Digital Speech Processing
Final Project Report

Paper Survey About Language Model and Chinese Lyrics Decoding Using Trigram Language Model

電機四 林孟瑾 B04901009

1 Paper survey about language model

I really interested in the concepts of language model so I search for renown papers to understand more about the developments of language model.

1. A neural probabilistic language model [1]

According to Google Scholar website, there are 4685 citations about this paper.

(1) Introduction of this work

The most difficult issue in language modeling is curse dimensionality. Statistical language modeling utilizes word order to address this problem by constructing conditional probability between the neighboring words and using back-off and smoothing methods to deal with the words absent in training corpus. However, the author thinks that there are two issues that can be improved.

i. Contexts farther than 2 words
ii. Semantic and grammar similarity

To achieve these two goals, this work uses neural network model to deal with high-dimensional discrete distribution. It aims to train continuous distributed feature vectors and express the joint probability function of word sequences based on word feature vectors.

(2) Neural network model

The training set is a sequence of words of vocabulary V which is large but finite set. The objective is to get the model $f(w_t,\dots,w_{t-n+1}) = \hat{P}(w_t|w_1^{t-1})$ which gives high out-of-

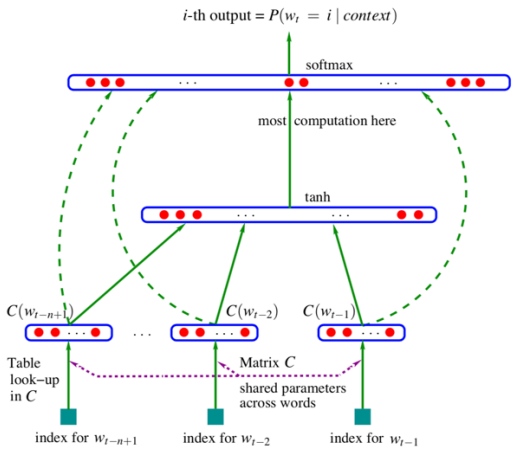
sample likelihood. There are two parts of $f(w_t, \ldots, w_{t-n+1}) = \hat{P}(w_t|w_1^{t-1})$:

i. Distributed feature vectors relating to each word in the vocabulary, expressed by a mapping C from any element i of V to a real vector $C(i) \in \mathbb{R}^m$

ii. Probability function over words, expressed by a function g mapping input sequence of feature vectors for words to a conditional probability distribution over words in V for the next word w_t.

Therefore, the neural architecture as the following graph is a composition of two mappings C and g. C can be shared across all words in the context.

$$f(i, w_{t-1}, \ldots, w_{t-n+1}) = g(i, C(w_{t-1}), \ldots, C(w_{t-n+1}))$$



Training process includes looking for overall parameter set $\theta = (C, \omega)$ which maximizes L:

$$L = \frac{1}{T}\sum_t logf(w_t, w_{t-1}, \ldots, w_{t-n+1}; \theta) + R(\theta)$$

(3) Parallel Implementation

The computations of neural network model are far larger than traditional n-gram model. Nowadays, people use GPU to train neural network models to reduce computation time. So I think that GPU might not be common at the time this work published.

The parts requiring most computations are the activations of the output layers. Therefore, this work tries the methods of data parallel processing and parameter parallel processing to deal with immense computations.

i. Data parallel processing
Each processor processes different subset of data and store data in the shared memory. The asynchronous implementation is used to allow each processor write data at any time without waiting others to release the lock. The noise that some processors overwrite data is negligible. However, shared memory parallel computer is too expensive.

ii. Parameter parallel processing
This method is about parallelizing across the parameters of the output units. Each CPU computes the un-normalized probability for a subset of output data and updates the corresponding parameters. In parallelized stochastic gradient ascent, CPUs need to communicate the normalization factor of the output softmax and the gradients on the hidden layer and word feature layer. Still, there will be some computation duplicates among CPUs which are negligible.

The computation for processor is summarized as follows.

i. Forward phase
   a. Forward computation for the word features layer
   b. Forward computation for the hidden layer
   c. Forward computation for output units in i-th block
   d. Computation and sharing the sum of the probabilities for output units among processors using MPI library
   e. Probability normalization
   f. Log-likelihood update

ii. Backward/update phase
   a. Backward gradient computation for output units in i-th block
   b. Summing and sharing $\frac{\partial L}{\partial x}$ and $\frac{\partial L}{\partial a}$ across processors using MPI library
   c. Hidden layer weights back-propagation and update.
   d. Word feature vectors update for the input words

In addition, the mini-batch method is mentioned in the paper. By performing communications every K examples, it can reduce communication latency and save time.

(4) Result
The performance of neural network model is compared to n-gram models based on Brown corpus and AP News corpus using perplexity.

i. Neural network can take advantage of more context
ii. Hidden units are useful
iii. Mixing output probabilities of neural network with interpolated trigram is of good performance

(5) Extension and future work
i. An energy minimization network
In this model, output words also are mapped as its feature vector. The output of the model is as following:

$$E(w_{t-n+1},\dots,w_t) = v.\tanh(d + Hx) + \sum_{i=0}^{n-1} b_{w_{t-i}}$$

where b is the vector of biases, d is the vector of hidden units biases, v is the output weight vector and H is the hidden layer weight matrix. And that input and output words contribute to x:

$$x = (C(w_t), C(w_{t-1}), C(w_{t-2}), \dots, C(w_{t-n+1}))$$

And the conditional probability is computed as follows:

$$\hat{P}(w_t|w_{t-1},\dots,w_{t-n+1}) = \frac{e^{-E(w_{t-n+1},\dots,w_t)}}{\sum_i e^{-E(w_{t-n+1},\dots,w_{t-1},i)}}$$

In addition, this model can handle OOV easily than previous model. It takes weighted convex combination of feature vectors of other words with the weights proportional to their conditional probability. And the new word can be incorporated in V and re-compute the probability distribution in the new set.

ii. Decomposing the network in sub-network to train the model more efficiently

iii. Representing conditional probability using tree structure to reduce computation time by a factor of $\frac{|V|}{\log |V|}$

iv. Propagating gradients only from a subset of the output words by selecting important words to accelerate training

v. Introducing a-priori knowledge including semantic and grammar information

vi. Interpreting the word feature representation learned by neural network model

vii. Associating each word with multiple points in word feature space to handle polysemous words.

(6) Conclusion

The neural network model proposed in this paper uses learned distributed representation to inform model about the combinatorial number of other sentences. In addition, this smooth and compact representation can accommodate more conditioning variables to improve statistical language model.

2 Chinese lyrics decoding using trigram language model

1. Introduction

When I listen to the Chinese songs, I often misunderstand the exactly words in the lyrics because the tones of the words cannot be recognized in the songs. Human can figure out the possible lyrics using our language knowledge. How about the computer? While taking DSP course, I constantly think about how to solve this problem using computer program. I think this is a practical issue in life so I decide to handle this problem in the final project.

This computer program aims to decode purely ZhuYin lyrics sequences into Chinese characters using language model.

Input: ㄌㄧㄤˇㄓㄨㄌㄧㄤˇㄓㄨㄆㄠˇㄉㄜˊㄎㄨㄞˋ
Output: 兩隻老虎兩隻老虎跑得快

It contains two main steps:
1. Language model training
   Train the trigram language model using srilm library [2] and lyric corpus collected from website [3]. While constructing the corpus, compute the perplexity to see the improvement of performance of language model.
2. ZhuYin lyrics sequences decoding
   Using language model to decode the sequences made of solely ZhuYin. Compare the results with using trigram language model trained by corpus in hw3.

2. Execution

The commands corresponding to each task are as following.
1. Language model training
   a. Collect new lyrics data in “lyrics_utf.txt”
   b. Type “bash train_corpus.sh” to expand lyrics corpus and train trigram language model
   c. Type “make run_perplexity” to compute perplexity of the language model by running calculate_perplexity.cpp

2. ZhuYin lyrics sequences decoding
   a. Put the testing Chinese lyrics in directory finaldata/ and name it test_ori.txt.
   b. Type “bash convert_test_zhuyin.sh” to convert test_ori.txt to purely Zhuyin sequences test_zhuyin.txt.

c. Type “make run” to produce result file in result_final/run/run_test.txt
d. Type “make run_compare” to produce result file using trigram language model trained by hw3 corpus in result_final/run_compare/run_compare_test.txt

4. Details

I collect lyrics data of Chinese pop songs from website [3] and use srilm library [2] to train trigram language model. During data collection and training, I examine the change of perplexity of language model with respect to testing data. The sentences in testing data are completely different from the training corpus.

1. Data collection

Since that some characteristics of Chinese lyrics are different from the normal documents such as news reports or articles, it is necessary to use lyrics data set to train language model. There are some traits of Chinese lyrics which are different from normal documents:

a. Repetition
   Ex. 兩隻老虎兩隻老虎　跑得快跑得快
b. Inversion
   Ex. 當有你的溫熱我想我很快樂
c. Rhyme
   Ex. 一閃一閃亮晶晶 滿天都是小星星 掛在天空放光明

Therefore, using the data set of Chinese lyrics, the language model can “learn” various characteristics of this kind of data. I collect lyrics data of several Chinese pop music singers’ songs from website [3] to construct lyrics corpus.

Besides, it is important to handle big5 format correctly when collecting data. I convert the original format when copy data from website (utf-8) into big5 by the following command in train_corpus.sh.

```
cat corpus_utf_train.txt | iconv -c -f UTF-8 -t BIG5 > big_corpus_train_big5.txt
```

2. Perplexity computation

At first, I refer to class materials for the computation of perplexity and I use the total number of words in the testing corpus for the N_D in the formula. However, I find that the result of perplexity is strange. So I search for some information about the computation of perplexity. And I found that N_D is related to the number of counts of the n-grams.

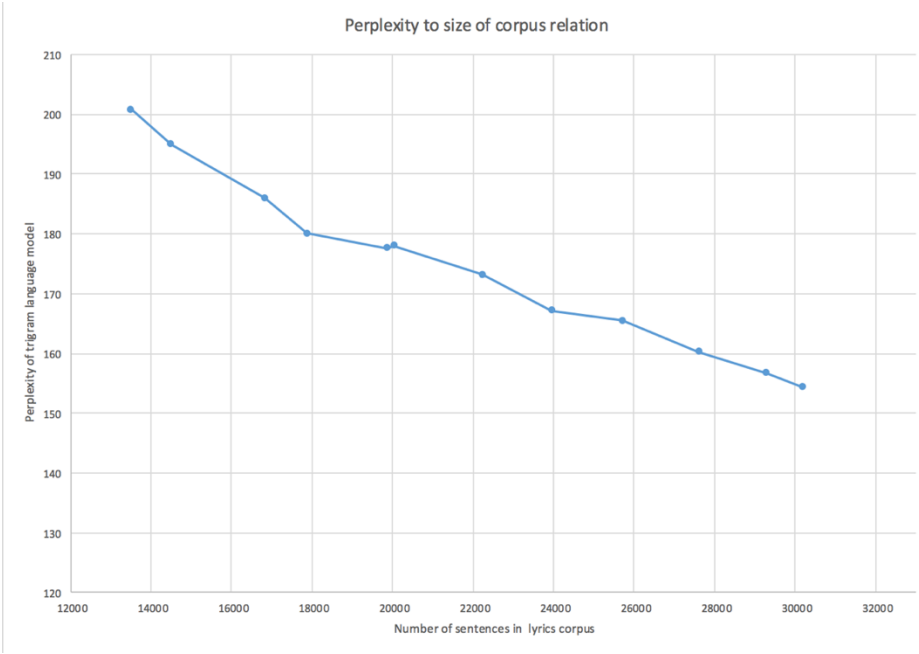
To compute the perplexity and verify performance of language model, I collect some random lyrics without repeating with the data in the corpus to construct testing corpus which contains 733 lines of lyrics.

The following chart shows part of the process of collecting different singer's lyrics set from website [3] and the trend that perplexity decreases while constructing the lyrics corpus. For each singer, I collect all of their songs they published before. It shows that the performance respective to testing corpus is gradually improved. In "吳亦凡"'s case, though, adding lyrics of 吳亦凡's songs results in the increase of perplexity. It might because that lyrics data in this case is not useful for the testing corpus. The lyrics corpus after removing empty lines are saved in "corpus_utf.txt".

	Perplexity	Number of sentences	Average reduced perplexity per sentence
The corpus before adding the following singers' lyrics	200.808	13510	
1.蔡健雅	194.92	14507	-0.00590572
2.王心凌	186.011	16828	-0.00383843
3.李宗盛	180.097	17882	*-0.00561101*
4.方大同	177.639	19865	-0.0012395

5.吳亦凡	178.018	20050	0.0020487
6.蕭敬騰	173.14	22241	-0.0022264
7.許茹芸	167.175	23957	-0.0034761
8.張震嶽	165.481	25714	-0.0009641
9.謝和弦	160.232	27622	-0.0027511
10.李榮浩	156.687	29298	-0.0021152
11.田馥甄	154.322	30211	-0.0025904

The following graph shows the degree of reducing perplexity of each singer’s data. The data of “李宗盛” has the greatest performance of reducing perplexity. It might be due to that his songs seldom repeat lyrics compared to other singers. Or that his songs have more variable words and structures of sentences to improve language model.



5. Results

1. Children song lyrics test data

I train the trigram language model using the corpus in hw3 to compare the results of small test data containing some children song lyrics. Since the corpus in hw3 contains mainly news documents, the performance of resulting trigram_ref model is worse than the trigram_train which use lyrics corpus to train. As a result, data collection can extremely influence the performance of certain tasks.

answer 兩隻老虎兩隻老虎 跑得快跑得快 一閃一閃亮晶晶 滿天都是小星星 掛在天上放光明 好像許多小眼睛 妹妹背著洋娃娃 走到花園來看花 娃娃哭著叫媽媽	result of trigram_train	<s> 兩 隻 老 虎 兩 隻 老 虎 </s> <s> 跑 的 快 跑 的 塊 </s> <s> 一 閃 一 閃 亮 晶 晶 </s> <s> 滿 天 都 是 小 星 星 </s> <s> 掛 在 天 上 放 光 明 </s> <s> 好 像 許 多 笑 眼 睛 </s> <s> 沒 沒 背 著 洋 娃 娃 </s> <s> 走 到 話 原 來 看 花 </s> <s> 娃 娃 哭 著 腳 媽 媽 </s>
	result of trigram_ref	<s> 兩 隻 老 虎 兩 隻 老 虎 </s> <s> 跑 得 快 跑 得 快 </s> <s> 以 善 意 閃 亮 晶 晶 </s> <s> 滿 天 都 是 小 行 星 </s> <s> 掛 在 天 上 放 光 明 </s> <s> 好 像 許 多 小 眼 睛 </s> <s> 妹 妹 被 這 樣 娃 娃 </s> <s> 走 到 花 源 來 看 花 </s> <s> 娃 娃 哭 著 腳 媽 媽 </s>

2. Pop song lyrics test data

After converting Chinese lyrics data into purely ZhuYin sequences, I compare the results between two trigram language model using famous pop songs. Some songs are new publications while some are the old singers' works. Generally, the performance of trigram language model trained by lyrics corpus to decode ZhuYin sequences is much better.

	accuracy of trigram_train	accuracy trigram_ref	accuracy improvement
周杰倫 - 不愛我就拉倒	0.835735	0.723343	0.155379
周杰倫 - 等你下課	0.887701	0.794118	0.117845
周杰倫 - 千里香	0.679104	0.597015	0.137499
陳綺貞 - 沙發海	0.845411	0.763285	0.107595
陳綺貞 - 變色龍	0.915966	0.840336	0.089999
陳綺貞 - 蜉蝣	0.954198	0.732824	0.302084

鄧麗君 - 月亮代表我的心	0.958549	0.854922	0.121212
鄧麗君 - 甜蜜蜜	0.888268	0.581006	0.528845
鄧麗君 - 夜來香	0.664894	0.478723	0.388891
鄧麗君 - 但願人長久	0.482517	0.391608	0.232143
王傑 - 一場遊戲一場夢	0.905882	0.885294	0.023256
王傑 - 忘了你忘了我	0.923077	0.852564	0.082707
王傑 - 流浪的心	0.814159	0.725664	0.121950
張學友 - 吻別	0.927807	0.815508	0.137704
張學友 - 一路上有你	0.747312	0.642473	0.163180
張學友 - 一千個傷心的理由	0.877049	0.729508	0.202247

The following graph shows the lyrics of the famous song “鄧麗君 - 甜蜜蜜”. And the chart shows the results of decoding test_zhuyin.txt

甜蜜蜜你笑得甜蜜蜜
好像花兒開在春風裡 開在春風裡
在那裡在那裡見過你 你的笑容這樣熟悉
我一時想不起
啊 在夢裡
夢裡夢裡見過你 甜蜜笑得多甜蜜
是你是你夢見的就是你
在那裡在那裡見過你 你的笑容這樣熟悉
我一時想不起
啊 在夢裡

test_zhuyin.txt	ㄊㄧㄢㄇㄧㄢㄋㄧˇㄒㄧㄠㄉㄜㄊㄧㄢㄇㄧㄢㄇㄧ ㄏㄠㄒㄧㄤㄏㄨㄚㄦㄎㄞㄗㄞㄔㄨㄣㄈㄥㄌㄧˇ ㄗㄞㄋㄚㄌㄧˇㄗㄞㄋㄚㄌㄧˇㄐㄧㄢㄧㄋㄧˇㄉㄜㄧㄠㄖㄨㄛㄧㄠㄒㄧˇ ㄨㄛㄧㄕㄒㄧㄤㄅㄨㄑㄧ ㄚㄗㄞㄇㄥˋㄌㄧˇ ㄇㄥˋㄌㄧˇㄇㄥˋㄌㄧˇㄐㄧㄢㄧㄋㄧˇㄊㄧㄢㄇㄧㄢㄒㄧㄠㄉㄜㄉㄨㄛㄊㄧㄢㄇㄧ ㄕㄋㄧˇㄕㄋㄧˇㄇㄥˋㄐㄧㄢㄉㄜㄐㄧㄡㄕㄋㄧˇ ㄗㄞㄋㄚㄌㄧˇㄗㄞㄋㄚㄌㄧˇㄐㄧㄢㄧㄋㄧˇㄊㄧㄢㄇㄧㄢㄒㄧㄠㄉㄜㄧㄠㄖㄨㄛㄧㄠㄒㄧˇ ㄨㄛㄧㄕㄒㄧㄤㄅㄨㄑㄧ ㄚㄗㄞㄇㄥˋㄌㄧˇ
result of trigram_train	<s> 甜 蜜 蜜 你 笑 的 甜 蜜 蜜 </s> <s> 好 像 花 兒 開 在 春 風 離 開 在 春 風 裡 </s> <s> 在 哪 裡 在 哪 裡 見 過 你 你 的 笑 容 這 樣 熟 悉 </s> <s> 我 已 是 想 不 起 </s> <s> 啊 在 夢 裡 </s> <s> 夢 裡 夢 裡 見 過 你 甜 蜜 笑 得 多 甜 蜜 </s> <s> 是 你 是 你 夢 見 的 就 是 妳 </s> <s> 在 哪 裡 在 哪 裡 見 過 你 你 的 笑 容 這 樣 熟 悉 </s> <s> 我 已 是 想 不 起 </s> <s> 啊 在 夢 裡 </s>
result of trigram_ref	<s> 甜 蜜 迷 你 小 的 甜 蜜 蜜 </s> <s> 好 像 華 爾 凱 在 春 風 力 凱 在 春 風 力 </s> <s> 在 那 裡 在 那 裡 建 國 尼 尼 的 笑 容 這 樣 熟 悉 </s> <s> 我 一 事 項 不 起 </s> <s> 阿 災 盟 力 </s> <s> 孟 麗 孟 力 建 國 擬 甜 蜜 小 得 多 田 迷 </s> <s> 使 你 是 妮 夢 見 得 就 是 你 </s> <s> 在 那 裡 在 那 裡 建 國 尼 尼 的 笑 容 這 樣 熟 悉 </s> <s> 我 一 事 項 不 起 </s> <s> 阿 災 盟 力 </s>

In addition, the following graph shows the lyrics of another famous song “曾淑勤 - 魯冰花”. And the chart shows the results of decoding test_zhuyin.txt

當手中握住繁華 心情卻變得荒蕪
才發現世上 一切都會變卦
當青春剩下日記 烏絲就要變成白髮
不變的只有那首歌 在心中來回的唱
天上的星星不說話 地上的娃娃想媽媽
天上的眼睛眨呀眨 媽媽的心呀魯冰花
家鄉的茶園開滿花 媽媽的心肝在天涯
夜夜想起媽媽的話 閃閃的淚光魯冰花

test_zhuyin.txt	ㄉㄤ ㄕㄡ ㄓㄨㄥ ㄨㄛ ㄓㄨ ㄔㄤ ㄈㄢ ㄖㄨㄥ ㄒㄧㄣ ㄑㄧㄥ ㄑㄧㄝ ㄈㄚ ㄨㄟ ㄅㄧㄢ ㄘㄞ ㄈㄚ ㄒㄧㄢ ㄕˋ ㄕㄤˋ ㄧ ㄑㄧ ㄉㄡ ㄏㄨㄟ ㄅㄧㄢ ㄅㄨ ㄉㄤ ㄑㄧㄥ ㄔㄨㄣ ㄕㄥˋ ㄒㄧㄚ ㄖˋ ㄐㄧˋ ㄨˇ ㄙ ㄐㄧㄡ ㄧㄠˋ ㄅㄧㄢ ㄔㄥˊ ㄅㄞˊ ㄈㄚ ㄅㄨˋ ㄅㄧㄢˋ ㄉㄜ ㄓˇ ㄧㄡˇ ㄋㄚˋ ㄕㄡˇ ㄍㄜ ㄓㄨㄥ ㄌㄞˊ ㄏㄨㄟ ㄉㄜ ㄔㄤˋ ㄊㄧㄢ ㄕㄤˋ ㄉㄜ ㄒㄧㄥ ㄒㄧㄥ ㄅㄨˋ ㄕㄨㄛ ㄏㄨㄚˋ ㄉㄧˋ ㄕㄤˋ ㄉㄜ ㄨㄚ ㄨㄚ ㄒㄧㄤˇ ㄇㄚ ㄇㄚ ㄊㄧㄢ ㄕㄤˋ ㄉㄜ ㄧㄢˇ ㄐㄧㄥ ㄓㄚˋ ㄎㄚ ㄓㄚˋ ㄇㄚ ㄇㄚ ㄉㄜ ㄒㄧㄣ ㄍㄚ ㄌㄨˋ ㄅㄧㄥ ㄏㄨㄚˋ ㄐㄧㄚˇ ㄒㄧㄤˇ ㄉㄜ ㄔㄚˊ ㄩㄢˇ ㄎㄞ ㄇㄢˇ ㄏㄨㄚ ㄇㄚ ㄇㄚ ㄉㄜ ㄒㄧㄣ ㄍㄢ ㄗㄞˋ ㄊㄧㄢ ㄒㄧㄚˋ ㄎㄜ ㄧㄝ ㄧㄝ ㄒㄧㄤˇ ㄑㄧˇ ㄇㄚ ㄇㄚ ㄉㄜ ㄏㄨㄚ ㄕㄢ ㄕㄢ ㄉㄧˋ ㄌㄟˋ ㄍㄨㄤ ㄅㄧㄥ ㄏㄨㄚˋ ㄎㄜ
result of trigram_train	<s> 當 手 中 握 住 繁 花 心 情 卻 變 的 荒 蕪 </s> <s> 才 發 現 世 上 一 切 都 會 變 卦 </s> <s> 當 青 春 盛 夏 日 記 五 四 就 要 變 成 白 髮 </s> <s> 不 變 的 只 有 那 首 歌 在 心 中 來 會 的 唱 </s> <s> 天 上 的 星 星 不 說 話 地 上 的 娃 娃 想 媽 媽 </s> <s> 天 上 的 眼 睛 炸 呀 炸 媽 媽 的 新 芽 路 冰 話 </s> <s> 假 想 的 差 遠 開 滿 花 媽 媽 的 心 肝 在 天 呀 </s> <s> 夜 夜 想 起 媽 媽 的 花 閃 閃 的 淚 光 路 冰 話 </s>
result of trigram_ref	<s> 當 手 中 握 主 繁 花 心 情 卻 變 的 荒 蕪 </s> <s> 才 發 現 實 上 一 切 都 會 變 卦 </s> <s> 當 青 春 生 下 日 及 五 四 九 要 變 成 敗 法 </s> <s> 不 便 的 只 有 那 首 歌 仔 信 中 來 會 的 廠 </s> <s> 天 上 的 行 省 部 說 話 地 上 的 哇 哇 向 媽 媽 </s> <s> 天 上 的 眼 睛 眨 呀 眨 媽 媽 的 新 亞 陸 炳 化 </s> <s> 家 鄉 的 茶 園 開 漫 畫 媽 媽 的 新 幹 再 添 壓 </s> <s> 爺 爺 想 起 媽 媽 的 話 山 山 地 雷 光 路 並 劃 </s>

It can be shown that tasks relating to specific language domain can be achieved by training the data set of that specific domain more effectively.

3 Reference

[1] Yoshua Bengio, Rejean Ducharme and Pascal Vincent. 2003. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137-1155

[2] http://www.speech.sri.com/projects/srilm/

[3] https://www.kkbox.com/hk/tc/artist/