Named Entity Recognition and Compositional Morphology for Word Representations in Chinese

Emily Ling
Computer Science
Stanford University
eling8@stanford.edu

Cindy Lin
Computer Science
Stanford University
cinlin@stanford.edu

Christopher Heung
Computer Science
Stanford University
cheung2@stanford.edu

Abstract

We approach the task of named entity recognition (NER) for Chinese by representing each entity with a composition of its traits: the token-entity, its characters, and its characters' main radicals. Character and radical-level information for each entity are included to provide additional relationships that might not be strictly captured within a token-entity's word embedding during training. We learn using neural networks that are some combination of the following traits: unidirectional or bidirectional; single or multi-layer; simple, gated recurrent unit (GRU), or long short term memory (LSTM) celled. We achieve a maximum not-O token-level F1 score of 76 and entity-level F1 of 70.

1 Introduction

NER is a task in Natural Language Processing (NLP) that is concerned with taking an input sentence and identifying entities in the text which fall into categories such as PERSON, ORGANIZATION, and LOCATION. Most NER efforts have been focused on classifying entities in English sentences, while other languages have had less focus. This paper seeks to further explore NER for the Chinese written language.

There are key differences between the Chinese and English languages. Chinese entities are composed of 50,000 characters, unlike English entities which are composed of 26 letters. Each Chinese character holds intrinsic meaning, but also embodies and contributes to a wide variety of meanings depending on which entity it is part of and its surrounding context. This differs from English, as the letter "c" by itself holds no intrinsic meaning. Additionally, obvious English features used in NER such as word capitalization do not exist in Chinese. The Chinese written language also does not contain spaces between entities, so word segmentation is an additional challenge. Thus, imperfect segmentation techniques must be performed on datasets before NER-work, adding another layer of difficulty.

An additional feature we have decided to include that is specific to the Chinese language is each character's main radical. A radical is a sub-part of a character. There are 214 radicals, which were identified historically by the Qing Emperor Kangxi; today, radicals are used as the basis for Chinese dictionary indexing. Radicals help to convey additional meaning behind each character. Take for example, the characters 食 (food), 餓 (hungry), 飯 (rice), and 餐 (meal). All of these characters have the main radical 食 and are related to food. We establish and feed to our neural network additional relational information through radical embeddings we train using a skip-gram word2vec model.

2 Related Work

Chinese NLP is a field that has produced a fair amount of mixed results. Peng and Dredze (2015) from Johns Hopkins University worked with Chinese NER on a dataset of Weibo messages [1], one

of the first groups to apply Chinese NER to less formal domains. The group produces F1 scores consistently below 48.90 using word and character embeddings trained with word2vec with skip-gram and negative sampling. This is in stark contrast to English NER, which generally produces F1 scores between 80 and 95 [2].

From a morphological perspective, radicals have been previously used in NLP tasks. Fandrianto, Natarajan, and Zhu (2012) from Stanford University applied Chinese radicals to the tasks of language modeling, part-of-speech tagging, and word segmentation [3]. They noted no significant improvements from the inclusion of radicals, but did note that in specific niche cases such as out-of-vocab words, the radicals were useful. While this paper did not employ deep learning and was not focused on NER, it explores the inclusion of sub-components of the Chinese language for NLP tasks.

Chinese NER has also experienced exciting improvements in the last few years. A successful bilingual approach was implemented by Wang, Che, and Manning (2013) from Stanford University, showing vast improvement in the Chinese NER task by using a semi-supervised learning model on unannotated bilingual text and achieving a maximum F1 score of 74.32 [4]. Additionally, Wu, Jiang, Lei, and Xu (2015) from the University of Texas Health Science Center explored the use of a deep neural network applied on a singular domain, Chinese clinical text, and were able to achieve a F1 of 92.80 [5]. We aim to generalize even further in comparison to their approach, using data from multiple domains and source types and not restricting ourselves to just clinical text.

3 Approach

3.1 Data

3.1.1 Chinese Corpus

We use the OntoNote (v5.0) corpus [6], a dataset put together collaboratively by BBN Technologies, the University of Colorado, the University of Pennsylvania, and the Information Sciences Institute. The dataset draws from a wide range of text sources, including news, telephone conversations, websites, talk shows, broadcast television, and more.

In addition to structural and semantic information, the data also contains manually annotated Named Entity tags for nineteen types such as: PERSON, FACILITY, ORGANIZATION, LOCATION, PRODUCT, EVENT, WORK OF ART, etc. The tag "O" is used to denote the lack of an entity. Because many of these tags are in similar categories or rarely appear in the dataset, we reduce the number of Named Entity tags in our model from nineteen to five for our purposes. Considering only a few major categories during NER is precendented in many of the papers mentioned above. The five categories we used and the subcategories from the original dataset are shown below:

Table 1. Reduced labels and corresponding corpus labels

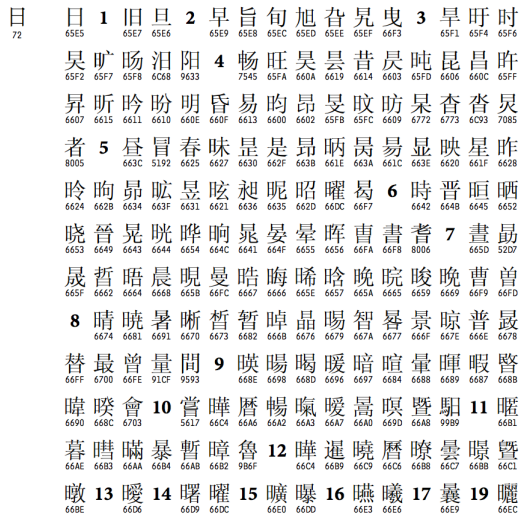
Category	Tags from original dataset
LOC	LOC, GPE
ORG	ORG, NORP
PERSON	PERSON
QUANT	MONEY, QUANTITY, ORDINAL, CARDINAL, DATE, TIME, PERCENT
O	O, FAC, PRODUCT, EVENT, WORK OF ART, LAW, LANGUAGE

3.1.2 Radicals

We use the Unihan Database [7] to obtain the main radical for each Chinese character. We specify "main" here, as most Chinese characters contain multiple radicals. The database also contains information on remaining stroke count, which we have chosen not to include in our training as we thought it would be less correlated to character meaning.

As an example of data from the Unihan database, below are all the characters with the main radical 日, meaning day. The bolded numbers indicate remaining stroke count in the character after the main radical is discounted.

Image 1. Chinese characters with the main radical 日



For our purposes, we train radical embeddings for all 214 radicals on the OntoNote corpus using a word2vec model with skip-gram and cross-entropy loss.

3.1.3 Word Embeddings

Facebook recently published pre-trained word embeddings for 90 languages including Chinese, trained using a skip-gram model described in Bojanowski (2016) [8]. These vectors are of dimension 300 and correspond to entities rather than individual characters. We use both these pre-trained word embeddings as well as randomly initialized embeddings in our work. From the OntoNote corpus, around 76.8% of our tokens have corresponding Facebook embeddings, whereas the remainder do not and are randomly initialized during training.

3.2 Input Features

We use several combinations of features when feeding our input into the model, composed of token, character, and main radical-level embeddings:

Table 2. Feature sets

Category	Features (n)
token	token (1)
char	character (1)
token-char	token, first character of token (2)
char-rad	character, main radical of character (2)
token-char-char	token, first, second characters of token (3)
token-char-rad	token, first character of token, main radical of first character (3)

For example, given the token 冰淇淋, the following describes a singular input to the model given the feature set type:

token: [冰淇淋]
char: [冰]
token-char: [冰淇淋 , 冰]
char-rad: [冰 , 水]
token-char-char: [冰淇淋 , 冰 , 淇]
token-char-rad: [冰淇淋 , 冰 , 水]

For the feature set types char and char-rad, every character is inputed and later labeled as a separate entity. Thus for feature set char-rad, 冰淇淋 is fed to the model as three inputs: [冰, 水], [淇, 水], [淋, 水].

3.3 Models

3.3.1 Baseline

We implemented a baseline that parses the training data, records the entity tags for each character, and then naively labels a test token using the most frequently encountered entity tag for all characters in the token. We default to labeling as 'O' if the test token did not show up in training. An example of a character and its encountered entities:

民: {
  ORG: 1,
  PERSON: 4,
  WORK OF ART: 1
}

3.3.2 Neural Network

For parameters for all neural networks, we have chosen to set them as detailed:

Table 3. Neural network parameters

dropout	0.5
learning rate	0.005
embed size	150
hidden size	300
window size	2

Adjustments to the above parameters were applied to learning rate and window size with no difference observed in the resulting F1 score. The remaining parameters were chosen to maintain computational feasibility without sacrificing significant performance.

3.3.3 Unidirectional Recurrent Neural Network

$$\begin{aligned} e^{(t)} &= x^{(t)}L \\ h^{(t)} &= \sigma(h^{(t-1)}W_h + e^{(t)}W_x + b_1) \\ \hat{y}^{(t)} &= softmax(h^{(t)}U + b_2) \end{aligned}$$

We use an unidirectional recurrent neural network (RNN) and initialized our weights and biases using the Xavier optimization function.

3.3.4 Unidirectional RNN with GRU

$$\begin{aligned} z^{(t)} &= \sigma(x^{(t)}U_z + h^{(t-1)}W_z + b_z) \\ r^{(t)} &= \sigma(x^{(t)}U_r + h^{(t-1)}W_r + b_r) \\ \tilde{h}^{(t)} &= tanh(x^{(t)}U_h + r^{(t)} \circ h^{(t-1)}W_h + b_h) \\ h^{(t)} &= z^{(t)} \circ h^{(t-1)} + (1 - z^{(t)}) \circ \tilde{h}^{(t)} \end{aligned}$$

We use an unidirectional RNN with GRU cells, which are better at capturing long-term dependencies due to the fact that GRU cells have update and reset gates.

3.3.5 Unidirectional RNN with LSTM

$$\begin{aligned} i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \\ f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \\ o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \\ \tilde{c}_t &= \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\ h_t &= o_t \circ \tanh(c_t) \end{aligned}$$

We use an unidirectional RNN with LSTM cells, which are similarly motivated in comparison to GRU cells, but contain more gates: new memory generation, input, forget, final memory generation, and output/exposure gates.

3.3.6 Bidirectional, Multi-Layer RNN

We use bidirectional RNNs with simple, GRU, and LSTM cells to make predictions based on tokens both before and after our current token. This enables us to capture even more complex relationships existing in our corpus.

3.4 Evaluation

We split the OntoNote dataset into 80/10/10 portions for train, dev, and test, respectively. We then output label predictions per token, compare them to the golden labels, and calculate the F1 score. Since we break down our entities into small components during learning, we thought it prudent to examine not only the entity-level precision, recall, and F1 scores, but also the token-level precision, recall, and F1 scores pertaining to the not-O labels, ie. PERSON, ORG, LOC, and QUANT.

The F1 score is defined as follows:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

where precision denotes the ratio of correct labels to total number of labels our model predicted and recall denotes the ratio of correct labels to total number of actual labels in the golden set.

4 Experiments

4.1 Data

Our baseline model yielded a token-level F1 score of 34. We began by testing the 6 different input feature sets on a unidirectional RNN using a simple RNN and the neural network parameters in Table 3. Below are the resulting F1 scores.

Table 4. Unidirectional RNNs with randomly initialized word/radical embeddings

Features	PERSON	ORG	LOC	QUANT	Not-O	Entity
char	74	54	68	81	71	55
char-rad	74	47	67	81	69	55
token	63	54	68	79	69	61
token-char	69	59	74	81	73	66
token-char-rad	75	56	70	82	73	67
token-char-char	77	58	73	81	74	68

To compare the effectiveness of the Facebook pretrained word embeddings and our word2vec-trained radical embeddings, we have the following runs, named in the form

$$\{\text{word embedding}\}\text{-}\{\text{radical embedding}\}$$

where 'r' indicates randomly initialized whereas 'p' indicates the use of pretrained embeddings.

Table 5. Unidirectional RNNs with pre-trained word and/or radical embeddings

Features	Embed	PERSON	ORG	LOC	QUANT	Not-O	Entity
token-char-char	p-r	80	61	75	81	76	70
char-rad	r-p	76	56	68	80	71	55
token-char-rad	r-p	70	60	74	81	74	65

We ran the most promising configurations above with GRU and LSTM RNNs using Facebook's word embeddings and randomly initialized radical embeddings.

Table 6. Unidirectional GRU, LSTM RNNs with pretrained word embeddings

Cell	Features	PERSON	ORG	LOC	QUANT	Not-O	Entity
GRU	token-char-char	79	61	74	81	75	70
GRU	token-char-rad	76	52	70	80	71	64
LSTM	token-char-char	81	58	73	82	76	70
LSTM	token-char-rad	77	59	72	82	74	68

Lastly, we selected the two highest-performing configurations from above to run on bidirectional, single and multi-layer LSTM RNNs, all using Facebook's word embeddings.

Table 7. Bidirectional single, multi-layer LSTM RNNs with pretrained word embeddings

Features	Layers	PERSON	ORG	LOC	QUANT	Not-O	Entity
token-char-char	1	78	62	74	81	75	69
token-char-char	3	76	60	72	82	75	69
token-char-rad	1	75	59	68	81	73	67
token-char-rad	3	75	59	74	81	74	68

4.2 Analysis

4.2.1 Unidirectional RNNs

In Table 4, we first compare the performance of character-level and token-level features. At first glance, "token" outperforms "char" in entity-level F1 with scores of 61% to 55%, respectively. However, "char" achieves an 11% increase in PERSON F1 which contributes to a 2% increase in not-O F1 compared to "token." We conjecture that character-level features are especially helpful in identifying names, since full names are unlikely to appear in training data unless the person is famous. Because there is a small subset of Chinese characters used as common last names, character-level features have much more predictive power than token-level features. With this in mind, combining character-level features with token-level features in "token-char" and "token-char-char" significantly improves performance, bringing entity-level F1 up to 68% and achieving significant improvements in all F1 categories. Lastly, "token-char-char" is the highest performing model in our comparison thus far, which shows that character-level information, when combined with the token as a whole, is extremely predictive in labeling.

To assess radical-level features, we compare "token-char" and "token-char-rad" and see that radicals as a feature do not clearly improve performance. An exception occurs for the PERSON F1 score, which sees a 6% increase with radicals. This suggests that certain radicals are highly indicative of Chinese names.

Comparing Facebook's pretrained word embeddings with randomly initialized embeddings in Table 5, we observe that pretrained word embeddings improved F1 scores in almost every category for "token-char-char"; most notably, not-O token-level and entity-level F1 scores both jumped 2% with the use of pretrained embeddings. Next, comparing our pretrained radical embeddings with randomly initialized embeddings, we see that pretrained embeddings improved F1 scores for "char-rad," with a 9% increase in ORG F1. This suggests a complex relationship between a radical and its usage in an ORG entity that cannot be adequately captured with randomly initialized embeddings. However, pretrained radical embeddings had little impact on "token-char-rad" F1 scores, improving ORG and not-O level scores but worsening PERSON and entity-level scores.

4.2.2 Unidirectional GRU, LSTM RNNs

Overall, using GRU and LSTM cells instead of simple cells did not improve the model. They decreased or matched performance in every category when compared to RNNs with simple cells with the exception of the PERSON F1 score in "token-char-rad." In this special case, GRUs improved the F1 score from 70% to 76% and LSTMs to 77%. This suggests that GRUs and LSTMs are more capable of capturing context from previous tokens in the sentence when predicting a PERSON label. They are also superior at discerning whether or not an encountered common family name is being used in a PERSON or not-PERSON context. Because LSTM cells at least matched performance of RNN cells in every category, we proceeded with LSTM cells for later experiments.

4.2.3 Bidirectional RNNs

Using bidirectional LSTM RNNs either decreased or matched performance in most categories compared to unidirectional LSTM RNNs. Of note is the ORG category in "token-char-char," which increased from 58% to 62%/60% for bidirectional single/multilayer neural nets, respectively. This

suggests that predicting organization labels is significantly aided by context from tokens occurring after the ORG entity.

The positive benefit of the triple-layered neural net is seen when comparing within the "token-char-rad" feature set runs. The LOC category F1 score increased from 68% to 74% when the number of layers was increased from 1 to 3. This suggests that the added complexity when predicting location entities is extremely helpful.

Otherwise, no drastic improvement is seen in the bidirectional runs in comparison to unidirectional ones, possibly due to overfitting. It might also be true that Chinese entities are not as affected by proceeding tokens, and thus, using the bidirectional net harmed rather than helped our predictive ability.

5 Conclusions

Overall, we see that "token-char-char" is the best performing model. In particular, the unidirectional RNN using pretrained Facebook word embeddings achieved the highest not-O F1 score of 76% and entity-level F1 score of 70%. The inclusion of characters in the feature set helped tremendously overall while the inclusion of radicals helped specific entity types (ie. PERSON) but not the overall score.

For future work, we will investigate using all of the token's characters when constructing features, as well as train the bidirectional LSTM on more epochs to ensure convergence. Further fine-tuning of parameters such as learning rate and step size could also improve results.

6 Acknowledgements

We would like to thank Chris Manning for his mentorship on this project as well as the Stanford Computer Science Department for providing access to the OntoNotes corpus and Microsoft Azure virtual machines. We would also like to thank the CS224N course staff for the code provided for assignment 3, which we used as the starting point for our project.

References

[1] Peng, N., and Dredze, M. (2015). Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 548-554). Lisbon, Portugal: Association for Computational Linguistics. Retrieved from https://aclweb.org/anthology/D/D15/D15-1064.pdf

[2] Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (pp. 363-370). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.3115/1219840.1219885

[3] Fandrianto, A., Natarajan, A., and Zhu, H. (2012). Chinese Radicals in NLP Task. Stanford University. Retrieved from https://nlp.stanford.edu/courses/cs224n/2012/reports/report.pdf

[4] Wang, M., Che, W., and Manning, C. D. (2013). Effective Bilingual Constraints for Semi-supervised Learning of Named Entity Recognizers. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (pp. 919-925). Bellevue, Washington: AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=2891460.2891588

[5] Wu, Y., Jiang, M., Lei, J., and Xu, H. (2015). Named Entity Recognition in Chinese Clinical Text Using Deep Neural Network. Studies in Health Technology and Informatics, 216, 624-628.

[6] Weischedel, R. (2013, October 16). OntoNotes Release 5.0 LDC2013T19. Retrieved March 18, 2017, from https://catalog.ldc.upenn.edu/LDC2013T19

[7] Unicode, Inc. (2017). Unihan Database Lookup. Retrieved March 18, 2017, from http://www.unicode.org/charts/unihan.html

[8] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching Word Vectors with Subword Information. arXiv:1607.04606 [Cs]. Retrieved from http://arxiv.org/abs/1607.04606

7 Group Member Contributions

Cindy Lin: wrote project proposal, researched prior work, located and acquired data, split data, adapted word2vec code to work on Chinese, trained word and radical embeddings, tested quality of pretrained embeddings, debugged performance issues and problems, planned and ran large number of tests, wrote scripts to run tests, wrote and finalized report, designed and edited and finalized poster

Emily Ling: wrote project proposal, researched prior work, located and acquired data, cleaned and processed data, got basic NER model running on our data, implemented radical/token/character-level features, implemented multilayer and bidirectional RNN, tested quality of pretrained embeddings, debugged performance issues and problems, tuned RNN parameters, planned and ran large number of tests, edited and finalized report

Chris Heung: made baseline model train predictions, helped add pretrained radical embeddings to models, helped run results and evaluate models, wrote rough draft of report, made rough draft of poster