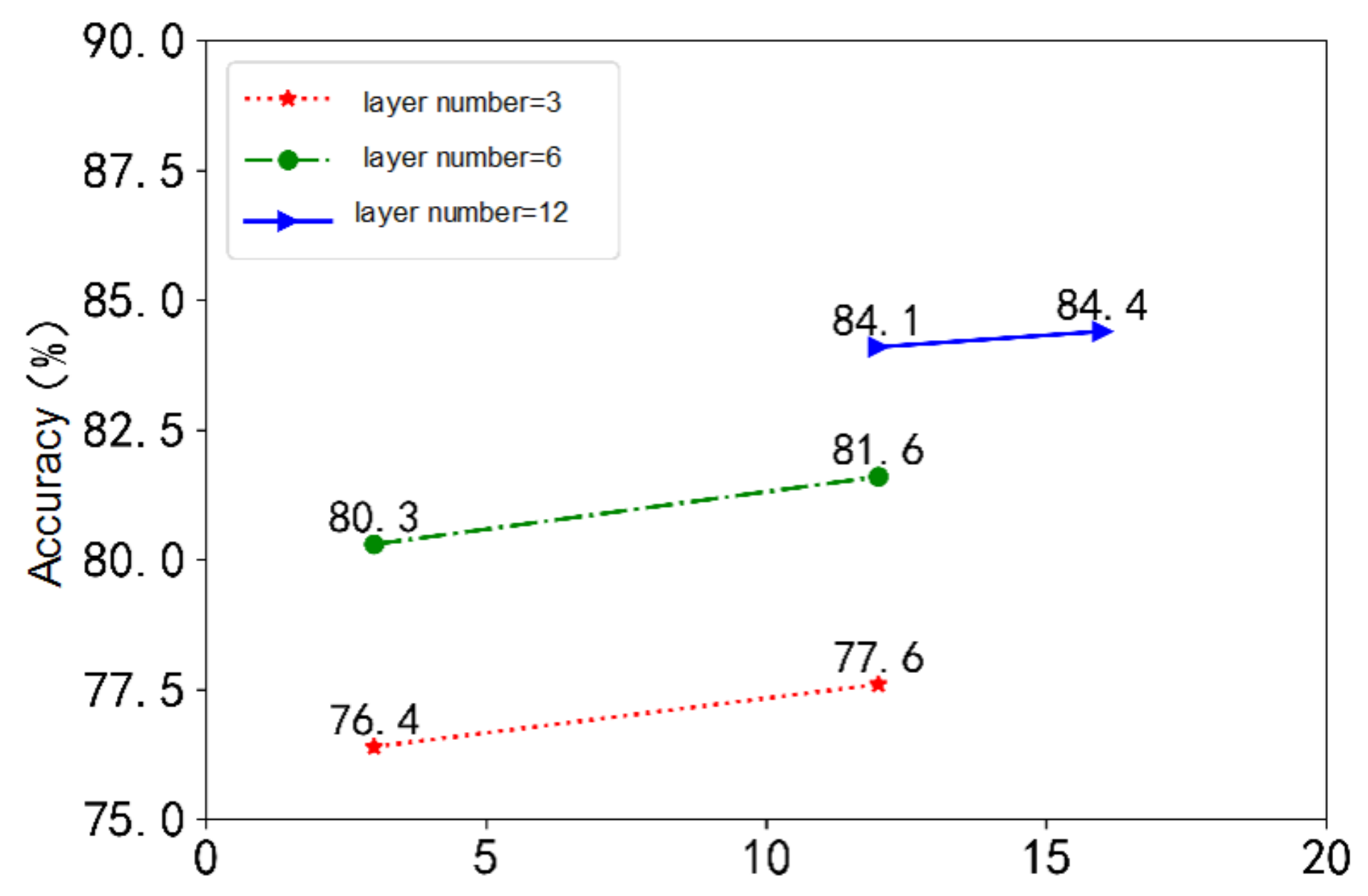
Results

CoDyWor VS Other Word Embedding



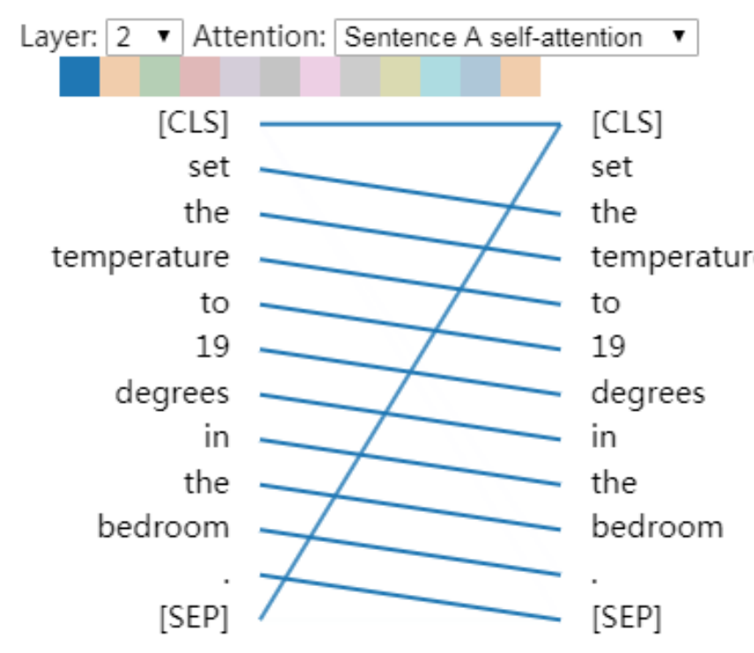
CoDyWor is significantly superior to the current popular word embedding methods in logical reasoning (MultiNLI dataset), named entity recognition (CoNLL03 dataset), and reading comprehension (SQuAD dataset) tasks.

Model capacity impact

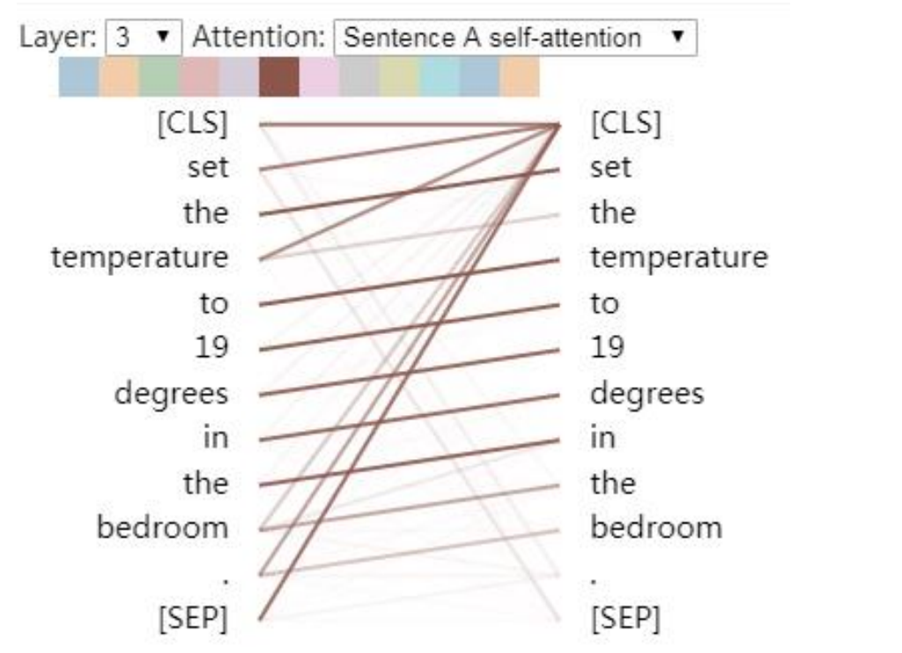


It can be found that increasing the number of layers of Transformer within a certain range or increasing the number of self-attention heads in Transformer can both improve the inference accuracy of the model.

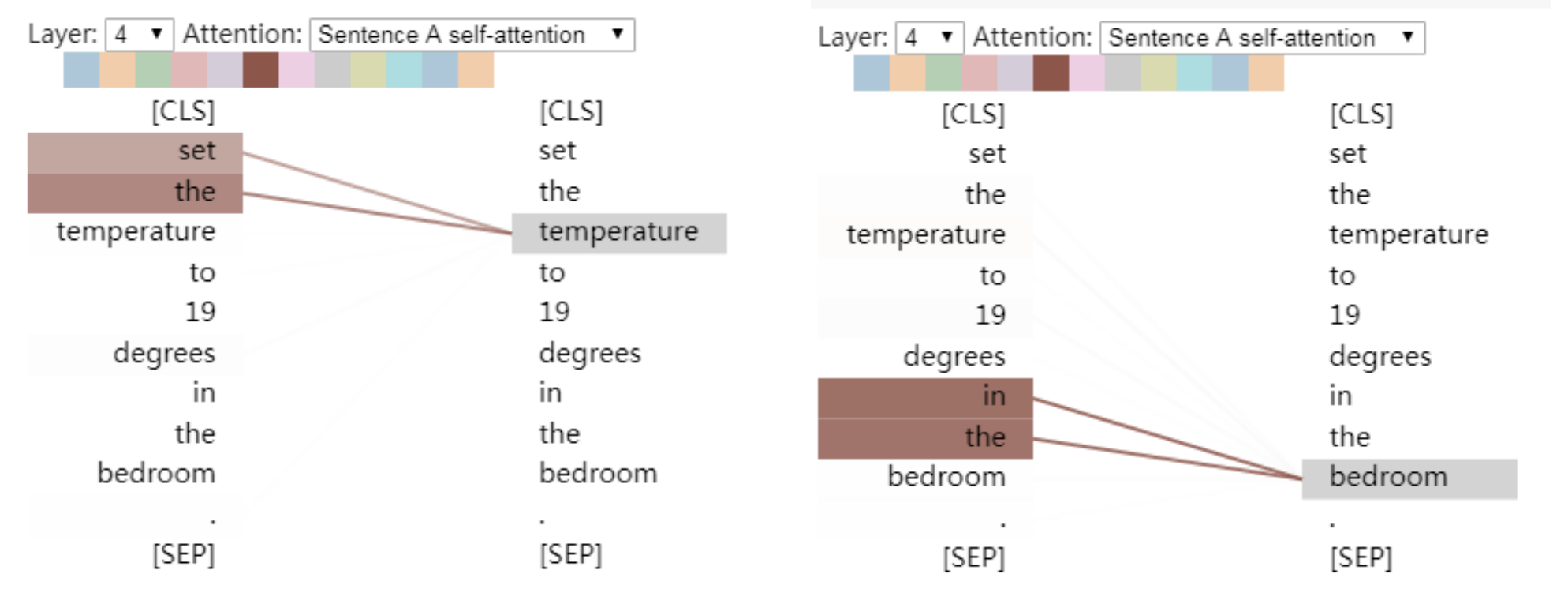
Visualizing Attention in the Transformer


Layer: 2 Attention: Sentence A self-attention
[CLS] set the temperature to 19 degrees in the bedroom . [SEP]

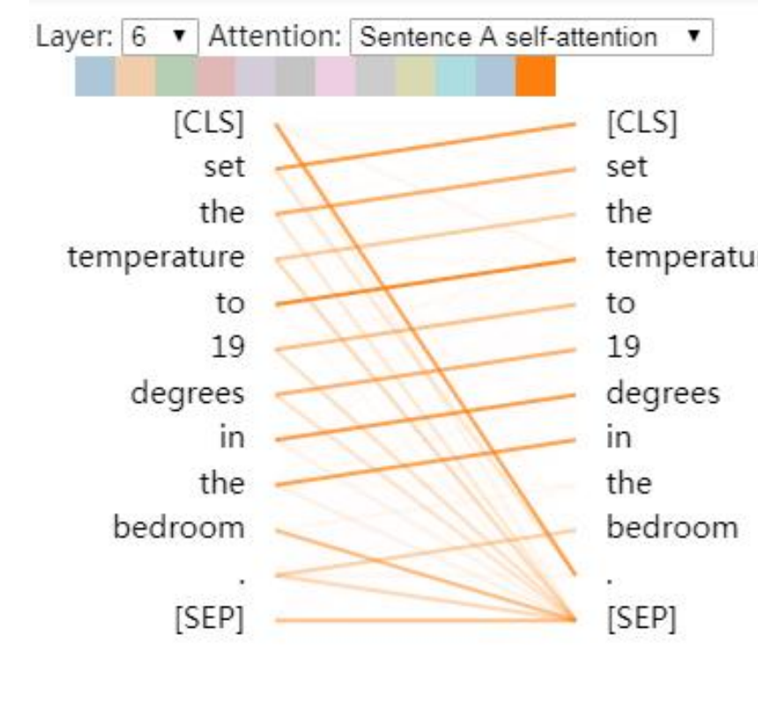
Attention to next word


Layer: 3 Attention: Sentence A self-attention
[CLS] set the temperature to 19 degrees in the bedroom . [SEP]

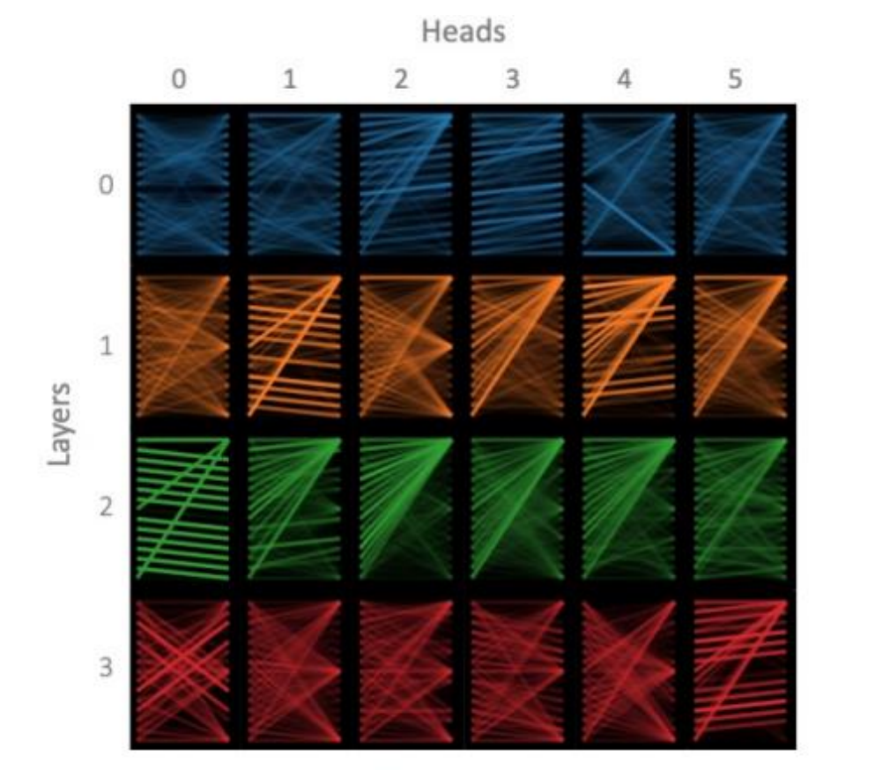
Attention to previous word


Layer: 4 Attention: Sentence A self-attention
[CLS] set the temperature to 19 degrees in the bedroom . [SEP]

Attention to identical/related words


Layer: 6 Attention: Sentence A self-attention
[CLS] set the temperature to 19 degrees in the bedroom . [SEP]

Attention to delimiter tokens



Attention across all of the model's layers and heads

Conclusion

This paper proposes an efficient, simply-structured deep context dynamic word representation model CoDyWor that can be widely used in natural language processing tasks. The context dynamic word representation generated by the model can be used for natural language processing tasks such as logical reasoning, named entity recognition and reading comprehension and so forth, and may be universally utilized to a certain extent. The context dynamic word representation generated by the CoDyWor model in the above tasks performs better than the current mainstream static word representations.

Acknowledgements

The work was partially supported by the China Postdoctoral Science Foundation under Grant No. 2019M653400; the Sichuan Science and Technology Program under Grant Nos. 2018GZ0253, 2019YFS0236, 2018GZ0182, 2018GZ0093 and 2018GZDZX0039.