Computational Linguistics Final Project Report
William LaCroix
Matrikulnummer: 7038732
Winter Semester 2022/2023

The intent behind this project was to implement a Long Short-Term Memory (LSTM) language model using an unannotated corpus of text in Mandarin Chinese, which would then be fine tuned the LSTM via curated mini-corpora based on the standardized Chinese proficiency test – Hanyu Shuiping Kaoshi (HSK) – with levels HSK 1 (roughly equivalent to CEFR A1) through HSK 6 (roughly equivalent to CEFR C1). Language models fine-tuned at each of these proficiency levels could then be used to generate graded texts at the appropriate level for a user learning Mandarin Chinese, in order to supplement their input via new and varied sentences. This helps to solve one of the fundamental problems with typical spaced repetition methods of language learning, like flashcards, where the input is static, and the learner ends up learning words in only a specific context. Native input is one of the primary tools a language learner can use to expose themselves to content in varied contexts, but by its nature, the difficulty of native content is not suitable for the majority of language learners. The middle ground between these two input methods are texts referred to as "graded readers", where the vocabulary and grammar is gated by proficiency level. Thus the goal of this project is to train six separate language models, each capable of producing graded, grammatically accurate texts suitable for learners at various levels of proficiency in Mandarin Chinese.

The first hurdle in this project was curating the primary corpus for training the initial LSTM model. I began with an annotated treebank CHTB (Xue, 2013), but was unable to get either the LSTK tree parser or my own XML parser to iterate over the entries, due to a combination of file encoding and data storage convention issues. I had similar encoding and parsing issues with the Lancaster Corpus of Mandarin Chinese (McEnery & Xiao, 2004), which used an XML-encoded, POS-annotated, string-literal sentence representation. I was able to make significantly more progress using the zhTenTen Corpus, from the TenTen Corpus Family (Jakubíček, et al., 2013), which comprised one million lines of text in each of three categories: news, wiki, and web. These texts were scraped from the web, and in order to clean the files for preprocessing, I standardized the punctuation between various full-width and half-width characters, converted numerals to chinese characters, and removed any other text that was not a GBK encoder-compliant Chinese character. The unintended side effect of all of this text scrubbing was that after the initial training of the base LSTM on either corpus, the text output was absolute gibberish, and the neural network's (NN) loss function stayed unacceptably high. More on that later in the evaluation, but suffice it to say: "garbage in, garbage out."

My hope for the final language models was that fine tuning the graded LSTMs would help to patch up the shaky grammar and incoherent ramblings caused by the zhTenTen corpus' aggressive scrubbing. Which leads to the model's current greatest shortcoming: the mini-corpora. In order to source enough examples of graded text, I settled on asking chatGPT (at the time of writing, the freely available version of chatGPT is based on the GPT-3.5 language model) to generate example sentences based on the prompt:

> "You speak Mandarin Chinese at an HSK (n) level. Please generate 100 unique sentences, where each includes HSK (n) vocabulary and grammar, with no pinyin or translation."

Repeating this prompt, I gathered 500 unique sentence samples for each of the six proficiency levels. At which point I decided to quickly test train the LSTM on a smaller corpus consisting of the combined 3,000 example sentences generated by chatGPT, and it was immediately obvious, based on cross-entropy loss and visual inspection, that the new language model was more coherent and grammatically correct. Or at least overfitted. Thus I decided to move forward using the new 3,000 sentence chatGPT corpus as the baseline model, and fine tune each graded model based on the subset mini-corpora. Each corpus used for training was then preprocessed by building index: character and character: index dictionaries, which allowed for the texts to be represented as lists of integer indexes, as opposed to unicode character values.

Since this was my first foray into neural networks and LSTMs, the model architecture follows the open source guide "Text generation via RNN and LSTMs (PyTorch)" by Purva Singh (Singh, 2020). The NN was fed batched data via a dataloader object, which is essentially responsible for setting up Pytorch tensors of word indexes to be used in calculating the word embeddings. The general architecture of the language model involves three layer types: an embedding layer which is responsible for converting the batched tensors into word embeddings which can be passed as input to the contiguous LSTM layer, which itself has two connected layers of LSTM cells. The cells in the contiguous layer take as input either the word embeddings from the embedding layer, or the output and hidden state from the previous contiguous LSTM layer, and in turn produce an output as well as a hidden state. This is the forward propagation of the LSTM layer. This output and hidden state are then passed either to the next contiguous LSTM layer, or to the fully connected layer, which is responsible for mapping the output and hidden state to a probability distribution over the next most likely word in the model's vocabulary. During generation, the probability distribution is conditioned based on a priming character for generation of the first generated character in text, which is in turn used to condition the next character, and so on.

- Explanation of the fine-tuning process and hyperparameter tuning

During the initial training on the 1 million line zhTenTen corpus (initially tested on a 10 thousand line version), I tested several iterations while varying the model's hyperparameters in order to convince myself that I was using the best possible model to learn this specific data. Variations of the following were each tested individually, on either 10k or 1M (or both) zhTenTen corpora, and the cross-entropy loss values were compared:

- Number of Epochs: how many times to cycle through the data for training.
- Learning Rate: how much weight or influence each new cycle is given.
- Dropout: how often or strongly the model 'forgets' data. Used during Forget Gate.
- Embedding Dimension: number of features with which to vectorize text input.
- Hidden Dimension: must be compatible with embedding dimension for matrix math.
- Number of RNN Layers: how many fully connected LSTM cell layers to train over.

Tests 1 through 9 all had a cross-entropy loss value converging between 4.5 and 6.2, which was an overall improvement over the untrained model's loss value around 7.5, left a lot to be desired. After settling on values for the hyperparameters based on the results from the zhTenTen corpora, I had finally curated the graded corpora via chatGPT, and decided to train a new model based on the 3,000 sentence corpus of all six levels. The results are listed in the notebook as Test 10. This time, the same hyperparameters which lead to convergence around 4.5 ended up at a cross-entropy loss of only 0.4 after approximately 90 epochs. This was obviously a huge improvement, but whether that is due to the model having cleaner and more grammatically correct text to learn from, or just because the compiled HSK corpus was significantly smaller than either of the zhTenTen corpora, leading to overfitting, it's hard to say. What was immediately apparent, however, was that the initial results of training over the HSK corpus produced much more coherent, if less varied, texts during generation. But I'm getting ahead of myself.

During training and hyperparameter tweaking, the primary method of evaluation for the model was its cross-entropy loss value, which uses a softmax function to quantify the difference between the model's output and the 'correct prediction'. Approximate threshold values for qualifying the results of a given model are given below (Wang, 2021):

- Cross-Entropy = 0.00: Perfect probabilities.
- Cross-Entropy < 0.02: Great probabilities.
- Cross-Entropy < 0.05: On the right track.
- Cross-Entropy < 0.20: Fine.
- Cross-Entropy > 0.30: Not great.
- Cross-Entropy > 1.00: Terrible.
- Cross-Entropy > 2.00 Something is broken.

Based on these metrics, the pretrained NN model based on the zhTenTen corpus does not even come close to a reasonable threshold, whereas the pretrained model based on the HSK corpus scores a solid “not great”, which is clearly an improvement. Results of the initial training can be found in Appendix A

Examples of the generated text are included in Appendix B. As a metric for whether the graded model learned language at the various proficiency levels, I decided to count the number of unique characters in a generated text of 10,000 characters at each proficiency level. This exposed one of the weaknesses of my GPT HSK corpora, in that the number of unique characters encountered in training is not accurate to the number of unique characters in the official word lists for those levels.

Trial generation of 10,000 characters at each level produced by priming with:
['我', '你', '他', '她', '这', '人', '的', '是', '那', '一']

- HSK 1: 352 (366 in training data vs 177 in official word list)
- HSK 2: 428 (458 in training data vs 358 in official word list)
- HSK 3: 535 (579 in training data vs 624 in official word list)
- HSK 4: 628 (711 in training data vs 1076 in official word list)
- HSK 5: 670 (736 in training data vs 1712 in official word list)
- HSK 6: 723 (861 in training data vs 2636 in official word list)

While the GPT generated texts do increase steadily with proficiency level, it seems to have overcomplicated the first 3 levels, while drastically underrepresenting the last three. I believe this is partially a case of Zipf’s Law at work. Since each of the mini-corpora were generated to include only 500 sentences each, and GPT is generating probabilistic random sentences, at the higher proficiency levels where the number of unique characters increases dramatically, they would be uncommon characters. So while the corpora do not accurately cover the official word lists, the comparison of generated texts against the character count in the training data clearly show that the fine-tuned graded language models do adjust to the proficiency level when, having been initially trained on sentences from all levels 1 through 6, they are then retrained for some additional epochs on the specific subcorpora.

These results accomplish the primary goal of this project, which was to generate text at various proficiency levels for language learners. In its current iteration, there are no sentence breaks, as the model is probabilistically generating a set number of characters based on a given primer word. One way to extend the model’s usefulness would be to intentionally incorporate words at a given level, in order to cover the full vocabulary, since words with low relative probabilities are much less likely to be generated. This could be done by rotating through the word list and using each word as the generative primer, though this comes with the obvious limitation of always placing target vocab at the beginning of the sentence, which is not always appropriate, depending on word type and usage. An alternative would be incorporating some sort of interventionary technique, when given a target word’s part of speech (POS) tag, the model can generate words based on a separate primer, and then by checking POS tags of the generated words, could insert the target vocabulary. One would need to take care that this is done in a way that the target word is still appropriate and felicitous in context, which could simply be checked against some threshold level of the probability distribution of that word given the current context.

Currently, there are two shortcomings in the model and project that affect its validity:

First, and most obvious, is that the text generation for a given level gate does not conform to the standardized word list for that level. This is simply a product of the training data. If I were building the training corpora again for this project, I would want to find a way to not only generate significantly more training sentences, but also explicitly filter the training sentences for appropriate vocabulary. For the lower three levels (HSK 1 through HSK 3), this could be relatively simple, using significantly larger sets of sample sentences generated for each level, then filtering off sentences which include vocabulary from outside that level. The problem with the training sets developed for the higher three levels (HSK 4 through HSK 6) was that the small sample size using probabilistic generation via chatGPT simply did not cover the breadth of characters and vocabulary words in the higher levels, especially HSK 6. Where the standardized word list has 5000 words, and 2636 unique characters, this model was only exposed to 861 unique characters across 3,000 sentences. Going back to Zipf's Law, in order to build a corpus that covers the whole vocabulary at each level, checks would have to be done in order to be sure that every word from the official word list appears in at least one generated sentence, ideally more. This could be brute-forced with a large enough generated data set and filtering, or more easily with targeted prompts triggering inclusion of missing words.

Second, currently there is no way to evaluate the relative grammar level of each of the level gated models. Not only was there no reasonable way to level gate the input of the training data, but I have no method to evaluate the accuracy and conformity of the generated output per level. It is my hope that these concepts are picked up during the fine tuning of each gated model. And while I do expect that the model learned some of the differences in sentence structure from the training data, I have no confidence in the training data having been generated with appropriately gated sentence structure and grammar. One could develop a level gated context free grammar of the language based on the official HSK list of grammar. Using this grammar to parse generated sentences as a validity checker would allow curation of truly level-appropriate text generation. This then comes with some of the same issues as with the word list above, namely frequency and volume. In order to guarantee coverage of the full range of grammar points at a given level, there would either have to be a large enough volume of training data that includes all of those constructs, or the model would need some interventionary method to trigger production of missing grammar structures.

For each of the above issues, I am concerned with interventionary methods, in that by definition, across a relatively small corpus, forcing coverage of grammar points and vocabulary items or characters could violate the natural distribution under Zipf's Law. Training a language model off of a corpus with strangely distributed grammar and vocabulary would obviously cause issues with the naturalness of the generated texts, for better or worse. The downside of this, being a somewhat unnatural distribution of generated text, may well be countered by the usefulness of the increased exposure to less common vocabulary and grammar from the perspective of the language learner.

In closing, I believe this project accomplished the primary goal on a conceptual level, being level gated text generation, even though the levels do not conform precisely to the intended official proficiency levels. Again, I believe this to be due primarily to the training data available for the model to learn from. However it is encouraging to see that this sort of level gated text generation is already mostly accomplishable via simple chatGPT prompts. I believe that with better prompt engineering, more accurately level gated text could be generated ad nauseam from the currently available large language models. Furthermore, I believe this is incredibly promising for the use of these sorts of tools in pedagogical pursuits. Here is an example text generated via currently available chatGPT (running GPT3.5), as a way to illustrate some of the pedagogical potential of these language models:

> Prompt: Pretend you speak only Mandarin Chinese at an HSK 1 level. Please write me a young adult themed short story using only HSK 1 level grammar and words from the following list: [official HSK 1 150 word list]
>
> Response:
>
> 我是一名学生，今天在学校。我有一个好朋友，他也在学校。我们在教室里学习汉语。老师很好，她教我们说话、写字和认识汉字。我们学了一些汉字和基本的语法规则。我们也学习了一些地理知识，知道北京在中国的上面。我们都很喜欢学习汉语，因为汉语很有意思。我们喜欢看汉语书，听汉语歌，也喜欢吃中国菜。在学校的时候，我们一起吃饭，一起上课。今天是星期五，明天我们不来学校，我们要在家里休息。我们希望天气好，可以去公园玩。我们喜欢在公园里玩，看花儿，看狗狗。我们都很高兴，因为我们是好朋友。

Despite several attempts to reengineer the prompt in order to filter and limit the generated output, I was unable to elicit accurately level gated text including only words from the HSK 1 vocabulary list, even when the list of specific words was specified in the prompt. This is currently one of many critical thinking shortcomings in the GPT3.5 model, though GPT4 is reported to have significantly better reasoning skills, and I am excited to see what sorts of output it is capable of generating.

Appendix A - Testing Hyperparameters:

Test 1 - 10k zhTenTen corpus
Training parameters:
- sequence_length = 10
- batch_size = 128
- number_epochs = 2
- learning_rate = 0.001

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 250
- number_layers = 2

Results:
- Loss: 5.437215335845948
- Runtime: 2.75min

Test 2 - 10k zhTenTen corpus
Training parameters:
- sequence_length = 10
- batch_size = 128
- number_epochs = 2
- learning_rate = 0.01

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 250
- number_layers = 2

Results:
- Loss: 6.019902499198913

Test 3 - 10k zhTenTen corpus
Training parameters:
- sequence_length = 10
- batch_size = 128
- number_epochs = 2
- learning_rate = 0.01

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 64
- number_layers = 2

Results:
- Loss: 6.160614488601684

Test 4 - 10k zhTenTen corpus
Training parameters:
- sequence_length = 20
- batch_size = 128
- number_epochs = 2
- learning_rate = 0.01

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 64
- number_layers = 2

Results:
- Loss: 5.390186637878418
- Runtime: 10min

Test 5 - 10k zhTenTen corpus
Training parameters:
- sequence_length = 5
- batch_size = 128
- number_epochs = 2
- learning_rate = 0.01

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 64
- number_layers = 2

Results:
- Loss: 5.460773259162903
- Runtime: 2min

Test 6 - 10k zhTenTen corpus
Training parameters:
- sequence_length = 10
- batch_size = 128
- number_epochs = 10
- learning_rate = 0.001

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 250
- number_layers = 2

Results:
- Loss: 4.637505566596984
- Runtime: 20min

Test 7 - 10k zhTenTen corpus
Training parameters:
- sequence_length = 5
- batch_size = 128
- number_epochs = 10
- learning_rate = 0.001

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 250
- number_layers = 2

Results:
- Loss: 4.744933343887329
- Runtime: 15min

Test 8 - 1M zhTenTen corpus
Training parameters:
- sequence_length = 5
- batch_size = 128
- number_epochs = 10
- learning_rate = 0.001

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 250
- number_layers = 2

Results:
- Loss: 4.614034634113311
- Runtime: 300min

Test 9 - 1M zhTenTen corpus
Training parameters:
- sequence_length = 10
- batch_size = 128
- number_epochs = 1
- learning_rate = 0.001

Model parameters:
- embedding_dimension = 200
- hidden_dimension = 250
- number_layers = 2

Results:
- Loss: 4.64064909362793
- Runtime: 184min

Test 10 - full HSK corpus
Training parameters:

- sequence_length = 10
- batch_size = 128
- number_epochs = 100
- learning_rate = 0.001

Model parameters:

- embedding_dimension = 200
- hidden_dimension = 250
- number_layers = 2

Results:

- Loss: 0.4045983849465847 - clearly the winner
- Runtime: 14min

Appendix B - Generated Text:

Sample dump of generated HSK 1 text:

我们可以去公园玩儿你也一样我喜欢吃水果请问洗手间在哪儿你喜欢喝茶还是喝咖啡你喜欢吃中餐还是西餐我们要去图书馆看书你怎么拼写我是学生你今天要做什么我的中文老师很好你的汉语说得很好我的手机很便宜你是老师吗这你有没有兄弟姐妹你喜欢吃蔬菜吗我叫李小明他的狗很可爱我们可以一起吃饭你喜欢喝什么茶我们要去邮局寄信我们今天晚上去唱歌他的电话号码是多少我的生日是七月二号我们一起吃饭吧你喜欢喝牛奶还叫五明这是哪我的生日是他是老师我有一十哥我们的学校在山上我家有两个狗我们要去游泳我想吃面条我想吃点儿水果我要去旅游不客气我不明白太便宜了我不喜欢吃蔬菜我要去银行取钱你的家人都在哪里我喜欢吃米饭我们可以在这儿坐下来吗我们一起去她你喜欢吃什么水果我爸爸是工程师我的生日是十二月十五号那是什么我的家在中国你想喝茶还是咖啡我的中文不好我的爱好是看电影我爱中国我要去上学他很喜欢读书我们今天去买东西吧我要去打篮球请问你家在哪儿今天是星期这个学校很好你喜欢运动吗我喜欢旅游我的电话号码是多少我想吃饭你的狗很可爱我们去看电影吧我很喜欢打篮球我的家在纽约我的手机丢了太贵了你有很多朋友吗我们可以一起去逛街我不会说英语我们要去散步我叫张三我的生日人我们可以去公园玩儿你也一样我喜欢吃水果请问洗手间在哪儿你喜欢喝茶还是喝咖啡你喜欢吃中餐还是西餐我们要去图书馆看书你怎么拼写我喜欢吃面条晚安我的家在上海我有一个好朋友对不起这是你的钱包吗我可以用一下你的的你有公吗这个电影怎么样这个衣服很贵他是我的老师我们一起去吃饭吧请问你多大了我的家乡很美他的爱好是打篮球我会说一点儿英语他们每天早上六点起床我想去看展览我的学校在城市中心我是美国人你喜欢吃哪种甜食他们有是你什么我喜欢读小说她很漂亮你的爱好是什么你喜欢吃什么我们去公园玩吧你去过哪些地方他有两个儿子我们要做作业她在哪儿再见这个房间很大我们去散步吧他有一个妹妹这是你的书包吗我不太会说英文她的生日是在九月你喜那我们学校里学习我来自中国请问这里有没有餐厅我们去旅游吧他们的爸爸妈妈很忙你会唱歌吗我的家乡有很多好吃的食物你能帮我学习中文吗我家有四口人我想买一件衬衫我喜欢听流行音乐你好你叫什么名字我的生日是八月十五一我们一起去看电影吧她的电话号码是多少我们的公司在城市中心我每天晚上十点睡觉我的朋友很多我们去唱歌我们去公园玩吧我们要坐公共汽车去学校我的电话号码是123456789你每天都做什么我是一名学生我在中国呆

Sample dump of generated HSK 3 text:

我的学校有很多俱乐部比如篮球俱乐部、音乐俱乐部等等这个包很实用可以放很多东西这个鞋子很漂亮但是有点贵我们可以一起去游泳今天天气很好我们可以去公园散步我们班上有很多同学他们都很聪明我的家乡有很多美丽的风景你身步你想不想去打篮球我已经叫了几个人了我们可以在这里买水果这个学期的课程很有趣她的狗很可爱总是跟着她走我们的公司有一个很好的福利制度员工可以享受很多福利我的朋友很多他们来自不同的国家她的电话号码是多少他身有一个很有才华的音乐家今天我打算在家里休息因为我感觉有点累你在哪儿工作现在是晚上七点钟我该做晚饭了这个菜很辣我们要去机场接他你的中文学得怎么样我的朋友今天晚上要来我家吃饭我的中文水平还需要提高他很高她身高他们在学习中文歌曲我喜欢吃水果尤其是苹果和橙子我们可以在家看电视也可以去外面逛街这个城市很美这个房间很大有一个双人床我们的学校有很多俱乐部比如篮球俱乐部、音乐俱乐部等等这个包很实用可以放很多东西这这个地方很有名这个城市非常繁忙他学习很努力我们可以一起去看展览我喜欢听古典音乐我想去旅游我们可以去逛街或者看电影他的中文水平很高我的家乡有很多山这个问题很容易你应该知道答案我的手机不见了我很着急他的身高人妈天都我们可以在饭店吃饭这个饭店的菜很好吃你应该试试看我喜欢吃炒饭我们在学校学习汉语我很喜欢旅游去过很多不同的地方昨天我去商店买了一些水果和蔬菜我们今天去逛街我的朋友在中国学习汉语这个电视太贵了我爱吃的玩身有很多风景名胜我的家人很支持我让我感到很温暖我们需要一些水果和蔬菜你想喝茶还是咖啡今天天气很好我打算去公园散步你想不想去打篮球我已经叫了几个人了我们可以在这个咖啡馆见面我在公园里散步这个地方很漂亮是玩身你这个地方很适合散步和跑步我们在学习写汉字我们班有三十个学生我的朋友给我推荐了一个很好的电影我准备去看看你有空的时候可以来我家玩你昨天晚上看了什么电影我的同学很聪明她总是能解决难题我们可以在餐厅里那高身体育这个电视太贵了我爱吃中国菜我们可以在餐厅吃饭也可以在家做饭我们在学习汉语词汇这个电影很感人我看了好几遍这个电视节目很有趣这个地方很热闹他喜欢听流行音乐和摇滚乐我们可以在这里买水果这个学期的课程一身友名胜他是公我们班有三十个学生其中有十五个是女生这个电脑很快我每天早上六点钟起床然后去跑步我们可以在电影院看电影很有趣我最喜欢的颜色是蓝色因为它很舒服我们在超市买东西你可以帮我找一下吗我想学会游泳这

Sample dump of generated HSK 4 text:

我们可以通过网络购物来节省时间和精力这个问题非常复杂需要认真考虑这道菜的味道很好我很喜欢我最近在学习如何做菜我喜欢唱歌跳舞我们需要多锻炼身体我们可以在公园里打篮球我们可以一起去逛街购物我们的公司每年都会你不对了我们需要把这个问题解决掉不能拖延他的作品得到了许多人的认可她正在做饭你可以等一下我们可以一起看电影我想买一双新鞋子我们可以在这里享受美食我们应该努力改善自己的生活质量我的中文课程非常有趣老师很好他味道我们应该尽量减少使用化妆品这个问题很简单你应该知道怎么做我们可以坐出租车去我们应该鼓励孩子们多读书我们应该遵守交通规则这样才能保证安全我们应该尽力保护环境减少污染我的生日快到了我想请你们一起庆祝他她道不远我们需要一个更好的沟通方式你喜欢听流行音乐还是古典音乐我们明天一起去爬山怎么样我们在图书馆里看书安静得很我们必须尽快采取措施避免那多的损失我们应该关注年轻人的教育问题我的家离学校不远我们需要一个这个问题需要仔细考虑我们需要购买一些新的技能我们可以在家里看电视我们应该尽量避免使用一次性餐具这是我第一次来中国感觉非常新奇你可以告诉我怎么做吗我们可以在这个公园里散步我们可以去看一场音乐会感受音乐的魅人不懂他是一个很有经验的工程师她的演讲引起了观众的强烈反响我们可以一起去看演唱会这个地方的交通非常方便很容易到达我的爱好是旅游和尝试不同的美食我们在上海边了天语我们需要健康饮食这样才能保持身体健康我们公的道菜非常忙我们在餐厅里用餐我们可以去游乐场玩度过一个愉快的周末我们需要认真学习汉语我喜欢看电影特别是科幻片我们可以在网上买书我们可以去买一些纪念品我的朋友们很友好总是愿意帮助我我们每天早上都要起床很早是一个电视比那个电视贵汉语我们可以在图书馆学习我喜欢喝绿茶我昨天晚上去了一家新的餐厅你喜欢什么样的音乐她的英语说得非常好几乎没有口音我们可以坐地铁去博物馆我们需要更好地管理我们的时间提高工作效率我们可以那道不能随文化我的朋友很擅长画画每次都可以画出美丽的作品我们需要一些更好的想法我的家乡在山东你喜欢看电视剧还是电影我们可以在这里停留一会儿我们可以在健身房里锻炼身体我们需要一些额外的帮助我的家乡有很多名一么个电影有很多美食他是一位很有耐心的人他的建议非常有帮助这个电影非常好看这个问题很复杂我们需要仔细考虑一下我昨天晚上看了一部很好看的电视剧我的手机没电了我们应该尽量避免浪费食物我们应该尽可能地保护环境

Sample dump of generated HSK 5 text:

我们公司的业务范围正在逐渐扩大她很有耐心总是能够解决问题他的团队合作能力非常强能够带领团队完成任务他非常努力工作很快就升职了他的音乐风格非常独特这个会议的主要议题是如何提高效率他一直在关注市场动态为公司你保成功地进入市场他的经历让他成为了一个非常有见识的人我们要始终保持谦虚谨慎的态度不断提高自己的能力他们的文化背景和生活习惯与我们不同但我们应该尊重彼此在这个时代数字化和信息化已经成为了主流趋势我们需要他的培养非常生动让人印象深刻我们需要认真考虑每一个决策和选择的影响这个工程需要我们充分发挥团队合作精神才能取得成功这个计划的实施需要大量的资金支持我们要积极参与公益活动为社会贡献力量这个项目需要我们更好她的育利这个问题需要我们认真思考和解决在中国的传统文化中礼仪是非常重要的我们应该及时调整和优化我们的工作计划在这个项目中我们需要紧密合作共同完成我们应该秉持诚信、公正、透明的原则他们的团队合作能力很强完这个公司的技术非常漂亮值得一去她的成就为年轻女性树立了榜样我们需要在市场上建立更好的口碑我们需要积极应对市场变化这个项目需要一个详细的市场调研我们希望通过合理的规划使城市的交通更加便利这个方案需要一个可人才能达到你的目标这个企业的发展潜力非常巨大她是一个非常有魅力的人我们应该积极思考和解决问题而不是抱怨和埋怨我们需要建立更加有效的沟通机制来提高工作效率我们需要提高员工的技能和素质这场比赛的结果令人失望的可持续产品的市场需求非常旺盛他的学术研究受到了广泛的认可我们要关注环境保护我们需要提高产品的质量和品牌形象这个国家的经济正在快速发展我们需要更多的创业精神来推动经济发展这个计划需要我们投入更多的时间和是支利这个企业的发展潜力非常巨大她是一个非常有魅力的人我们应该积极思考和解决问题而不是抱怨和埋怨我们需要建立更加有效的沟通渠道来避免误解我们应该注重道德和人文素质的培养现在的年轻人非常注重工作与生活的平那地利进说这个项目需要我们集体合作才能完成我们需要更多的人手来完成这项任务这个计划需要一个强大的执行团队在生活中我们需要学会平衡工作和家庭在这个聚会上我认识了很多新朋友我们公司的业务范围正在逐渐扩大她很一外完成了一个重要的项目我们应该坚持自己的原则不要妥协这个公司的发展潜力非常大有着广阔的市场前景我们需要更加注重人才培养和引进在工作中我们需要注重细节我们需要对这个方案进行全面的评估她的音乐很感人这个问

Sample dump of generated HSK 6 text:
我们的环境和污染这个项目有启我们需要充分发挥团队合作的精神她的表演深深地感动了我这个项目需要投入大量的资金和人力支持才能取得成功她是一位非常有艺术天赋和表演才能的人能够打动观众的心中国古代的文化底蕴非常你人深深地感到敬佩我们需要探讨这个问题的实质这个城市的交通状况非常糟糕需要改善这个企业的管理团队非常有能力业绩一直保持增长他的研究成果得到了学术界的高度评价通过多种渠道扩大知识面可以帮助我们更好地适应社他领导才能取得重大突破我们需要更多的信息和有才能取得好的成果他的才华和勤奋使得他成为了行业内的佼佼者那个地区的经济正在迅速发展吸引了大量的游客你的身材确实很匀称真让人羡慕我们需要为客户提供更加个性化的服她非常得体了大家的认可我们需要加强科学技术创新推动社会进步和发展我们应该更加注重人民群众的利益她的专业技能非常精湛在全球化的时代跨文化沟通和合作变得越来越重要这个问题很复杂需要我们深入研究学习一门外语对这个明确的时间表他在商业领域有着广泛的人脉能够为公司带来更多的商机这个国家的政策对于外资非常开放在这个领域我们需要不断学习和提高自己是我们每个人的责任生命中最重要的是什么对我来说是健康和家人青少年应该养人扩扩大业务这个问题的答案很复杂需要仔细研究才能得出结论景区内的游客络绎不绝景色非常优美我们需要尽快解决这个问题否则会影响项目进度他是一位谦虚和有责任心的人总是能够为他人着想和帮助别人我们需要对市场进行的挑战这个问题需要我们从多个角度来思考和解决在国际贸易中了解市场和客户需求非常重要才能更好地开拓市场我们要尽可能地减少浪费对于这个项目我们需要先制定一个详细的计划再开始实施我对中国的文化和历史非常感兴趣是有挑战统的共同努力和汗水这篇文章非常深刻引人深思让我有了新的领悟那个公司在国内外市场都有很大的影响力是行业内的领导者我们应该注重身体健康保持合理的饮食和运动习惯这个问题的解决需要我们共同努力才能取得最那有了大家的信任这个项目需要我们做好充分的准备和规划我们应该学会如何处理突发事件不要惊慌失措我们应该尽可能减少我们的碳排放以应对气候变化在市场竞争日益激烈的情况下品牌形象非常重要这个项目需要大量的投资和一些有重要的启示意义在这个国家人们对环境保护越来越重视在工作中我们要善于合作才能取得更好的成果我们应该珍惜时间努力提高自己的能力要想成功必须具备良好的领导能力我们要学会保护自己的隐私不要轻易地透露个人信

Sample dump of generated HSK 2 text:
我很远天不很好我们可以在图书馆学习或者看书你喜欢喝咖啡还是茶我们去年去了北京旅游对我有一个弟弟我们可以去买东西我是一个学生他们住在一个大城市里这个城市的交通很方便有地铁和公交车他是我最好的朋友他们每周都你喜欢吃中国菜我想喝一杯茶我喜欢吃面条这是一本书我喜欢听音乐你呢我们在哪里旅游了我喜欢吃米饭和面条我们可以一起去购物我的朋友们喜欢看电影他的汉语发音很好听起来很标准我的同学们都很友好我们经常一起玩你可以他很高很有趣我的天气很热这个问题很重要她喜欢唱歌你们公司有多少人他是一个很好的老师我们可以去那家饭店吃晚饭她每天都喝茶我的父母都很健康我的生日是十月十号他的生日是明天我们可以在那个音乐厅听音乐会他是一名她城城市的夜景很美你要不要来看看他喜欢旅游昨天晚上我很或快书他喜欢看电视剧你呢我的妈妈会做很多好吃的菜我想买一张火车票去北京对不起我今天没有空我会说一点儿这个地方很安静我们可以去购物中心买东西你想吃什么这张我的手机很新他是我的同学他们每天早上都去跑步他们一起去看电影了这个花园很漂亮我们可以在这里喝一杯咖啡我的家离学校很近我们一起去餐馆吃饭吧我们可以在那个体育馆健身你喜欢吃中国菜吗这是一个好主意我们要在人馆很大有很多人口我喜欢喝茶这个问题很容易这个电影很有趣你想喝什么饮料我们可以去图书馆看书我的姐姐在学习英语我们可以在那家书店买书我们可以在那个电影院看电影这个房间很干净他的电脑坏了你喜欢喝咖啡吗我们可的亮你要不要来看看你有没有听过这首歌我们要在餐厅吃晚饭我喜欢蓝色他是我的老师我们可以在公园里钓鱼我们明天去购物中心我们明天去旅游我们可以一起去旅行他的家离学校很远不会我只会唱歌我的弟弟很聪明你是中国人吗是了你要不要试试你喜欢吃中国菜我想喝一杯茶我喜欢吃面条这个电影很有趣你想喝什么饮料我们可以一起去钓鱼我们去学校吧她每天都锻炼身体他们今天去哪里旅游了我喜欢吃米饭和面条我们可以一起去玩游戏他的中文不太好但那我们可以在这里练习英语他是我的老师他教我们语文我们明天去旅游我们可以一起去游泳这个书包很漂亮这个城市很大今天天气很好我们去公园吧她的生日是八月二十一日我的朋友们都很有趣我们一起玩得很开心我们可以在公园一了你喜欢喝咖啡吗我们可以一起去逛街买衣服我们在公园里散步我的朋友是医生我的手机丢了很不方便我们可以一起去看展览我们可以在这里打电话你喜欢吃米饭吗他喜欢看电视我们今天晚上去看电影我想去北京旅游她的家人都

Note:
Prompt used to generate level gated sentences with chatGPT:

"You speak Mandarin Chinese at an hsk(#) level. Please generate 100 unique sentences, using hsk(#) grammar and vocabulary, with no pinyin or translation"

Changed prompt to:

"You speak Mandarin Chinese at an HSK (n) level. Please generate 100 unique sentences, where each includes HSK (n) vocabulary and grammar, with no pinyin or translation"

Because it still seemed to be generating many low-level sentences (not that HSK 1 sentences wouldn't be comprehensible to HSK 4 speakers, but they won't necessarily improve my model). The new prompt seemed to generate more complex sentences with more varied vocabulary.

Which tended to give me a network error around 30 to 50 sentences, so I just re-ran the prompt until I had built up a larger selection of sentences, ran a filter to remove duplicates, and called it a day.

References

- Xue, Nianwen, et al. Chinese Treebank 8.0 LDC2013T21. Web Download. Philadelphia: Linguistic Data Consortium, 2013.
- McEnery & Xiao, The Lancaster Corpus of Mandarin Chinese: A Corpus for Monolingual and Contrastive Language Study (LREC 2004)
- Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., & Suchomel, V. (2013, July). The TenTen corpus family. In 7th International Corpus Linguistics Conference CL (pp. 125-127).
- Singh, P. Text Generation via RNN and LSTMs - PyTorch. Kaggle. July, 2020. Retrieved from: https://www.kaggle.com/code/purvasingh/text-generation-via-rnn-and-lstms-pytorch/
- Wang, B. Cross Entropy Loss in PyTorch. Medium. January, 2021. Retrieved from: https://medium.com/swlh/cross-entropy-loss-in-pytorch-c010faf97bab