Jointly Learning Linguistically Rich Chinese Character Representation with Multi-Head Attended Convolutional Network

Zhangyu Wang
zhangyuwang@umass.edu

1 Introduction

Word embeddings have been increasingly augmented with sub-word level information for many applications such as named entity recognition (Lample et al., 2016), part-of-speech tagging (Plank et al., 2016), dependency parsing (Ballesteros et al., 2015; Yu and Vu, 2017), and language modelling (Kim et al., 2015). Most of these models employ a CNN or a BiLSTM that takes as input the characters of a word and outputs a character-based word representation. The intuition is linguistically straightforward: many languages, especially the morphologically rich ones, have strong sub-word structures that are highly semantically informative.

Chinese character representation, however, is an even more complex scenario. In general, standard character-based approach performs less successfully on Chinese word embedding tasks, for Chinese characters, unlike alphabetic languages, are logographic and hide huge amount of semantic information in their sub-character structures. Attempts have been made to utilize this information by decomposing Chinese characters into more basic parts along different granularity, like radicals (Sun et al., 2014), components (Li et al., 2015), or strokes (Cao et al., 2018). However, these methods do not perform consistently better in downstream applications and have various restrictions. Research also shows that simple sub-character level embeddings are still not enough but it requires fine-grained joint learning simultaneously on word, character, and sub-character component levels (Yu et al., 2017).

A very recent research demonstrates that due to the logographic essence of Chinese characters, a tailored convolutional network (CNN) can automatically extract glyph features from images of characters and learn the glyph-based character embeddings with both a task-specific objective and a regularizing auxiliary objective (Wu et al., 2019). To introduce more sub-character information and augment the training data in order to prevent overfitting, this work also innovatively proposed to use historical scripts instead of solely standard simplified Chinese characters. They report that glyph-based character embeddings perform consistently better in 13 general downstream NLP tasks, which is an enormous leap forward compared with previous works.

However, the glyph-based model needs further improvements. It treats Chinese characters as pure images and feeds the extracted features simply to a standard image classification task, which is a complete under-exploitation of the overly rich semantic information within Chinese characters. The key problem is, while in the paper they did observe the two distinct approaches of decomposing Chinese characters into more primitive parts in previous works, that is, function-decomposition (decomposing characters into separated functional, meaningful radicals) (Shi et al., 2015; Li et al., 2015; Yin et al., 2016; Sun et al., 2014) and shape-decomposition (decomposing characters into separated functionless, meaningless strokes and stroke combinations, based on Wubi encoding scheme) (Xue Tan et al., 2018), they failed to notice that these two approaches are not mutually exclusive, but simply different aspects of the underlying structure of Chinese characters. In fact, linguistically, a component of a Chinese character can have one of the three sub-character functions: denoting the shape of the character (字形), denoting the meaning of the character (字义) and denoting the sound of the character (字音). Learning on a simple image classification task is confusing the three distinct functions of character components and loses a huge amount of semantic information.

We propose, to learn the glyph-based character representation, we need to simultaneously attend to all three aspects of graphics, semantics, and phonetics. The importance of learning about shapes and meanings is fully explored in previous works (Shi et al., 2015; Xue Tan et al., 2018), while the employment of phonetic information in Chinese character representation is rarely researched (Zhu et al., 2018). We insist that this is a critical component of our model, because it is an extremely common phenomenon in Chinese (especially phrases, idioms and historical or literal texts) that two characters are semantically associated when they sound similarly (通假字).

To enable such multi-task learning, we propose to design three respective auxiliary tasks based on the hidden state of the Tianzige-CNN architecture (Wu et al., 2019). For the shape task, unlike the naive image classification task in Wu et al. (2019), we will ask the model to "split" the character into composing radicals. This will force the CNN to identify correct sub-character components and help with meaning and sound tasks. For the meaning task, we will ask the model to predict related characters. For the sound task, we will ask the model to predict the consonant part and vowel part of the character's pronunciation respectively (all Chinese characters are monosyllabic). We will also adopt the scheduled loss trade-off method used in Wu et al. (2019), as the meaning task and the sound task rely on the segmentation of characters by the CNN part to make a prediction, and thus the model will learn to represent graphic information significantly faster than semantic and phonetic information.

Collecting data for our model is not a hard work. For the shape task, there are existing radical-decomposition look-up tables for Chinese characters (Sun et al., 2014). For the meaning and the sound tasks, digital versions of traditional Chinese dictionaries[1] and linguistic works[2] are the best source of training data, because they present their entries as "A is defined as B, and sounds like C in terms of consonant, like D in terms of vowel". Very simple rule-based algorithms will be good enough to extract training pairs.

[1]For example, Kang-Xi-Zi-Dian, 康熙字典
[2]For example, Shuo-Wen-Jie-Zi, 说文解字

2 Related work

Decomposing words into more primitive components is a long-studied problem. In the perfect theoretical setting of a neural $n\text{-}gram$ model of language, the semantic role of a word is only dependent on its neighbors (Chen and Goodman, 1996; Bengio et al., 2003; Mikolov et al., 2013). If n and the size of dataset goes to infinity, theoretically we should be able to learn the accurate meaning of the word. For synthetic languages, like English and German, however, this hypothesis does not hold even if we have infinite data and computation resources. A huge amount of semantic information is enclosed inside the word and invisible to the model. For example, "is" and "was" have almost identical context, while the relationship between "event" and "eventually" can not be effectively represented a priori (Kim et al., 2015). There are some other issues when dataset is limited in size or comes from some special domains, for example, that rare words and unseen words will be learned and embedded very poorly. Apparently, if we can decompose words into more primitive components, these problems will be largely alleviated.

One straightforward attempt to solving this problem is sememe computation and knowledge graph. WordNet (Miller, 1995) and HowNet (Dong and Dong, 2003) are two important examples. The assumption is that word senses can be decomposed into indivisible units, i.e., sememes. Papers report that structured lexical database help improving word embeddings (Niu et al., 2017) and other downstream tasks (Xie et al., 2017; Zeng et al., 2018). However, these methods are all based on huge, expensive human-annotated dataset. Another concern is that semantics of words flows with time; it is way better if we have some automated, data-driven methods that learn sub-word level semantics within the training corpora. Sub-word level language modeling is the direct solution (Mikolov et al., 2011). Specifically, for most alphabetic languages, character-based neural language models are most intuitive. The learned character-level language model is able to identify word components like prefixes, suffixes and others (Kim et al., 2015). Reasonably, character-based neural language models are able to deal with rare and unseen words effectively.

Chinese, however, is a different case. Similarly, Chinese words are composed of characters.

Differently, they are logographic instead of alphabetic. There are up to 10000 characters, among which 3000 are commonly used, and in most cases there are no more than 5 characters in one Chinese word. This makes encoding Chinese characters naturally sparse: we just can't simply use one-hot vectors for representing them. We need dense representation of Chinese characters.

Previous work has mostly focused on decomposing Chinese characters into more primitive components. There are two major approaches. One approach is function-decomposition. Lexicologically, a Chinese character can be divided into different radicals, each having its own function. Li et al. (2015) and Sun et al. (2014) used skip-gram (Mikolov et al., 2013) to learn radical representation. Some improvements in downstream tasks are reported. Yin et al. (2016) found radical representation helpful in word-similarity tasks, while Nguyen et al. (2018) reported that radical representation can be used to improve word pronunciation prediction. Another approach is shape-decomposition. Chinese characters can be divided into meaningless strokes, and listing the strokes in some way describes the shape of the character. Wubi is a popular stroke-based Chinese character encoding scheme and that enables generating standard stroke decomposition for each character. Xue Tan et al. (2018) found Wubi encoding improves performance in machine translation tasks.

One disadvantage of these methods is that they require standard, unified radical/stroke decomposition, while there are multiple historically and currently used scripts of Chinese characters, many of which are very different from each other in terms of radical and stroke composition (Wu et al., 2019). "Standard decomposition" is analogous to hand-written features in computer vision. If we can extract these features directly from data, we will expect to have better model generalization. The logographic features of Chinese characters encourage people to run CNN on glyphs. Dai and Cai (2017) firstly explored this possibility on language modeling and word segmentation tasks but reported negatively. Xue Tan et al. (2018) applied CNN to Wubi encoded Chinese character vectors with GloVe (Pennington et al., 2014) and skip-gram (Mikolov et al., 2013). It reported improvements on word semantic tasks. Liu et al. (2017) and Zhang and LeCun (2017) firstly attempted to learn visual features directly from character images of Chinese, Japanese and Korean, alongside with other encoded features. Despite the controversy on whether Korean characters should be considered as "logographic" (they are factually alphabetic), it is a great progress to realize that visual information of Chinese characters contains important semantic knowledge. In Wu et al. (2019), a tailored CNN architecture (Tianzige-CNN), which is very shallow and employs group CNN (Krizhevsky et al., 2017; Zhang et al., 2017), is designed to extract visual features from Chinese character images and learns character embeddings on both a task-specific objective and an auxiliary regularizing image classification objective. Huge efforts have been made to prevent overfitting, as the Chinese character dataset is very small and each character image is only 12x12 pixels in size. The authors innovatively proposed to use historical scripts of Chinese characters as data augmentation (Perez and Wang, 2017). Consistent performance improvement over 13 downstream Chinese NLP tasks was reported. This work demonstrates the huge potential of image-based Chinese character representation.

3 Our Approach

Previous works, especially the Glyce model (Wu et al., 2019) shows the power of learning Chinese character embeddings on CNN-extracted visual features. We find the image classification auxiliary objective particularly important in successful learning. This objective forces the model to attend to the graphical features that can be used to classify character images, and prevents the CNN model from overfitting on the task-specific objective, as the dataset is fairly small, with around 10000 distinct data points. This resembles the concept of multi-task learning (Collobert et al., 2011; Chen et al., 2017; Hashimoto et al., 2016; FitzGerald et al., 2015).

While this scheme proves to be very powerful, there are two critical weaknesses that must be addressed. The first weakness is that the image classification auxiliary objective is itself prune to overfitting. As is described in the paper (Wu et al., 2019), the hidden state of CNN will be forwarded directly to an image classification objective to predict its corresponding charID. This may incur problems, if we compare this to classical CIFAR-10 and CIFAR-100 tasks, because we have 10000 classes (each character is a class) while for

each class we have roughly 9 training images (Wu et al., 2019). Considering the Tianzige-CNN they proposed has a maximum channel number of 1024 and a minimum channel number of 256, it is very easy for the model to remember the training data and generalize poorly. The second weakness is that the model is not fully utilizing the information inside the character image. As we have discussed in Section 1, any segmentation of a Chinese character may serve as one of the three roles: graphic indicator, semantic indicator, or phonetic indicator. Applying a simple image classification task to the entire character image is confusing the three roles of sub-character segments and loses a huge proportion of lexical information.

To alleviate these two problems, we propose to train a similar Tianzige-CNN model with three fine-grained auxiliary objectives that forces the model to extract graphic, semantic and phonetic features respectively. Inspired by the works on radical embedding (Li et al., 2015; Sun et al., 2014; Yin et al., 2016; Nguyen et al., 2018) and stroke embedding (Xue Tan et al., 2018), the graphic objective is to predict the radicals and strokes in the given character image. Empirically, for sampling training data-target pairs, it will resemble the concept of skip-gram and negative sampling (Mikolov et al., 2013). For prediction, one possible approach is to directly do linear classification on CNN output, and another idea is to learn embeddings of radicals and strokes beforehand, and apply attention mechanism to obtain top-k predictions. We will try both architecture to decide which one performs better. As for the semantic objective and phonetic objective, the most intuitive task is to predict the top-k most semantically/phonetically closely related characters with regard to the given image, where ground-truth can be easily sampled from dictionary entries. Similarly, the prediction process can be either simple linear classification or enhanced with self-attention. This requires some burn-in epochs, that is, we need to firstly train the model solely on the graphic objective for some epochs to obtain the initial character embeddings, and then we can do self-attention on the entire character vocabulary. This can be implemented by adding a scheduled loss trade-off (Wu et al., 2019). If we add attention layer before each classification objective, this is very similar to multi-headed attention mechanism (Bahdanau et al., 2014). Such model architecture will overcome the aforementioned weaknesses in the following ways. For weakness one, the most obvious improvement is that we have 3 distinct regularizing objectives, instead of one, and that forces the model to generalize better. Besides, as the objectives are now predicting radicals/strokes/semantically relative characters/phonetically relative characters, the number of classes is reduced to around 100, which is comparable to CIFAR-100, and each character can fall into multiple classes, which in return assign more data points to each class, and thus our model is less likely to overfit. For weakness two, it's very straightforward that the three fine-grained objectives are respectively utilizing the graphic, semantic and phonetic information of the character images.

An additional idea is to utilize the sequential information of Chinese characters. As is shown in Figure 1, the scripts evolve with time. In most cases, the earlier the script is, the more logographic features it has, and the better it helps with learning character shapes, but the later the script is, the more complex combination of radicals and more regular phonetic components it has (because people started to create more compound characters to convey more compound concepts, and the usage of phonetic radicals was more standardized), and the better it helps with semantic and phonetic tasks. We can add an RNN and an attention layer on top of the CNN output and learn to attend to the script with most information needed for the specific task. The concern may be that it increases the complexity of the model and may cause more overfitting. We will carefully research the feasibility of this idea.

The baseline we are going to compare to is simply the Glyce model proposed by Wu et al. (2019). They reported consistent improvement on 13 Chinese NLP tasks: (1) character-Level language modeling, (2) word-Level language modeling, (3) Chinese word segmentation, (4) name entity recognition, (5) part of speech tagging, (6) dependency parsing, (7) semantic role labeling, (8) sentence semantic similarity, (9) intention classification, (10) Chinese-English machine translation, (11) sentiment analysis, (12) document classification, (13) discourse parsing. We will replace the Glyce-char embeddings in this model with our proposed embeddings, and remain the architecture of learning Glyce-word embeddings unchanged.

Figure 1: Historical evolution of character Ma (马, *horse*). From left to right, Seal Script (篆书, 200 B.C.), Traditional Regular Script (楷书繁体, 400 A.D.), and Simplified Regular Script (楷书简体, 1960 A.D.).

Having obtained the new word embeddings, we will feed them to the 13 Chinese NLP tasks listed above to see if we can consistently beat the reported state-of-the-art.

3.1 Milestones & Schedule

The schedule of the task is very clear.

1. Acquire and preprocess data (2-3 weeks). This includes: (1) asking the authors of Wu et al. (2019) for there Chinese character image dataset both for reproducing baseline performance and for training our model; (2) crawling and preprocessing digital versions of Chinese dictionaries (both historical and contemporary) to produce labeled data.

2. Build models for task (1-2 weeks). Implementation and unit test should be quick, but there need to be more time for carefully revising the architecture of the proposed model.

3. Run baseline codes simultaneously on Gypsum, or maybe Colab (1-2 weeks. It depends on how crowded the Gypsum server is).

4. Write progress report alongside with the data preparation and implementation (due Apr 1).

5. Analyze the output of the experiments, do an statistical analysis (2 weeks).

6. Work on final report and presentation (2 weeks).

4 Data

There are two parts of the data needed for experimentation. Firstly is the Chinese character image dataset, which is already collected and preprocessed by authors of Wu et al. (2019). We need to contact them for access. Secondly is the semantic dataset and the phonetic dataset. We need to crawl digital versions of Chinese dictionaries and extract needed data-target pairs for semantic and phonetic tasks. Part of these datasets can be obtained from existing ones. For example, stroke decomposition of characters can be inferred from the Wubi encoding (Xue Tan et al., 2018), and the radical decomposition can come straightly from the dataset used in Sun et al. (2014). There is no need for human-annotation. Of course, if time and resources allow, it's definitely better if we have a fully human-annotated dataset that is tailored for training our model.

5 Tools

We are mainly going to use the following tools:

1. Python, PyTorch and TorchVision for implementing models and pre-processing data.

2. THULAC (THU Lexical Analyzer for Chinese) for Chinese text tokenization and word segmentation (Sun et al., 2016).

3. GPU resources. One option is UMass Gypsum Cluster, another is the recommended Google Colab[3].

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Ballesteros, M., Dyer, C., and Smith, N. A. (2015). Improved transition-based parsing by modeling characters instead of words with lstms. *CoRR*, abs/1508.00657.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Cao, S., Lu, W., Zhou, J., and Li, X. (2018). cw2vec: Learning chinese word embeddings with stroke n-gram information. In *AAAI*.

[3] https://colab.research.google.com

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chen, X., Shi, Z., Qiu, X., and Huang, X. (2017). Adversarial multi-criteria learning for chinese word segmentation. *CoRR*, abs/1704.07556.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Dai, F. and Cai, Z. (2017). Glyph-aware embedding of chinese characters. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 64–69. Association for Computational Linguistics.

Dong, Z. and Dong, Q. (2003). Hownet - a hybrid language and knowledge resource. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pages 820–824.

FitzGerald, N., Täckström, O., Ganchev, K., and Das, D. (2015). Semantic role labeling with neural network factors. In *EMNLP*.

Hashimoto, K., Xiong, C., Tsuruoka, Y., and Socher, R. (2016). A joint many-task model: Growing a neural network for multiple NLP tasks. *CoRR*, abs/1611.01587.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2015). Character-aware neural language models. *CoRR*, abs/1508.06615.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Li, Y., Li, W., Sun, F., and Li, S. (2015). Component-enhanced chinese character embeddings. *CoRR*, abs/1508.06669.

Liu, F., Lu, H., Lo, C., and Neubig, G. (2017). Learning character-level compositionality with visual features. *CoRR*, abs/1704.04859.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Mikolov, T., Sutskever, I., Deoras, A., Le, H.-S., Kombrink, S., and Cernocký, J. (2011). Subword language modeling with neural networks.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Nguyen, M., Ngo, H. G., and Chen, N. F. (2018). Multimodal neural pronunciation modeling for spoken languages with logographic origin. *CoRR*, abs/1809.04203.

Niu, Y., Xie, R., Liu, Z., and Sun, M. (2017). Improved word representation learning with sememes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2049–2058. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621.

Plank, B., Søgaard, A., and Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *CoRR*, abs/1604.05529.

Shi, X., Zhai, J., Yang, X., Xie, Z., and Liu, C. (2015). Radical embedding: Delving deeper to chinese radicals. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 594–598. Association for Computational Linguistics.

Sun, M., Chen, X., Zhang, K., Guo, Z., and Liu, Z. (2016). Thulac: An efficient lexical analyzer for chinese.

Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., and Wang, X. (2014). Radical-enhanced chinese character embedding. *CoRR*, abs/1404.4714.

Wu, W., Meng, Y., Han, Q., Li, M., Li, X., Mei, J., Nie, P., Sun, X., and Li, J. (2019). Glyce: Glyph-vectors for Chinese Character Representations. *arXiv e-prints*.

Xie, R., Yuan, X., Liu, Z., and Sun, M. (2017). Lexical sememe prediction via word embeddings and matrix factorization. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4200–4206.

Xue Tan, M., Hu, Y., I. Nikolov, N., and H. R. Hahnloser, R. (2018). wubi2en: Character-level chinese-english translation through ascii encoding.

Yin, R., Wang, Q., Li, P., Li, R., and Wang, B. (2016). Multi-granularity chinese word embedding. In *EMNLP*.

Yu, J., Jian, X., Xin, H., and Song, Y. (2017). Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 286–291. Association for Computational Linguistics.

Yu, X. and Vu, N. T. (2017). Character composition model with convolutional neural networks for dependency parsing on morphologically rich languages. *CoRR*, abs/1705.10814.

Zeng, X., Yang, C., Tu, C., Liu, Z., and Sun, M. (2018). Chinese liwc lexicon expansion via hierarchical classification of word embeddings with sememe attention. In *AAAI*.

Zhang, T., Qi, G., Xiao, B., and Wang, J. (2017). Interleaved group convolutions for deep neural networks. *CoRR*, abs/1707.02725.

Zhang, X. and LeCun, Y. (2017). Which encoding is the best for text classification in chinese, english, japanese and korean? *CoRR*, abs/1708.02657.

Zhu, W., Jin, X., Ni, J., Wei, B., and Lu, Z. (2018). Improve word embedding using both writing and pronunciation. *PLOS ONE*, 13:e0208785.