A Transformer-Based Text Prediction Model for T9 Input Method

DONG CHENGZHI(690-37-4400)

1.Introduction

In this study, we address a sequence-to-sequence task within the context of the T9 input method, which can be regarded as a lossy encoding scheme for the English alphabet. We propose a character-level model based on the Transformer architecture, designed to predict the most probable sentence corresponding to a given input digit sequence. The predicted sentence can serve directly as the intended output of the input method. The model was trained on the 1M-scale benchmark dataset. Experimental results demonstrate the effectiveness of our approach: the model is able to accurately predict the vast majority of words within a sentence. However, due to challenges such as ambiguity in short sentences, the sentence-level accuracy remains suboptimal.

The code for this work is available on GitHub: github.com/Mutsuki0024/T9_Keyboard

2.Method

2.1 Problem Formulation

This study addresses the task of predicting the most probable character sequence given a sequence of numeric key presses in the context of a T9 input method. The T9 (Text on 9 keys) input method was commonly used before the widespread adoption of smartphones, during the era of feature phones. Due to the physical constraints of the devices, the main input interface typically consisted of only nine numeric keys (from 1 to 9). Among these, the key '1' was usually reserved for punctuation marks and symbols, while the keys '2' through '9' were mapped to the 26 letters of the English alphabet. For example, the digit '2' corresponds to the letters 'a', 'b', and 'c'. Thus, when a user inputs a letter, they are in fact pressing the numeric key associated with that letter.

This input scheme inherently introduces a form of lossy encoding, where multiple distinct words may correspond to the same numeric sequence. Consequently, the model must be capable of capturing contextual dependencies to accurately infer the most likely intended word or phrase. For instance, given the input sequence "43556 96753", the model is expected to correctly predict the intended output "hello world".

This task can be formalized as a classification problem. Once a numeric sequence is specified, the goal is to perform multi-class classification over all possible candidate words or sequences that could be mapped from the input, selecting the most contextually appropriate one. the given digit sequence $D = [d1, d2 \dots dn]$, the learning model M(·), and the output string S = [c1,c2 … cn], the process can be described as

$$S = M(\mathrm{D}) \tag{1}$$

2.2 Method

The overall architecture of the model is illustrated in Figure 1. This work adopts a character-level vocabulary. For the input digit sequence, each digit (including the space character) is mapped to a unique numerical ID ranging from 0 to 9. Similarly, for the output string, each character is also mapped to a unique numerical ID ranging from 0 to 27, with the ID 0 reserved for the padding character in both cases.

After the digit sequence is mapped to its corresponding ID sequence, it is passed through an embedding layer to obtain a sequence of character-level embedding vectors. Each embedding vector represents a token and is fed into a standard Transformer Encoder to model contextual and

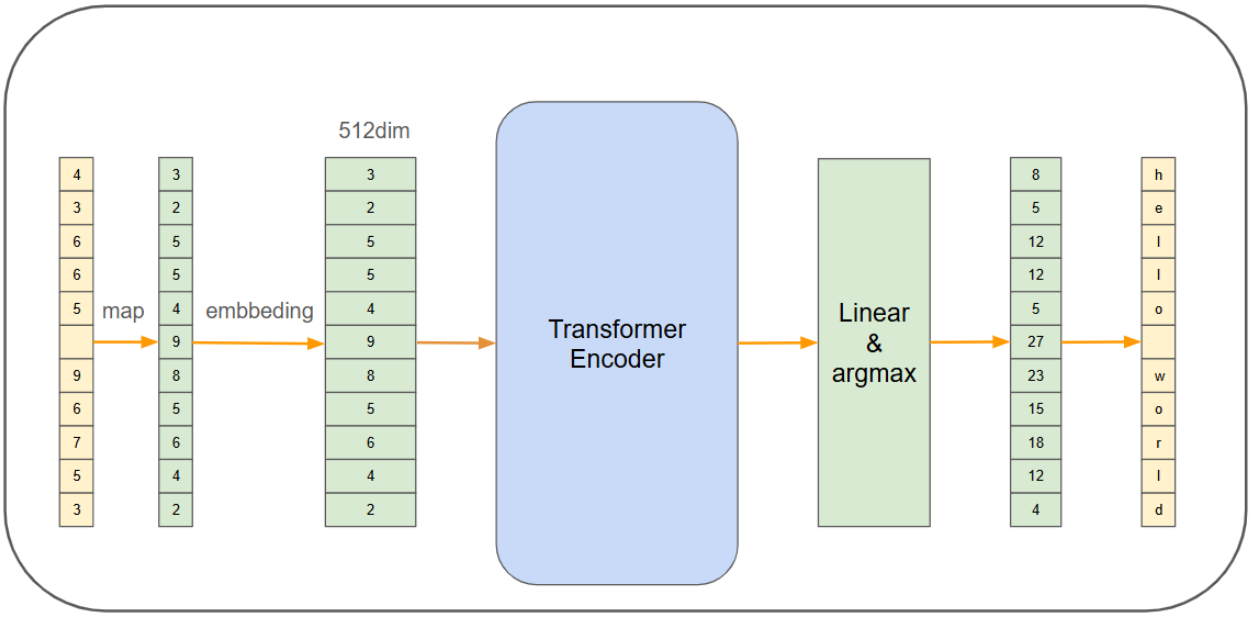

Figure 1

attention-related information. The output of the Transformer Encoder is then passed through a linear layer to produce, for each token, a predicted distribution over the character vocabulary. The most probable character ID for eac h position is selected using the argmax function, and the resulting sequence of IDs is mapped back to characters via the vocabulary, yielding the final predicted English sentence.

3.Experiment&Evaluation

3.1 Experiment Setup

The dataset used in this study is based on the 1M-scale benchmark. While maintaining general applicability, several preprocessing steps were applied to the original corpus to reduce the problem scale due to hardware limitations. Specifically: (1) the corpus was segmented into clauses using commas as delimiters; (2) all uppercase characters were converted to lowercase; (3) all punctuation marks were removed; (4) clauses longer than 25 characters or containing any word longer than 20 characters were excluded; and (5) clauses containing any numerical digits were also excluded from the training set.

Model training was conducted using a single NVIDIA GTX 4060 GPU. The model architecture was configured with an embedding dimension of 256 and a Transformer encoder consisting of 4 stacked layers with 8 attention heads. The maximum position encoding length was set to 512. The batch size was set to 64. The model parameters were randomly initialized and trained for 2 epochs, corresponding to approximately one million training steps. The training process employed the AdamW optimizer and used the cross-entropy loss function.

3.2 Result

During training, the model was evaluated on the validation set every 2,500 steps. Evaluation metrics included character-level accuracy (correct characters / total characters), word-level accuracy (correct words / total words), and sentence-level accuracy (fully correct sentences / total sentences).

Character-level, word-level, sentence-level accuracies and the trend of the running loss during training is shown in Appendix. Upon completion of training, the model achieved approximately 98.6%, 95.4%, and 70.0% accuracy on the validation set at the character, word, and sentence levels, respectively. These results indicate that, given an input sequence of digits, the model is capable of accurately generating the corresponding input method predictions and correctly predicting the vast majority of words, even under the presence of numerous proper nouns and semantically ambiguous terms in the corpus. The relatively lower sentence-level accuracy can be attributed primarily to sentence-level ambiguity (where a single digit sequence may correspond to multiple grammatically valid sentences) and the high proportion of very short sentences (of length three or less) in the dataset. Since shorter sentences tend to exhibit greater ambiguity, this limitation does not

necessarily undermine the model's overall predictive performance.

3.Conclusion

In summary, we developed a Transformer-based sequence-to-sequence model tailored for the T9 input method, which achieved promising accuracy in the sentence prediction task. Potential directions for future improvement include replacing the character-level modeling with word-level or subword-level modeling to better capture semantic information. Additionally, incorporating part-of-speech tags for each word and combining them with the word tokens to form mixed semantic units could further enhance the model's understanding of linguistic structure while mitigating ambiguity.

4.Appendix