Chinese Long and Short Form Choice Exploiting Neural Network Language Modeling Approaches

Lin Li
Utrecht University
Utrecht, the Netherlands
Qinghai Normal University
Xining, China
l.li1@uu.nl

Kees van Deemter
Utrecht University
Utrecht, the Netherlands
c.j.vandeemter@uu.nl

Denis Paperno
Utrecht University
Utrecht, the Netherlands
d.paperno@uu.nl

Abstract

Lexicalisation is one of the most challenging tasks of Natural Language Generation (NLG). This paper presents our work in choosing between long and short forms of elastic words in Chinese, which is a key aspect of lexicalisation. Long and short forms is a highly frequent linguistic phenomenon in Chinese such as 老虎-虎 *(laohu-hu, tiger)*. The choice of long and short form task aims to properly choose between long and short form for a given context to producing high-quality Chinese.

We tackle long and short form choice as a word prediction question with neural network language modeling approaches because of their powerful language representation capability. In this work, long and short form choice models based on the-state-of-art Neural Network Language Models (NNLMs) have been built, and a classical n-gram Language Model (LM) is constructed as a baseline system. A well-designed test set is constructed to evaluate our models, and results show that NNLMs-based models achieve significantly improved performance than the baseline system.

1 Introduction

The long and short form of an elastic word refers to words have different word length (i.e. number of syllables) but share at least one identical word meaning such as 丢失-丢 *(diushi-diu, lose)*. Duanmu(Duanmu, 2013) points out that as high as 80% percent of Chinese words has both long and short forms, therefore Chinese speakers need to make the choice between long and short forms during daily communication. Like human speakers and writers, the long and short form choice task also needs to be carefully resolved for various domain including Natural Language Generation(Inkpen and Hirst, 2004), Machine Translation(Nguyen and Chiang, 2017), and Style Transfer(Fu et al., 2018).

In this work, we focus on long and short forms that share at least one same word meaning and one same morpheme, but compose of different number of syllables. The long and short form choice task is formulated as Fill-in-the-blank (FITB) task(Inkpen and Hirst, 2004; Zweig et al., 2012), whose goal is to select a missing word for a sentence from a set of candidates. A FITB example used in this work is shown in Table 1.

Sentence	Long Form	Short Form
她去日本旅游时，必逛各种免税 _____。	*(1)* 商店	*(2)* 店
When travels to Japan, she must go to duty free_____.	*(1) shop*	*(2) shop*

Table 1: A long and short form choice FITB question example.

The lexical choice is difficult in the context of long and short forms for most language processing systems due to the identical word sense leading to their preceding and subsequent contexts are too similar to providing distinguishing information. To address this problem, we investigate in learning language representation by LMs to making elegant choice of long and short

Proceedings of the 19th China National Conference on Computational Linguistics, pages 874-880,
Hainan, China, October 30 - November 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

forms. This paper makes the following contributions: (1) propose long and short form choice models by making use of language modeling approaches LSTM-RNN LM and pre-trained LM (BERT(Devlin et al., 2018) and ERNIE(Sun et al., 2019) (2) to compare the performance of different LMs, constructing a well-designed test set for long and short form choice task.

The remainder of this paper is organized as follows. In Section 2, we discuss related work. Section 3 describes the language modeling methods we have used for our research and introduce our models. Section 4 presents our experimental results. We conclude with a discussion in Section 5.

2 Related work

A lot of words can be expressed by either a long form or a short form(Packard, 2000), for instance, elastic word, abbreviation, reduplication. In this work, we focus on the choice of long and short form of elastic words, that is, to choose between the long form (disyllabic) and short form (monosyllabic) of an elastic word that shares one morpheme and at least one same word meaning, and are interchangeable in some contexts(Duanmu and Dong, 2016). Previous work(Guo, 1938; Duanmu, 2013; Duanmu and Dong, 2016; Huang and Duanmu, 2013) show that as high as 90% Chinese word has long and short forms, which is a key issue in Chinese lexical choice. Li et al.(2019) investigated the problem of long and short form choice through human and corpus-based approaches, whose results support the statistical significant correlation between word length and the predictability of its context. Most previous work investigate the distribution and preference of long and short form based on corpus. It is still an open question to automated choose between long and short forms for a given context.

We framed the choosing between long and short forms as a FITB task proposed by Edmonds (1997) in English near-synonyms choice. Unsupervised statistical approaches were applied to accomplish FITB task in near-synonym choice, for instance, Co-occurrence Networks(Edmonds, 1997) and Pointwise Mutual Information (PMI)(Inkpen, 2007) were used to build up near-synonym choice model separately. Wang and Hirst(2010) explore lexical choice problem by capturing high dimensional information of target words ant their contexts thorough Latent Semantic Space.

Language models have obtained excellent performance in many language processing tasks, thus they have been also used to tackle the lexical choice task. A 5-gram language model(Islam and Inkpen, 2010) was trained from a large-scale Web corpus to choosing among English near-synonyms, following which Yu et al.(2011) implemented n-gram language model to Chinese near-synonym choice. N-gram model shows a better accuracy than PMI in near-synonym choice which is similar to our task. Neural Language Models overcome the limitation of n-gram language model by its powerful capability of long-range dependency. Recurrent Neural Networks (RNN)(Mirowski and Vlachos, 2015) and it variation Long-short Term Memory (LSTM)(Tran et al., 2016). Zweig et al.(Zweig et al., 2012) tackled the sentence completion problem with various approaches like language models. NNLMs achieved a better performance in these work, whose improvement can be attributed to its capability of capturing global information.

3 Long and Short Form Choice via Language Models

Language modeling is an effective approach to solve the task by computing occurrence probability of each candidate words. Given a context, the best long and short form can be chosen according to the probability acquired from language models. The state-of-the-art language modeling techniques and apply them to our task is described in this section.

3.1 N-gram Language Model

An input sentence S contains n words, i.e.,

$$S = \{w_1 w_2 w_3 ... w_i ... w_{n-2} w_{n-1} w_n\} \quad (1)$$

CCL2020

Proceedings of the 19th China National Conference on Computational Linguistics, pages 874-880,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

where w_i (i^{th} word of the sentence), denotes the lexical gap. The candidate words for the gap is $w_i = \{w_{long}, w_{short}\}$. Our task is to choose the w_i that best matches with the context.

N-gram language model, a classical probability language model, has succeeded in many previous work(Zweig et al., 2012; Yu et al., 2011; Islam and Inkpen, 2010) by capturing contiguous word associations in given contexts. A n-gram smoothed model(Islam and Inkpen, 2010) for long/short word choice is used as our baseline model, whose key idea of acquiring the probability of a string is defined as follow:

$$P(S) = \prod_{i=1}^{p+1} P(w^i|w_{i-n+1}^{i-1}) = \prod_{i=1}^{p+1} \frac{C(w_{i-n+1}^{i}) + M(w_{i-n+1}^{i-1})P(w_i|w_{i-n+2}^{i-1})}{C(w_{i-n+1}^{i-1}) + M(w_{i-n+1}^{i-1})} \tag{2}$$

$$M(w_{i-n+1}^{i-1}) = C(w_{i-n+1}^{i-1}) - \sum_{w_i} C(w_{i-n+1}^{i}) \tag{3}$$

where p is the number of words in the input sentence, i is the word position, $C(w_{i-n+1}^{i})$ and $C(w_{i-n+1}^{i-1})$ denotes the occurrence of the n-gram in the corpus, $P(w_i|w_{i-n+2}^{i-1})$ is the probability of w_i occurs given the words w_{i-n+1}^{i-1}, missing count $M(w_{i-n+1}^{i-1})$ is defined as 2.

The lexical gap of the input sentence S is replaced by long and short form separately, as follow:

$$S_1 = \{w_1 w_2 w_3 ... w_{long} ... w_{n-2} w_{n-1} w_n\}$$
$$S_2 = \{w_1 w_2 w_3 ... w_{short} ... w_{n-2} w_{n-1} w_n\}$$

Equation 1 is used to calculate $P(S_1)$ and $P(S_2)$, and take the target word in the sentence with higher probability as result. A disadvantage of n-gram model is not capable of maintaining long distance dependencies that play important role on long/short word choice. Hence, we proposed a neural language model to accomplish our task.

3.2 Recurrent Neural Networks (RNNs) Language Model

N-gram LM assigns probabilities to sentences by factorizing their likelihood into n-grams, whose modeling ability is limited because of data sparsity and long-distance dependency problem. NNLM have been proposed to model NL bymikolov2010recurrent, and outperform N-gram LM in many tasks(Mirowski and Vlachos, 2015; Tran et al., 2016) due to its ability of (1) each word w is represented as a low-dimensional density vector (2) retain long-span context information, which is failed captured by n-gram language model.

Recurrent Neural Networks (RNNs) have shown impressive performances on many sequential modeling tasks, thus we hypothesize that the performance of long/short form choice can be improved by adopting RNNs LM. Training a RNNs LM is difficult because of the vanishing and exploding gradient problems. Several variants of RNNs have been proposed to tackle with these two problems, among which Long Short-Term Memory is one of the most successful variants. In this work, we employ LSTM-RNNLMs to solve long/short form choice question. The LSTM adopted in this work is described as follows:

$$i_t = \sigma(U_i x_t + W_i s_{t-1} + V_i c_{t-1} + b_i)$$
$$f_t = \sigma(U_f x_t + W_f s_{t-1} + V_f c_{t-1} + b_f)$$
$$g_t = f\left(U x_t + W s_{t-1} + V c_{t-1} + b\right)$$
$$c_t = f_t \bigodot c_{t-1} + i_t \bigodot g_t$$
$$o_t = \sigma\left(U_o x_t + W_o s_{t-1} + V_o c_t + b_o\right)$$
$$s_t = o_t \cdot f\left(c_t\right)$$
$$y_t = g\left(V s_t + M x_t + d\right)$$

Proceedings of the 19th China National Conference on Computational Linguistics, pages 874-880,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

where x_t is input vector and y_t is output vector at time step t, i_t, f_t, o_t are input gate, forget gate and output gate respectively. c_{t-1} is the internal memory of unit, s_{t-1} is the LSTM hidden state at the previous time step. The uppercase (e.g., U_i and W) are weight matrices, the lowercase (e.g., b_i and b) is bias. f is the activation function and σ is the activation function for gates. The symbol $\odot$ is the Hadamard product or element-wise multiplication. Because of the architecture of LSTM-RNNLMs, the model has the potential to model long-span dependency.

3.3 Pre-trained Language Models

Language modeling aims to predict a distribution over a large scale of vocabulary items, by which solving the long/short form choice is a hard objective for our LSTM-RNNs acquired by limited size of training set and computation resource. We have a implicit assumption that the use of a powerful pre-trained language model is helpful to our task. Large-scale language models have achieved great success in many different Natural Language Understanding tasks. In this work, we focus on tackle our research question two very largely publicly LMs BERT and ERNIE.

LSTM-RNN LMs usually use the n preceding words as input to predict the next word $n+1$, which cannot capture subsequent words of the word $n+1$. BERT tackle this problem by retaining information of all the words in some fixed-length sequence. Thus, we re-implemented BERT as a long and short form predictor to assign probability for a target word in a given context. BERT' s model architecture is a multi-layer bidirectional Transformer encoder, whose success can be largely attributed to its Multi-Head Attention mechanism. By the attention mechanism, BERT is able to solving problems by learning the best representation through computing a weighted sum of the values of all words. The BERT-Base Chinese model adopted in this work is trained on a large scale of Chinese Simplified and Traditional corpus (based on an architecture of 12 layers, 768 hidden units, 12 heads, and 110M parameters). We tested the Bert with the methodology we used to test LSTM-RNNs.

ERNIE is a knowledge integration language representation model for Chinese, whose language representation is enhanced by using entity-level and phrase-level masking strategies in addition to a basic-level masking strategy. ERNIE has the same model structure as BERT-base, which uses 12 Transformer encoder layers, 768 hidden units and 12 attention heads.

4 Experiments and Results

Our baseline is a smoothed 4-gram language model, described in section 3.1. In our training data set, we keep the words occurring at least 50 times, and filter out 2-gram, 3-gram, and 4-gram that occur less than three times. For the model based on LSTM-RNN LM, we set the word embeddings as 300, the LSTM hidden states as 128, sentence max length as 50, and learning rate as 0.1.

4.1 Data Resources

A large scale corpus is used in this work, which is Chinese online news in June 2012 (approximately contains 64M Chinese words)[1]. We split the corpus into two parts: 90% of the corpus is used for training and 10% for testing. The same training set is employed to train the 5-gram LM and LSTM-RNN LM, which ensure the comparability of these two models.

To test our models, we carefully construct a test set based on the corpus. Firstly, we randomly choose 175 different long/short forms from. Then, 6 sentences for each of these long/short forms are extracted from the corpus, in which the sentences contain the same number of long and short forms. Finally, we get a test set by slightly editing these sentences manually, which consists of 1050 sentences.

[1]https://www.sogou.com/labs/resource/cs.php

Proceedings of the 19th China National Conference on Computational Linguistics, pages 874-880,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

4.2 Results

Table 2 summarizes our results tested by the identical test set, which shows that all our models based on NNLMs approaches perform better than the baseline model. The improvement in accuracy of LSTM-RNN is 3.43%; the accuracy has been improved 10.96% by adopting BERT; and ERNIE performs the best in our task whose accuracy reaches 82.67%. Our results show that NNLMs is more capable than Ngram LM in long and short form choice task. We think our model based on LSTM-RNNs LM is not as well-performed as the two pre-trained NNLMs is because of its simpler neural network architecture and a smaller training set.

4.3 Post-hoc Analysis

According to semantic relation of the two morphemes of long forms, the long and short forms can be categorized into 7 groups(Li et al., 2019). The X-XX category refers to reduplicated long and short forms such as 妈妈-妈 *(mama-ma, mother) or* 仅仅-仅 (jinjin-jin, only). All our models perform very well in predicting X-XX especially 5-gram LM performing the best, which suggests that the local context makes more contribution to the reduplication form choice than to other categories. Comparing with other categories of long and short forms, our models based on LSTM-RNN and ERNIE obviously perform bad in X-0X category, whose accuracy of this X-0X[2] is significant lower than the average accuracy (20.00% and 14.33% respectively). We think this is due to the comparatively low frequency of X-0X according to observation of our train set for LSTM-RNN LM.

Method	5-gram	LSTM-RNN	BERT	ERNIE
X-X'X	60.67%	77.33%	82.67%	88.00%
X-X0'	59.33%	78.00%	82.67%	78.67%
X-XY	62.67%	73.33%	82.67%	90.67%
X-0'X	66.67%	75.33%	75.33%	86.00%
X-XX	96.67%	88.00%	84.67%	87.33%
X-0X	71.33%	53.33%	76.00%	68.00%
X-X0	72.00%	68.00%	82.00%	80.00%
Accuracy	69.90%	73.33%	80.86%	82.67%

Table 2: Accuracy of language modeling methods tested by identical data set.

5 Conclusion

In this paper, we have investigated methods for answering long short form choice question. This question is significant because it is a key aspect of lexical choice which is still not well solved by many language processing systems. Through this work, we find that both all NNLM-based models do obviously outperform than Ngram LM. And our results show that all models perform very well in X-XX category but not very well in X-0X category. Our future work will be in the direction of eliminating the bias from NNLMs. Human evaluation for long and short form choice models also will be our further research content.

6 Acknowledgements

The first author of this paper received support from Qinghai Natural Science Foundation under Grant 2016-ZJ-931Q, Qinghai Major R&D Transformation Foundation under Grant 2019-GX-162, and National Natural Foundation under Grant 61862055, which is gratefully acknowledged.

[2]X-0X refers the long and short form like 小麦-麦 *(xiaomai-mai, wheat)*

References

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

San Duanmu. How many chinese words have elastic length. *Eastward flows the Great river: Festschrift in honor of Prof. William S.-Y. Wang on his 80th birthday*, pages 1–14, 2013.

Philip Edmonds. Choosing the word most typical in context using a lexical co-occurrence network. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 507–509. Association for Computational Linguistics, 1997.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Diana Zaiu Inkpen and Graeme Hirst. Near-synonym choice in natural language generation. In *Recent Advances in Natural Language Processing*, volume 3, pages 141–152, 2004.

Toan Q Nguyen and David Chiang. Improving lexical choice in neural machine translation. *arXiv preprint arXiv:1710.01329*, 2017.

Geoffrey Zweig, John C Platt, Christopher Meek, Christopher JC Burges, Ainur Yessenalina, and Qiang Liu. Computational approaches to sentence completion. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 601–610. Association for Computational Linguistics, 2012.

Jerome L Packard. *The morphology of Chinese: A linguistic and cognitive approach*. Cambridge University Press, 2000.

Shaoyu Guo. the function of elastic word length in Chinese. *Yen Ching Hsueh Pao*, 24:1–34, 1938.

San Duanmu and Yan Dong. Elastic words in chinese. *The Routledge Encyclopedia of the Chinese Language*, pages 452–468, 2016.

Lijun Huang and San Duanmu. a quantitative study of elastic word length in modern Chinese. *Linguistic Sciences*, 12(1):8–16, 2013.

Yan Dong. *The prosody and morphology of elastic words in Chinese: annotations and analyses*. PhD thesis, University of Michigan, 2015.

Lin Li, Kees van Deemter, Denis Paperno, and Jingyu Fan. Choosing between long and short word forms in mandarin. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 34–39, 2019.

Philip Edmonds. Choosing the word most typical in context using a lexical co-occurrence network. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 507–509. Association for Computational Linguistics, 1997.

Tong Wang and Graeme Hirst. Near-synonym lexical choice in latent semantic space. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1182–1190. Association for Computational Linguistics, 2010.

Liang-Chih Yu, Wei-Nan Chien, and Shih-Ting Chen. A baseline system for chinese near-synonym choice. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1366–1370, 2011.

Aminul Islam and Diana Inkpen. Near-synonym choice using a 5-gram language model. *Research in Computing Sciences*, 46:41–52, 2010.

Mary Gardiner and Mark Dras. Predicting word choice in affective text. *Natural Language Engineering*, 22(1):97–134, 2016.

Piotr Mirowski and Andreas Vlachos. Dependency recurrent neural language models for sentence completion. *arXiv preprint arXiv:1507.01193*, 2015.

Ke Tran, Arianna Bisazza, and Christof Monz. Recurrent memory networks for language modeling. *arXiv preprint arXiv:1601.01272*, 2016.

CCL2020

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, 2019.

Dilin Liu. Is it a chief, main, major, primary, or principal concern?: A corpus-based behavioral profile study of the near-synonyms. *International Journal of Corpus Linguistics*, 15(1):56–87, 2010.

Diana Inkpen. A statistical model for near-synonym choice. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):1–17, 2007.

Sun, Yu and Wang, Shuohuan and Li, Yukun and Feng, Shikun and Tian, Hao and Wu, Hua and Wang, Haifeng Ernie 2.0: A continual pre-training framework for language understanding arXiv preprint arXiv:1907.12412,2019.

Devlin, J., Chang, M. W., Lee, K., Toutanova, K. Pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805,2018.

Mikolov, M. Karafi´at, L. Burget, J. Cernock´y, and S. Khudanpur. Recurrent neural network based language model. In INTERSPEECH, pages 1045–1048, 2010.

CCL2020

Proceedings of the 19th China National Conference on Computational Linguistics, pages 874-880,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China