vvPoem: An Open Source Toolkit for

Chinese poem generation based neural memory

Center for Speech and Language Technologies, Tsinghua University

August,2017

1. **Introduction**

vvPoem is part of the vivi project, an ambitious project aiming to brining up a

smart virtual girl with the capability of listening, speaking, drawing, writing,

pouting, consoling.... It is a girl of our lab, and grow with us.

This project was lauhed in CSLT at Tsinghua University in 2016, with the name

after our first female researcher Xuwei.

vvPoem is the first component: we let vivi can create traditional Chinese poems.

Our effort was started from the first half year of 2015, with some people involved,

including Chao Xing, Haichao Yu, Tianyi. However, only after the participantion

of Qixin Wang, progress was clearly achieved. The first breakthrough was taken

in Song Iambics generation[1], and then quatrains[2]. We obtained very impressive

results and experts participating the evaluation were largely astonished. Many

techniques have been developed, including hybrid training, theme-aligned prediction.

After Qixin's leaving, Jiyuan Zhang took over vvPoem. However, due to the inefficient

communication, the true progress was not obtained until late November, 2016. After

bitter struggling with code combing, Jiyuan, lead by Dr. Yang Feng, could finally

reproduce Qixin's result with Tensorflow (Qixin used Theano). This is an important step

as Tensorflow seems much faster than Theano. Additinoally, Jiyuan also investigated

style modification with memory mechanism[3].

vvPoem is an ongoing project. Our goal is to let vivi can generate much better poems

than top experts, quickly and stable. Much work to do.

1. **Installation**
	1. **System Requirements**

vvPoem supports Linux or MacOS, and requires python 2.7. Besides, we highly recommand running vvPoem on GPU servers.

* 1. **Installing Prerequisites**

**2.2.1 CUDA Environment**

If you choose to use CPU, please skip this step.

Assume you want to run ViVi\_NMT on NVIDIA GPUs and [the CUDA toolkit](https://developer.nvidia.com/cuda-toolkit) version 7.5 has been installed in "/usr/local/cuda-7.5/", then environment variables need to be set:

export PATH=/usr/local/cuda-8.0/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/local/cuda-8.0/lib64:$LD\_LIBRARY\_PATH

**2.2.2 Tensorflow 0.10**

To have tensorflow 0.10 installed, serval methods can be applied. You can choose the appropriate version of the installation according to your needs according to the official website

* 1. **vvPoem Directories**

The source code of vvPoem is available . If you want to use vvPoem, you just need download and unpack it.

Entering the vvPoem directory,you will find two folders(train,predict) :

|-train: generate the model by training the vvPoem system

|-predict: generate poem by the chosen model

Main scripts:

train.py : Entry script to start the training process

predict.py : Entry script to start the prediction process

seq2seq.py : Create attention model and memory model

seq2seq\_model.py : Define the model structure used inseq2seq.py

GlobalParams.py : Parameter setting

memoryModule\_decoder.py : Create style memory

data\_utils.py : Some unitility functions

1. **Running vvPoem**
	1. **Train**

To train the model, Enter the‘train’directory and run "train.py" directly with default settings.

python trian.py

Model parameters and training settings can be set by command-line arguments, as follows:

--max\_gradient\_norm: Clip gradients to this norm, default is 1.0.

--batch\_size: Batch size to use during training, default is 80.

--hidden\_units: Size of hidden units for each layer, default is 500.

--hidden\_edim: Dimension of word embedding, default is 500.

--num\_layers: Number of layers of RNN, default is 1.

--keep\_prob: The keep probability used for dropout, default is 1.0.

--src\_vocab\_size: Vocabulary size of source language, default is 4179.

--trg\_vocab\_size: Vocabulary size of target language, default is 4179.

--data\_dir: Data directory, default is './data'.

--train\_dir: Training directory, default is './train/.

--max\_train\_data\_size: Limit on the size of training data (0: no limit), default is 0.

--steps\_per\_checkpoint: How many training steps to do per checkpoint, default is 1000.

Note that, we provide a poem dataset in 'resource/train\_data’, with 58 thousand poems in training set.

* 1. **Prediction**

To generate poem , Enter the‘train’directory, run the command below.

python memoryModule\_decoder.py --model --memory\_resource --type

python predict.py --model –memory\_resource –-memory\_weight --round

Model parameters should be the same as settings when training, and other parameters for decoding are as follows.

--memory\_resource: the resource used to generate memory.

--model: The checkpoint model to load.

--type: the format of the memory\_resource.

--rounds: start with 10, add one iteration per round, repeat rounds , to test the effect of the different round.


**Reference**

1. Qixin Wang, Tianyi Luo, Dong Wang, Chao Xing, "Chinese Song Iambics Generation with Neural Attention-based Model", IJCAI 2016

https://arxiv.org/abs/1604.06274

2. Qixin Wang, Tianyi Luo, Dong Wang, "Can Machine Generate Traditional Chinese Poetry? A Feigenbaum Test", BICS 2016

http://wangd.cslt.org/public/pdf/poem-bics16.pdf

3. Jiyuan Zhang, Yang Feng and Dong Wang, " Flexible and Creative Chinese Poetry Generation Using Neural Memory"

http://www.aclweb.org/anthology/P/P17/P17-1125.pdf