Neural Networks for Negation Cue Detection in Chinese

Hangfeng He[1] **Federico Fancellu**[2] **Bonnie Webber**[2]
[1]School of Electronics Engineering and Computer Science, Peking University
[2]ILCC, School of Informatics, University of Edinburgh
hangfenghe@pku.edu.cn, f.fancellu@sms.ed.ac.uk, bonnie@inf.ed.ac.uk

Abstract

Negation cue detection involves identifying the span inherently expressing negation in a negative sentence. In Chinese, negative cue detection is complicated by morphological proprieties of the language. Previous work has shown that negative cue detection in Chinese can benefit from specific lexical and morphemic features, as well as cross-lingual information. We show here that they are not necessary: A bi-directional LSTM can perform equally well, with minimal feature engineering. In particular, the use of a character-based model allows us to capture characteristics of negation cues in Chinese using word-embedding information only. Not only does our model performs on par with previous work, further error analysis clarifies what problems remain to be addressed.

1 Introduction

Negation cue detection is the task of recognizing the tokens (words, multi-word units or morphemes) inherently expressing negation. For instance, the task in (1) is to detect the negation cue "不(not)", indicating that the clause as a whole is negative.

(1) 所有住客均表示不会追究酒店的这次管理失职
(All of guests said that they would **not** investigate the dereliction of hotel.)

Previous work has addressed this task in English as a prerequisite for detecting negation scope (Fancellu et al., 2016; Cruz et al., 2015; Zou et al., 2013; Velldal et al., 2012; Zhu et al., 2010). But recently, the release of the CNeSp corpus (Zou et al., 2015) allows allows the task to be addressed in Chinese as well. Detecting negation cues in Chinese texts is difficult because character cues can be homographs of or contained within words not expressing negation. For instance, "非常(very)" and "未来(future)" are not negation cues, while "非(not)" and "未(not)" are. Moreover, even expressions that contain a negation cue may not correspond to clause-level negation, because the overall meaning of the expression is positive. This can be observed in the expression "非要", roughly corresponding to the English expression "couldn't help but/had to" which contains the negation cue "非", but which carries a positive meaning where the action indeed take place, as in:

(2) ...，到了后非要2 0 0元，...
...when we are arriving, they **had to** charge 200 yuan...

Finally, negation cues in Chinese are similar to English affixal cues (e.g. "*in*sufficient"), where they become integral part with the word they modify (e.g. 够("sufficient") → 不够("insufficient")). According to the CNeSp guidelines, both the negation affix and the root it attaches to are considered as part of the cue. The high combinatory power of negation affixes leads however to issues of data sparsity. This is particularly relevant in the context of the CNeSp corpus, given that about 12% of negation in the test set is not present in the training set (Zou et al., 2015, p. 660).

Specifically, using the CNeSp corpus, Zou et al. (2015) tried to automatically detect negation cues using a sequential classifier trained on a variety of features, including lexical (word n-grams), syntactic (PoS n-grams) and morphemic features (whether a character has appeared in training data as part of a cue). In addition, to address the problem of affixal negation cues producing tokens in the test set that did not appear in the training set, Chinese-to-English word-alignment was also

Proceedings of the Workshop Computational Semantics Beyond Events and Roles (SemBEaR), pages 59–63,
Valencia, Spain, April 4, 2017.

taken into account.

In contrast, the recent success of Neural Network models for negation scope detection (Fancellu et al., 2016) suggested investigating whether a character-based recurrent model can perform on par or better than this previous work. After describing our model in Section 2, we show in Section 3.3 that a character-level representation with no feature engineering is able to achieve similar *recall* as models that use word-alignment information, as well as other features, to tackle the problem of data sparsity. Compared to other sequence classifiers however, we show that neural networks tend to overpredict negation cues (thereby damaging *precision*) and suffer from insufficient training data.

2 The model

2.1 Input

We define a negative sentence as one that contains at least one negation cue. Given a sentence $ch = ch_1...ch_{|c|}$, we represent each character $ch_i \in ch$ as a d-dimensional character-embedding vector $ch_i^e \in \mathbb{R}^d$.

We define E_{ch}^{vxd} as the character-embedding matrix, where v is the vocabulary size. To represent a character along with its surrounding context in absence of any word segmentation, the input to the network is the concatenation of the current character ch_i with its neighboring characters in a fixed window size 2*m+1. Our input instance will therefore be the concatenation of a given character plus its m proceeding and m succeeding characters as follows, $ch_{i-m}^e...ch_{i-1}^e; ch_i^e; ch_{i+1}^e...ch_{i+m}^e$.

2.2 BiLSTM Neural Network

The model we are going to use for this task is a Bi-LSTM model. Similar to RNNs, these models are able to leverage long-distance relations to predict whether a character is part of a negation cue or not. LSTM have however the advantage of better retaining information when backpropagating the error. On top of this, the bi-directionality allows to process the input left-to-right and viceversa, allowing for the entire sentential context to be taken in consideration at prediction time.

The inner computation of the LSTM network is as follows:

$$i_t = sigmoid(W_{ix}ch_t + W_{ih}h_{t-1} + b_i)$$

$$f_t = sigmoid(W_{fx}ch_t + W_{fh}h_{t-1} + b_f)$$

$$o_t = sigmoid(W_{ox}ch_t + W_{oh}h_{t-1} + b_o)$$

$$c_t = f_t * c_{t-1} + i_t * tanh(W_{cx}ch_t + W_{ch}h_{t-1} + b_c)$$

$$h_t = o_t * tanh(c_t)$$

where W are the weight matrices, i_t, f_t, o_t and c_t are the input, forget, output gate and cell state at position t, b the bias vector and h_t the hidden state representation at time t. The prediction of label y_t is computed as:

$$y_t = softmax(W_{hy}[h_t^{forw}; h_t^{back}] + b_y) \quad (1)$$

where W_{yh} is the output layer weight matrix and $[h_t^{forw}; h_t^{back}]$ the concatenation of the hidden states as computed during the forward and backward pass.

2.3 Transition Probability

Although the bi-LSTM keeps an internal memory of the inputs previously visited, the predictions made are independent from each other. For this reason, we introduce a new joint model $p(s|ch)$, defined as:

$$p(s|ch) = \prod_{i=1}^{n} p(s_i|s_{i-1}, ch)$$

The only functional change to the original LSTM model is the addition of a 4-parameter transition matrix to create the dependence on s_{i-1}, enabling the use of standard inference algorithms. This enables us to train the model end-to-end.

3 Experiments

3.1 Data

We use the Chinese Negation and Speculation (CNeSp) corpus (Zou et al., 2015) in our experiments. It is divided into three sub-corpora: Product reviews (below as *product*), Financial Articles (*financial*) and Computer-related Articles (*scientific*). (Corpus statistics appear in Table 1.) We first train and test on each corpus separately. We use a fixed 70%/15%/15% split of these in order to define a fixed development set for error analysis, but this setup precludes direct comparison to with (Zou et al., 2015), since they used 10-fold cross-validation. Nevertheless, we felt a data analysis was crucial to understanding these systems, and we wanted a clear distinction between test (for reporting results) and development (for analysis). For completeness, we also show results on training and testing when all corpora are joined together.

Models	Financial Article			Product Review			Scientific Literature			All		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Baseline-Word	25.09	68.37	36.70	33.18	76.31	46.25	12.06	77.42	20.87	24.01	74.40	36.31
Baseline-Char	29.82	82.79	43.84	32.73	75.96	45.75	14.50	**93.55**	25.11	24.28	76.00	36.80
BiLSTM-*char*	61.94	71.16	66.23	78.93	**87.46**	**82.98**	**64.71**	35.48	**45.83**	69.08	**84.00**	75.81
+ Bigram	**65.15**	**73.02**	**68.86**	**79.05**	86.76	82.72	25.00	9.68	13.95	**71.70**	80.80	**75.98**
+ Transition	58.57	68.37	63.09	78.57	86.24	82.23	47.83	35.48	40.74	69.08	82.74	75.30

Table 2: Results on development set for each of the CNeSp subcorpora.

Models	Financial Article			Product Review			Scientific Literature			All		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Zou et al. (2015)	**72.77**	67.02	**69.78**	81.94	89.23	85.43	**75.17**	78.91	**76.99**	-	-	-
Baseline-Word	24.76	66.52	36.09	30.93	72.47	43.36	12.32	**83.33**	21.46	22.13	71.68	33.82
Baseline-Char	28.66	**78.11**	41.94	33.41	78.75	46.91	12.32	**83.33**	21.46	23.68	77.89	36.32
BiLSTM-*char*	62.92	64.81	63.85	85.02	**91.99**	**88.37**	20.83	16.67	18.52	70.50	82.24	75.92
+ Bigram	63.41	66.95	65.14	**85.06**	91.29	88.07	7.14	3.33	4.55	**73.83**	80.25	76.90
+ Transition	63.08	70.39	66.53	84.56	89.72	87.07	14.29	10.00	11.76	72.49	**82.48**	**77.16**

Table 3: Results on test set for each of the CNeSp subcorpora.

	Sentence Number	Cue Number
Financial	6550	1461
Product	4969	3914
Scientific	4626	171

Table 1: Details of the three CNeSp subcorpora.

3.2 Settings

We experimented with three different settings:

- Character (*char.*): the input is a *single* character embedding, concatenated with the embeddings of its neighboring characters in a window size *m*.
- Character-bigram (*bigram*): the input is character *bigram* embedding obtained by the concatenation of the embeddings of two adjacent characters. We concatenate a bigram embedding with the embeddings of the neighboring character bigrams in a window size *m*. This reflects the observation that most negation cues are bigrams.
- Transition: a transition-based component is applied on top of the network (§2.3)

Our model is trained using stochastic gradient descent with L2 regularization. Learning rate is 0.01 with decay rate 0.95, *m* is 2 to yield a window size of 5; character embedding dimension and feature embedding dimension are both 100, discount κ in margin loss is 0.2, and the hyper parameter for the $L2$ is 0.000001.

Baseline. To understand the difficulty of cue detection, we designed two naive baselines based on a list of all negation cues contained in the training data: 1) *Baseline-Word*, where we classify as negation cue a character or a span of characters if it appears on the list, and 2) *Baseline-Char*, where we first segment the test sentence[1] and consider a word as cue if it contains any element on the list.

3.3 Results

Results on the development and test sets are shown in Tables 2 and 3 respectively. Both baselines achieves low precision compared to a higher recall which indicates that the challenge of this task lies in not overpredicting the negation cue span. A comparison of our models shows that character bigram information does not contribute to better performance, nor does the transition based component. Interpreting the poor performance on the *scientific* set is however difficult since there are only 171 cues in 4262 sentences, and only 12 in the 463 test sentences, a sample too small to draw any conclusion.

Table 3 also shows that neural network models with minimal feature engineering perform on par or better than the highly engineered sequential model used by Zou et al. (2015). Their higher recall show that they capture more negation cues, which is important, given that the approach does not use any cross-lingual alignment information to deal with test cues not seen during training. Finally, the results of the *scientific* test set show the same problem of small sample size as with the development set.

[1]For the segmentation we used the NLPIR toolkit: `https://github.com/NLPIR-team/NLPIR`

4 Error Analysis

4.1 Financial articles

Most of the errors in the *financial* sub-corpus are under-prediction errors. For instance, in the sentence (3), our model predicts "不景" as the negative cue, which is the under-prediction of "不景气".

(3) ...,受经济**不景气**影响,...
(...,influenced by the economic **depression**,...)

In order to tackle this problem we carried out a small experiment where we post-process the results. We first used the NLPIR toolkit to automatically segment the sentence and if the detected cue is part of a word, then the whole word is considered as cue. The under-prediction error shows that the word segmentation information may be important in negation cue detection. When we apply this heuristic to the financial sub-corpus, we only noticed however only a small improvement across all measures as shown in Table 4.

	Precision	**Recall**	**F1**
Original	65.15	73.02	68.86
Post Process	**66.39**	**74.42**	**70.18**

Table 4: Difference between before and after post process in financial sub corpora

4.2 Product Review

Amongst the wrong predictions (121 in total) for the *Product Review* corpus, there are 61 sentences for which we predict more negative cues than gold one. These errors concern the most frequent negative cues such as "不(not)" and "没(not)". For instance, as shown in (4), our best model predicts "不(not)" as cue, which is different with the gold one.

(4) 房间设施一般，网速**不**仅慢还经常断网。
(The room facilities are common and the network **not** only is slow but also often disconnect.)

These errors show that even expression that contain a negation cue may not correspond to clause-level negation. We also hypothesized that these wrong predictions are due to the fact that our model are not fed any explicit syntactic or semantic information regarding the context of a given character. Future work could explore the possibility of augmenting the input with extra information such as part of speech tags.

5 Conclusions and Future Work

In the present paper we addressed the problem of automatically detecting the negation cue in Chinese. In particular, we investigated whether character - based neural networks are able to achieve on par or better performance than previous highly engineered sequence classifiers. Results confirm that these models can be a valid alternative to previous ones, although still suffering from overgenerating the negation cue. In the process, we also found that one of the corpora we tested with might not be suitable to be used on its own, given the lack of enough instances.

Given the positive results obtained for Chinese, future work should focus in testing the method in English as well.

Acknowledgments

This project was funded in part by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644402 (HimL). We also would like to thank the School of Electronic Engineering and Computer Science, Peking University for their support of Hangfeng He's internship at the University of Edinburgh.

The authors would like to thank the three anonymous reviewers for their comments.

References

Noa P. Cruz, Maite Taboada, and Ruslan Mitkov. 2015. A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology*.

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 495–504.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers, and the role of syntax. *Computational linguistics*, 38(2):369–410.

Qiaoming Zhu, Junhui Li, Hongling Wang, and Guodong Zhou. 2010. A unified framework for

scope learning via simplified shallow semantic parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 714–724. Association for Computational Linguistics.

Bowei Zou, Guodong Zhou, and Qiaoming Zhu. 2013. Tree kernel-based negation and speculation scope detection with structured syntactic parse features. In *EMNLP*, pages 968–976.

Bowei Zou, Qiaoming Zhu, and Guodong Zhou. 2015. Negation and speculation identification in chinese language. In *ACL (1)*, pages 656–665.