Bidirectional Attention for Offline CJK Character Recognition

A (Pre)Preprint

Tom Fuller

January 16, 2021

1 Introduction

Currently, character classification for Chinese, Japanese, and Korean (CJK) languages has been very limited. In real-time classification, results have since been improved from prior SVM implementations with convolutional neural networks (CNN). Although referred to here as ”CJK” languages/characters the term is used as a collective for the logographic characters used in Hong Kong, Taiwan, and Vietnam in addition to the aforementioned countries. Prior research has generally treated each Chinese character as a whole, ignoring the internal two-dimensional structure within each character. The most recent state-of-the-art work has started to incorporate structure, but only achieves 40.82% accuracy on unseen handwritten characters, despite 96.66% accuracy in recall [23]. For printed Chinese characters, methods using metadata (multi-typed attributes) pushes the zero-shot performance to 56.6% [5], but is unable to translate results to handwriting samples.

Online recognition has more success, and is popular in mobile computing devices [7], seeing heavy use in input method editors (IME). However, their reliance on temporal data makes them ineffective in OCR, and causes failures for incorrect stroke order.

Despite the over 40,000 characters needed for complete modern day coverage in present day Chinese, existing research has been reliant on the 3,755 characters of GB2312-80’s level-1 set and the CASIA handwriting dataset [12]. These characters form the ”base set” needed for general literacy of Chinese. For Japanese, a set of 2,136 characters for basic literacy, the *jōyō kanji*, is published by the Japanese Ministry of Education.

In other computer vision classification tasks, recognition across over 9,000 classes has been achieved, in the YOLOv2 architecture [15]. It is unsurprising that the handwriting recognition approaches in similar architectures all boast recognition rates of slightly over 96% for the 3,755 class size. The top three approaches are based on GoogLeNet [26], ResNet [5], and DenseNet [23].

The poor performance on characters landing outside of the level-1 metric is a problem that has not yet been solved. The current state-of-the-art method in both zero-shot and recall, DenseRAN [23], utilizes attention, but can fail easily if the wrong path in attention is taken. Thus, the intuition of this proposal is that a bidirectional and hierarchical attention mechanism is necessary within the Chinese, and by extension, CJK character classification task. Bidirectional graph traversal allows for an efficient and correct encoding sequence of characters to be built. This intuition serves as the basis of the structure necessary to achieve performance in the zero-shot case, as well as recognition across the over 40,000 classes of Chinese logographic characters of modern use.

2 CJK Character Structure

While it has been stated that filters of smaller size are not suitable to recognize the complex relationships between Chinese strokes in convolutional architectures [16], this is overlooking fundamental hierarchies and geometric properties of logographic drawings.

Many approaches have focused on a specific language or script, for example Chinese [11], Japanese [6], or Korean [8]. However, the structure can be generalized across CJK characters. In order to classify unseen characters, or a large number of characters, the intuition is that an approach leveraging these similarities be taken.

2.1 Phono-semantic Compound Structure of Chinese characters

Similar to the tree structure created by hyponyms and hypernyms of language which are captured within WordNet [14], there exists a phono-semantic compound structure within many individual Chinese characters. Recent research on native Chinese speakers has shown the Stroop effect to occur, delaying recognition if misrepresented [24], indicating that the character recognition pathway within humans relies on these compounds. By decomposing compound ideographs into their constituent radicals, a graph can be created similar to the morphological decomposition and semantic field relationships within English. As over 80% of the Chinese characters have been found to have this relationship [18]. The intuition is that this structure is critical to obtaining fast character recall, creating a meaningful attention mechanism, and increasing performance in zero-shot classification.

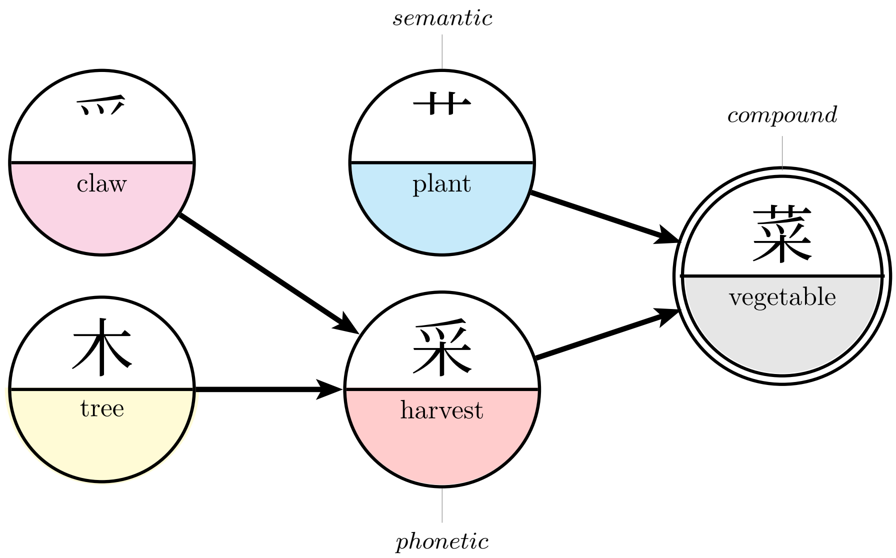

Figure 1: The phono-semantic relationship within the compound character, 菜 or *vegetable*. The semantic component is a linguistic *determinative*, providing the compound with elemental context. The phonetic component is semantically relevant here. Combined as *”harvested plant”* for the character meaning *vegetable*.

2.2 Kangxi Radical Position Encodings

Kangxi radicals are the 214 radicals/roots comprising the compound characters of Chinese characters in CJK languages. As there are much fewer roots than possible constituents, positional encoding serves as a meta-feature necessary for classification beyond any base set.

In the Japanese encoding system, constituents are split into seven groups. These are *top*, *bottom*, *left*, *right*, *kamae* (かまえ), *tare* (たれ), and *nyou* (にょう). The last three do not have equivalent words in English, but roughly translate to ”enclosure”, ”hang down”, and ”wrap around bottom” respectively. The remaining case is for radicals themselves, which do not have a position. Totaling to a feature vector of size 8.

It is worth noting that the *kamae* group represented by ▣ in Figure 2 is actually a grouping of positions that enclose another, but are less common than *tare* or *nyou*, and do not have a specific group. For example, 医 consists of *kamae* radical 匸 enclosing 矢. The radical 匸 does not have an explicit stroke closing right side, although it can be thought of as being closed by deliberately missing stroke, differentiating it from 囗. Similarly, the radical 凵 is enclosed by a missing stroke on the top.

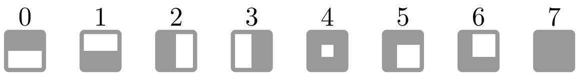

Figure 2: The different positions of radicals within a CJK character. The encoded position uses 3 only bits. From left to right: *top*, *bottom*, *left*, *right*, *kamae*, *tare*, *nyou*, *radical*

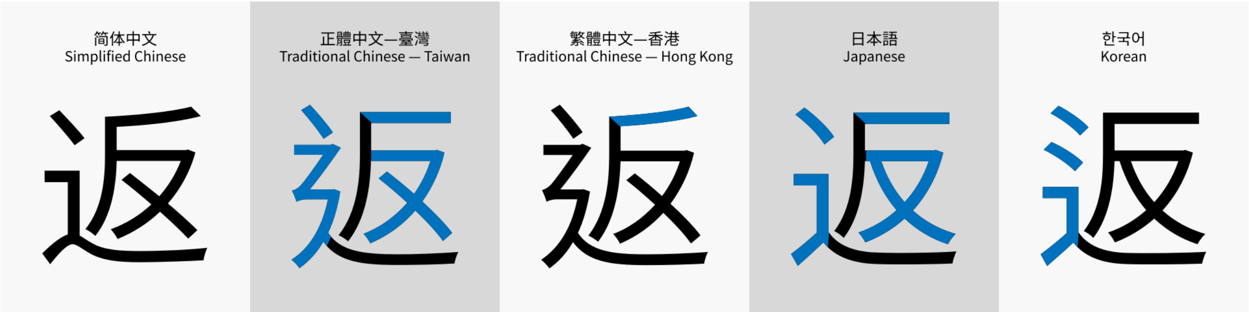

Figure 3: Regional differences within the Unicode character (U+8FD4). Image in figure from Wikipedia [20].

2.3 Han Unification & Unicode Ideographic Description Sequences

To simplify similar character representations across Chinese (*hanzi*), Japanese (*kanji*), and Korean (*hanja*) in the CJK languages, Unicode compresses similar representations into a single set of unified characters. This ”Han unification” combines glyph variants by grapheme into a single code point within Unicode. The traditional Chinese glyph for ”grass” uses four strokes in the radical (⺾), while simplified Chinese, Japanese, and Korean have three (⺿). The ⺾ and ⺿ characters both map to (U+8349, 草) and exist as variants. Due to the use as a radical, the four and three stroke variants contain separate encodings in the Unicode/CJK Radicals Supplement, allowing them to be rendered here. Figure 3 illustrates the effect on constituent radicals. The term *ideograph* is used when referring to the set of glyph variants representing a common idea.

Ideographic Description Sequences (IDS) in the Unicode standard [1] allows for any character to be encoded using logographic composition that is common across CJK characters. A representation sequence is shown in Figure 5. Enocding sequences are not unique, and have a many-to-one relationship. The large quantity of strokes in the character *biáng* in Figure 4 illustrates that despite variations in constituents, a core set of ideographs determines the compound character meaning.

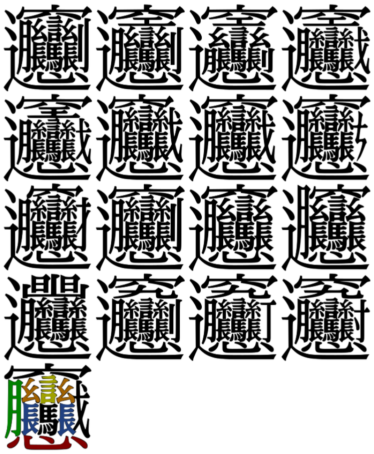

Figure 4: The 17 variants of the character *biáng*, a type of noodle. The last variant has constituents common across all variants colored to illustrate the core ideographs.
These core constituents illustrate the intuition behind a bidirectional, multi-headed attention model.

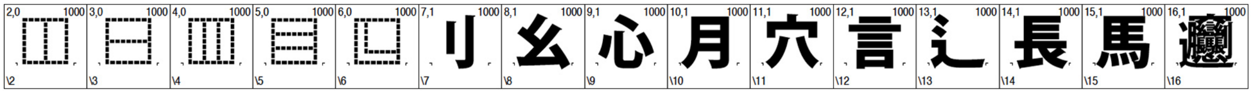

Figure 5: An IDS of length 15 encoding the rendering of *biáng* in OpenType. This sequence is just one capable of rendering *biáng*, prior to it's inclusion in 2020 [1]. Image in figure from Adobe [13].

3 Related Research

3.1 End-to-end Learning in Recurrent Attention Networks in Object Classification

Within Chinese text recognition, end-to-end [19] and bidirectional [25] approaches have been used, but only as far as sentence level character segmentation. However recurrent attention has found use in image classification tasks [22], greatly reducing the number of connections and reducing error rates by comparison to ResNet. This has found it's way into DenseRAN [23] at the character level, but does not consider the nested hierarchical nature of characters within the compounds.

3.2 BERT - Deep Bidirectional Transformers

Natural language processing has reached new heights with the novel approach of pre-trained bidirectional transformers in BERT [4]. While the concept of bi-directional attention is applied to the sentence level, I think the classic internet meme can illustrate the idea:

> *Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.*

Although the statement itself is not exactly true [3], typoglycemic text demonstrates that the human brain is not actually processing the entirety of a word when reading quickly. Since the "shape" of the word is important, when parsing the word between two space delimiters, the brain notes key features at the first and last position, and works it's way inwards until it arrives at what it has considerable confidence is the word.

The application of pre-training transformers in BERT has been demonstrated in sentence-level tasks, and fine tuning at the token level. This has been done using a "masked language model" (MLM) as the pre-training objective for the encoder. By answering the question of "Which characters need to appear before the others" in a string of text, the ability to preserve semantic recall despite fluctuations in character sequence is made possible. Within NLP translation problems, this is the segmentation and reordering of the constituent concepts within a sentence.

When the decoder is then presented the encoded sequence, it is able to decode the concept sequences back to a readable format in the desired translation language.

4 Proposed Architecture

The problem of CJK character classification can be simplified by utilizing the structure within the characters. In order to tackle the handwritten recognition task effectively, we must simplify the problem. The approach outlined here is designed with goals of future extensibility through transfer learning. Each section describes a module that can be individually trained to achieve the end goal of offline CJK character recognition.

4.1 Observe - Convolutional Handwriting Classifier

The input must break the handwritten character into the constituent radicals that compose it. This task is similar to the object detection and classification problem. When performing detection in regions within a handwritten CJK character, the end goal is to group strokes belonging to each component within a compound character at the current depth. This can be performed on multiple feature grids by the multi-headed attention, providing resolution granularity during constituent observation. For an input, the network will return a feature vector of up to n constituents, $\vec{c}_n =< \vec{O}_0, \vec{O}_1, ...\vec{O}_{n-1} >$ at the current depth. The tuple

$\vec{O}_n = (p_n, r_n)$ is the observed constituent at the current level, containing a position and radical encoding respectively.

The radical result, r_n will be marked by **?** if the constituent is unrecognized, signaling to the Transformer that additional depth is needed in a particular region to achieve tokenization by the Encoder.

4.2 Encode - Attention Based Character Tokenization

The second network structure is the first half of a Transformer, prevalent in linguistic models of natural language processing [21]. The Encoder uses the Observer network as a means of classification to tokenize a character into a sequence of embeddings. The embedding sequence leads to higher order classification at the Unicode character level, additionally driving the encoder-decoder attention.

Following the bidirectional attention mechanism of the Transformers as seen in BERT [4], attention can be given areas for use by the Observer network. While the linguistic case is unidimensional over a series in text, the lexicographic case mandates two dimensions to span the character space as shown in Figure 7 and 6.

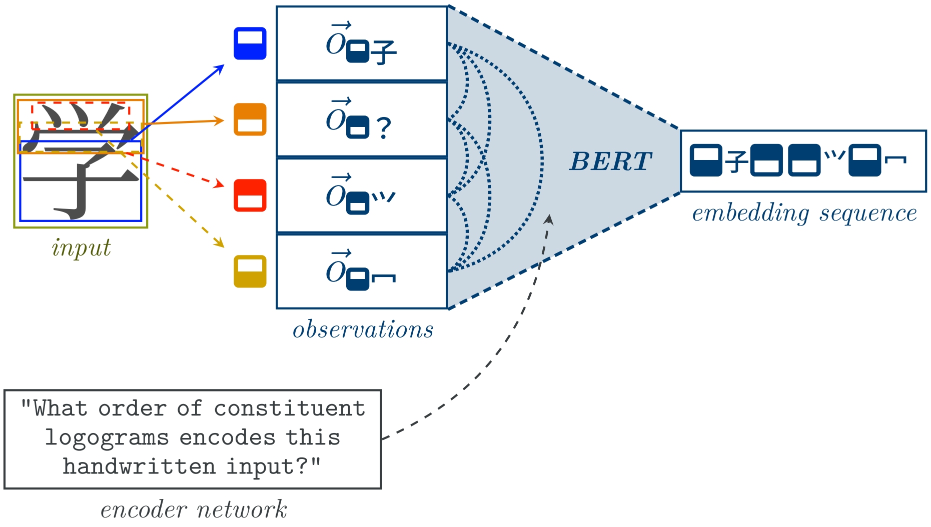

Figure 6: The interactions of the Encoder network. The encoder uses it's attention to query the Observer for observations in a given region. The observations are used to create an embedding sequence for the Decoder. The attention mechanism is then driven in jointly by both the encoder and the decoder's processing of the token sequence.

The Encode network needs only to provide enough focus to obtain a sequence of encoded inputs capable of returning the correct Unicode character by the Decoder. This in practice makes execution time much faster than the $O(n^2)$ theoretical bound of a Transformer. Time is saved by avoiding extraneous and potentially misleading classifications, which can be seen in Figure 4. Processing only core constituents solves the misalignment between focus order and stroke order. This also improves performance in characters with variants by resolving before attention is spent on the differing features.

4.3 Decode - Unicode Character Retrieval from Embedding Sequence

The final portion of the network will process two encoded sequences in parallel, as part of the bidirectional approach seen in BERT [4]. By creating two regions of focus, the result can be reached faster, and potentially without using all the information. The main result from the decoder is in providing a single UTF-16 character (represented across a weighted probability vector of all Unicode CJK characters).

Since the decode network takes only a tokenized embedding sequence, the decoder can be trained quickly with ground truth data outside of the computer vision components. Fine tuning of Transformers in subsequent training sessions has been quite effective [21] [4]. By training the fully connected output layers from the decoder, future questions can be answered, making this architecture extensible.

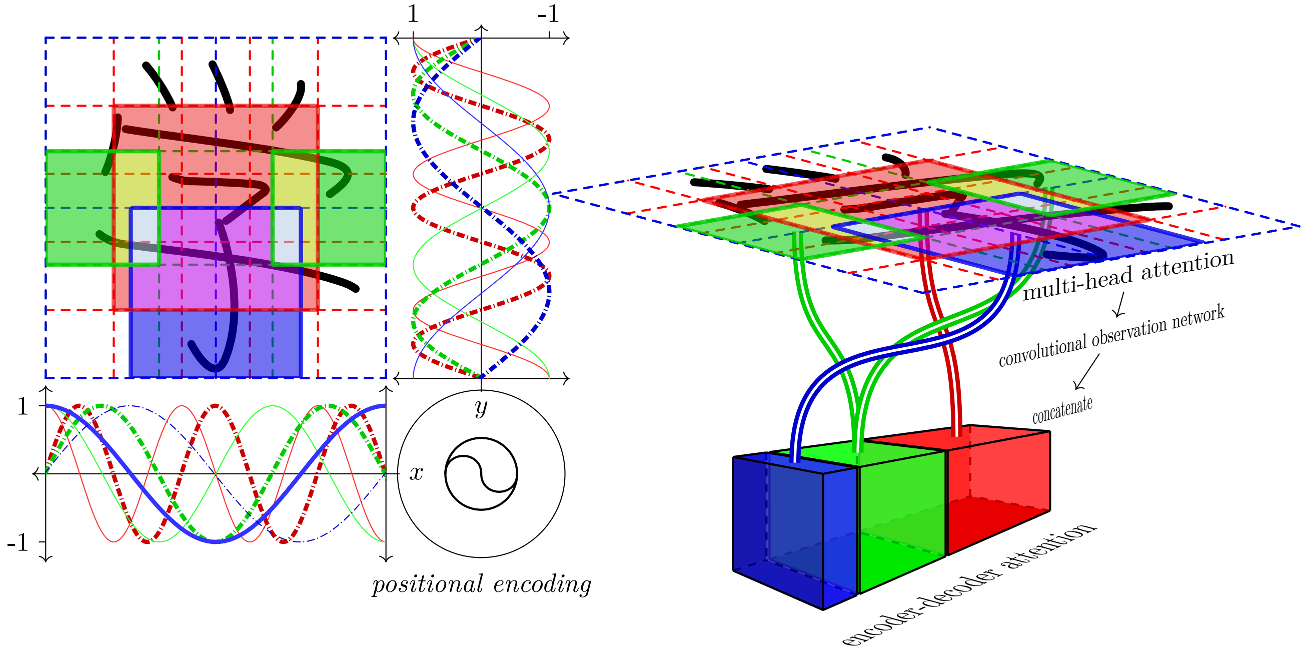

Figure 7: The positional encoding mechanism of the Transformers. On the left, input sinusoids are shown. The selected waves in bold are chosen arbitrarily from each sine and cosine pair. The waves of highest energy for a given input is selected to control the area of attention. On the right, the same attention selection for the three heads is shown being passed through the convolutional observer and concatenated. The concatenated result is then used by the Encoder and Decoder to determine future positions of attention.

4.4 Summary

The Observer network will learn

- ”Are any CJK constituents located in this handwriting sample? Where?”
- ”Is the constituent a known radical? should it be broken further?”

The Encode network will learn

- ”What does the position of one ideograph mean in relation to others?”
- ”What region should get attention so a tokenized sequence can be given to the decoder?”

The Decode network will learn

- ”What Unicode CJK character does this sequence of radicals and positions represent?”
- ”After what has been seen, what region should get attention next?”

The operational flow of the entire network is shown in Figure 8.

5 Overview

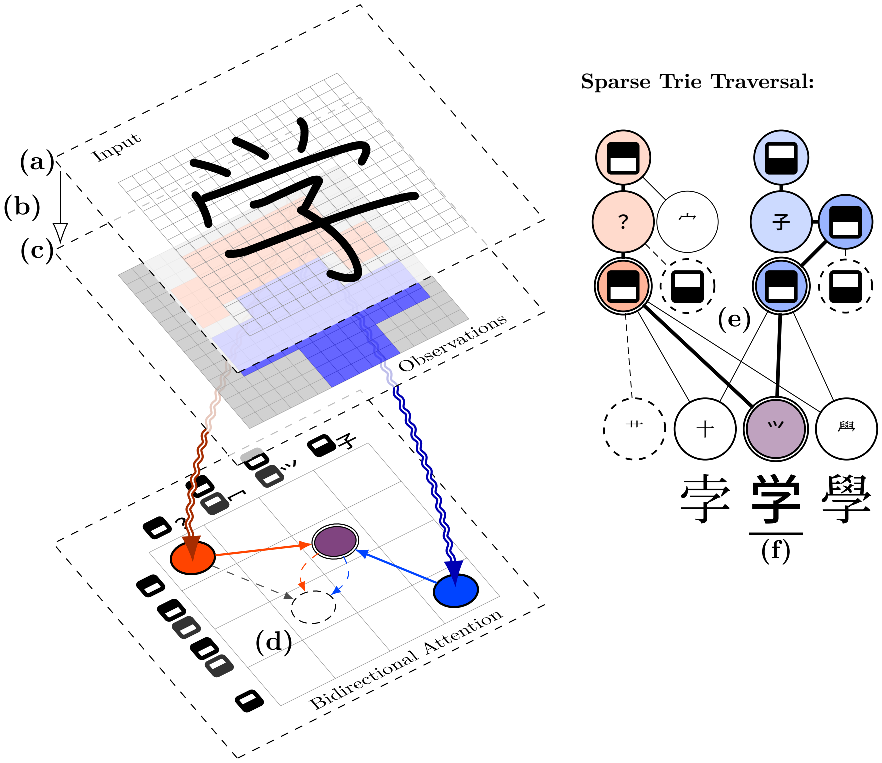

Figure 8: The overall network flow, illustrating the intuition behind the use of bidirectional attention. Removed nodes and edges are indicated with dashed lines. Non-traversed edges are shown with thin lines.

(a) The handwritten input character 学 (study) is given.
(b) The *self-attention* of the Transformer first decides regions of focus for the *bidirectional* cross attention. Forward attention is shown in orange-red, while the backwards attention is shown in blue.
(c) The first glimpse of the Observer, classifies both regions on the feature grid. Forward attention results in observation $\vec{f}_0$ = (⿱, ?). Backwards attention results in observation $\vec{b}_0$ = (⿱, 子).
(d) The bidirectional traversal removes edges from the opposing sparse trie with it's information.

- The position information of $\vec{f}_0$ and $\vec{b}_0$ removes connections to 艹 because the non-compound bottom is present with a compound top, removing 荐 and 菰.
- The remaining children of 艹 are removed (芥, 苓, 茶, 荅, 荼, 莟, 蒼, 蓉, 蔭, 蓼, 薈) because the bottom constituent is not 子.
- The ⿱⿱ connection is then fully removed without needing to observe whether 冖 or 𠆢.

(e) The encoder-decoder pair determines the next area of attention unanimously at ⿱⿱.
(f) The second observation sees ⺍, and the extraneous information of 冖 is not needed to encode the input, saving an observation cycle. The differentiation between (学/study), (孛/character), and (學/school) is achieved with the succinct encoding sequence: ⿱ 子 ⿱⿱ ⺍. When decoded, the result is 学.

6 Training & Data

To reduce the problem space for testing and initial design, only the 2,136 *jōyō kanji* will be used. These can be thought of as the core set of characters, as published by the Japanese Ministry of Education. As such this set of characters also has the most coverage in existing datasets. For *jōyō kanji*, KanjiVG provides a hierarchical structure of labeled constituents and positions arranged by stroke ordering [2]. The CASIA handwriting database will be used for handwritten data of common logographic characters [12].

Training will be done inverse to the flow of the end network, starting with the Decoder.

6.1 Training the Transformer

To train the transformer, it is important for the network to learn the next area of focus. To do this, the network must answer *"Does constituent A become before or after constituent B?"*, following the MLM described in BERT [4] [21].

For example using 学. When given ⬓子 starting state, should ⬒⺍ or ⬓冖 be focused to determine output 学. The correct choice is the one that prunes the most trees, which is ⺍ here. Thus the encoding should prefer the ⬒ as the area of focus when having state ⬓子 in the future.

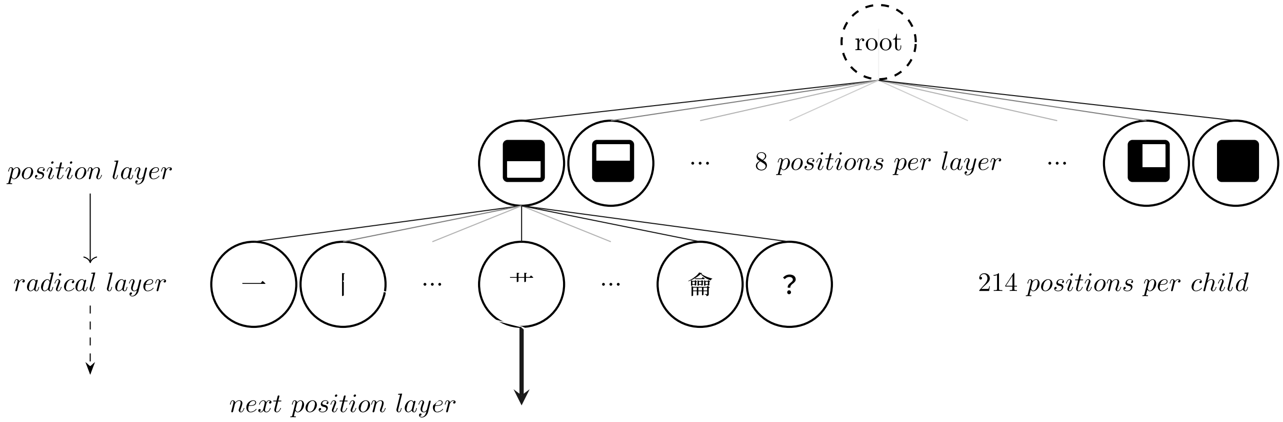

Figure 9: A tree (trie) illustrating the layers of embeddings that can be decoded by traversal, the final node contains a the Unicode character to be retrieved.

6.2 Training the Observer

Observer training will be done on an extremely dense 16x16 pixel map using rendered Unicode characters with a single tier of constituent radicals. After which the network will be trained on larger resolutions, increasing by factors of two until reaching the maximum of 512 x 512. For characters with more than two tiers of compound constituents, 32x32 will be the starting resolution. 64x64 for three tiers, etc. Each additional factor increased beyond the initial resolution will be trained on fewer epochs with a lower learning rate. This way the core features are learned at the lowest resolution and will translate when scaling up.

Handwriting samples from the CASIA-HWDB1.2 database [12] will be used to train and validate the observer's ability to detect and classify constituent radicals.

When the Observer network sees an input, the position labels of the constituents are validated first, backpropagation is done in accordance to the full loss function. However, if the classification of a radical within these bounds is incorrect, loss is only backpropogated for the radical classification layers within the architecture.

An important step in the attention mechanism is the *self-attention* mechanism, which is needed as a precursor to the *bidirectional* attention of the Transformer. YOLOv2 has improved bounding box prediction during training, utilizing k-means clustering as a means to seed bounds for individual classes [15]. This training is necessary to initially answer the questions

- ”Is this input a radical or a compound character?”
- ”If it is a compound, what are the initial position labels, and what regions should be observed by the forward and backward attention mechanisms?”

By training this self-attention separately across the subset of compounds within the CJK Unicode characters, initial attention locations can be learned from the bounding boxes and position labels. The self-attention mechanism of works similarly to the single surface attention seen in Residual Attention Networks [22] like DenseRAN [23].

6.3 Space Efficient Encoding

The existing GoogLeNet based method of Zhong et al. for handwritten Chinese character recognition has been compressed to the size of 27.77MB [26]. It is trained on the GB2312-80 standard level-1 set, containing 3,755 classes in classification, and boasts a recognition rate of 96.35%. The performance is comparable to other methods, with the leader at 96.66% [23], but has the fewest connections, taking the least space.

The classifier identifies across the 3,755 classes individually. However, the succinct representation of positional encoded radicals should be able to outperform. Although the final structure of the Observer has not been determined, reducing the number of classes to 215x8 will drastically cut the number of connections needed in the convolutional network.

As for decoding with the proposed implementation, the class size is a mere 1720 (215x8), and the cost function describing the weight of each transition can be encoded as a log probability, using only a 2 byte integer. The 3,755 classes only take 1.5 Bytes to represent.

Assuming a naive implementation where every node contains the 2 byte cost and a 2 byte integer representing the Unicode character, we can encode the transition trie of Figure 9. A naive two-dimensional matrix representation uses only (1720 x 1720 x 4 x 2) 23,667,200 Bytes (23.6MB) while wasting 4 bits per node. This is ignoring the fact that most connections will have a weight of zero, even at the 215x8 level. By using spare matrix compression, we can reduce the size drastically, which has found use in Google’s Japanese IME, which compressed it’s dictionary by 84% [9]. The sparse trie representation can be seen in Figure 8. Sparse tree boosting [10] can be used to avoid data redundancy, and the LOUDS structure [**?**] can be used in memory to represent them by removing the need for pointers.

7 A Data Set for Future Work

Following the work of WordNet [14] and ImageNet [17], I am proposing the combination of data into a network that will capture links between logographic characters on a level not previously accumulated. This data set would capture character variants, constituents, positions, phonemes, translations, stroke orderings, and various encoding representations. All of these are already available for most characters, and are derivable from existing data. However, nothing has been amassed capturing the relationship between child radicals and parent compounds. The beauty of these writing systems comes from the interconnected nature of the ideographs, rebuses, associative and phono-semantic compounds that comprise the written languages.

The modularity of the three networks of this proposed architecture allows for future fine tuning. The Decoder can be tuned to answer questions only answerable by relationships between data within this set. Given the phono-semantic nature of the characters, this can be extended to phonetic outputs in Pinyin or Hiragana. However, it may be possible that the creation of a concept graph for graphemes behind the semantic portion of the characters can provide benefits in predictive input for IMEs, sentiment analysis, or even previously undiscovered insights on ancient writings of the Oracle bone script. It is my hope that such a network can provide sources for answers to these problems.

References

[1] The unicode standard version 13.0 – core specification, Mar 2020. 3, 4

[2] U. Apel. Kanjivg, 2009. 8

[3] M. Davis. Psycholinguistic evidence on scrambled letters in reading, 2012. 4

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 4, 5, 8

[5] S. He and L. Schomaker. Open set chinese character recognition using multi-typed attributes. *CoRR*, abs/1808.08993, 2018. 1

[6] S. Jaeger, C.-L. Liu, and M. Nakagawa. The state of the art in japanese online handwriting recognition compared to techniques in western handwriting recognition. *International Journal on Document Analysis and Recognition*, 6:75–88, 10 2003. 2

[7] D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbune. Multi-language online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1

[8] P. K. Kim and H. J. Kim. Off-line handwritten korean character recognition based on stroke extraction and representation. *Pattern Recognition Letters*, 15(12):1245 – 1253, 1994. 2

[9] T. Kudo, T. Hanaoka, J. Mukai, Y. Tabata, and H. Komatsu. Efficient dictionary and language model compression for input method editors. In *Proceedings of the Workshop on Advances in Text Input Methods (WTIM 2011)*, pages 19–25, Chiang Mai, Thailand, Nov. 2011. Asian Federation of Natural Language Processing. 9

[10] T. Kudo, J. Suzuki, and H. Isozaki. Boosting-based parse reranking with subtree features. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, page 189 – 196, USA, 2005. Association for Computational Linguistics. 9

[11] C.-L. Liu, S. Jaeger, and M. Nakagawa. Online recognition of chinese characters: The state-of-the-art. *IEEE transactions on pattern analysis and machine intelligence*, 26:198–213, 03 2004. 2

[12] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang. Casia online and offline chinese handwriting databases. pages 37 – 41, 10 2011. 1, 8

[13] K. Lunde. Ids + opentype: Pseudo-encoding unencoded glyphs, Mar 2014. 4

[14] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39 – 41, Nov. 1995. 2, 9

[15] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016. 1, 9

[16] X. Ren, K. Chen, and J. Sun. A CNN based scene chinese text recognition algorithm with synthetic data engine. *CoRR*, abs/1604.01891, 2016. 1

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 9

[18] G. Sampson, C. Zhiqun, 散復生, and 陳志群. The reality of compound ideographs / 論會意字的眞實性. *Journal of Chinese Linguistics*, 41(2):255–272, 2013. 2

[19] F. Sheng, C. Zhai, Z. Chen, and B. Xu. End-to-end chinese image text recognition with attention model. In D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, editors, *Neural Information Processing*, pages 180–189, Cham, 2017. Springer International Publishing. 4

[20] User:Emphrase. Han unification - differences for the same unicode character (u+8fd4) in regional versions of source han sans, Dec 2020. 3

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 5, 8

[22] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. *CoRR*, abs/1704.06904, 2017. 4, 9

[23] W. Wang, J. Zhang, J. Du, Z. Wang, and Y. Zhu. Denseran for offline handwritten chinese character recognition. *CoRR*, abs/1808.04134, 2018. 1, 4, 9

[24] S.-L. Yeh, W.-L. Chou, and P. Ho. Lexical processing of chinese sub-character components: Semantic activation of phonetic radicals as revealed by the stroop effect. *Scientific Reports*, 7(1), Nov 2017. 2

[25] C. Zhai, Z. Chen, J. Li, and B. Xu. Chinese image text recognition with blstm-ctc: A segmentation-free method. In T. Tan, X. Li, X. Chen, J. Zhou, J. Yang, and H. Cheng, editors, *Pattern Recognition*, pages 525–536, Singapore, 2016. Springer Singapore. 4

[26] Z. Zhong, L. Jin, and Z. Xie. High performance offline handwritten chinese character recognition using googlenet and directional feature maps. *CoRR*, abs/1505.04925, 2015. 1, 9

List of Figures