INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A Chinese Dataset for Identifying Speakers in Novels

Jia-Xiang Chen, Zhen-Hua Ling, Li-Rong Dai

National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, P.R. China
cjx123@mail.ustc.edu.cn, {zhling,lrdai}@ustc.edu.cn

Abstract

Identifying speakers in novels aims at determining who says a quote in a given context by text analysis. This task is important for speech synthesis systems to assign appropriate voices to the quotes when producing audio books. Several English datasets have been constructed for this task. However, the difference between English and Chinese impedes processing Chinese novels using the models built on English datasets directly. Therefore, this paper presents a Chinese dataset, which contains 2,548 quotes from *World of Plainness*, a famous Chinese novel, with manually labelled speaker identities. Furthermore, two baseline speaker identification methods, i.e., a rule-based one and a classifier-based one, are designed and experimented using this Chinese dataset. These two methods achieve accuracies of 53.77% and 58.66% respectively on the test set.

1. Introduction

Dialogue, representing linguistic and social relationships between characters, is an important component of literature. Detecting quotes in novels is a necessary preprocessing step for understanding dialogues and has been studied in previous work [1, 2]. Mention detection methods [3, 4] have been proposed to help extract characters from novels since character analysis is fundamental to literary analysis, forming the basis of many computational studies in literary domains. Identifying speakers in novels aims at determining who says a quote in a given context by text analysis. This task is also important for text-to-speech synthesis. Given the character information of quotes, speech synthesis systems can assign appropriate voices to quotes in order to improve the intelligibility and expressiveness of synthetic audio books.

Several datasets with manually labelled characters for quotes in novels have been constructed in previous work. Elson and McKeown [5] took important first step towards automatic quote attribution. Their dataset, the Columbia Quote Speech Corpus (CQSC), is the most well-known dataset and was used by lots of studies on speaker identification in novels. He et al. [6] proposed a dataset based on the novel *Price and Prejudice*. The complete novel was annotated by a student of English literature. Muzny et al. [7] released QuoteLi3 (Quotes in Literary text from 3 novels) dataset, which combined the data from the above two datasets. There are also some other related datasets [8, 9]. Based on these datasets, some methods of identifying speakers in novels have been proposed, including rule-based ones [6, 10, 11], and machine learning-based ones [5, 12, 13]. However, all these existing datasets and studies were based on English novels. Considering the difference between English and Chinese, it is difficult to apply them to process Chines novels directly. Therefore, this paper presents a Chinese dataset for identifying speakers in novels.

This dataset was built by five steps using a Chinese novel *World of Plainness*. This novel is famous and lots of quotes are contained in its contents. Totally 2,968 quotes were extracted from the raw text of this novel. After removing the quotes which can not be assigned to a specific character after manual annotation, 2,548 quotes were kept to construct the final dataset.

Two baseline speaker identification methods, i.e., a rule-based one and a classifier-based one, were implemented using this Chinese dataset. The rule-based method was implemented based on the one proposed by Glass and Bangay [11]. In the classifier-based method, linguistic features were extracted from quotes and their contexts and a multi-layer perceptron (MLP) [14] model was built in a data-driven way. These two methods achieved accuracies of 53.77% and 58.66% respectively on our test set. Furthermore, an analysis was conducted by dividing the instances in the dataset into three categories according to whether the character metioins were explicit.

2. Related Work

In this paper, a quote means a sentence which is uttered by a character. A speaker is the character speaking the quote. A mention of a character means a span of texts which appears in the quote or its context and corresponds to an alias of the character.

The Columbia Quoted Speech Corpus [5] is a dataset that includes both quote-mention and quote-speaker labels. This dataset contains 11 novels and 3,176 instances totally. However, this dataset suffers from the disagreement of annotators on about 35% quotes [5]. The dataset built by He et al. [6] includes high-quality speaker labels. There are 1,901 instances in this dataset. And this dataset assumes that all quoted text within a paragraph should be attributed to the same speaker. While this assumption is correct for the novel *Pride and Prejudice*, it may be incorrect for other novels. Muzny et al. [7] combined the datasets of Elson and McKeown [5] and He et al. [6] to create a new one. The combined dataset covers 3 novels and 3,103 individual quotes. It is composed of expert-annotated dialogue from Jane Austen's *Pride and Prejudice* and *Emma*, and Anton Chekhov's *The Steppe*. Because every quote has a label for speaker and a label for mention, there are total 6,206 labels, more than 3,000 of which are newly annotated.

Pareti et al. [12] proposed an approach to identify speaker which focused on identifying spans associated with content (i.e., quotes), source (i.e., mentions), and cues (e.g., speech verbs) in newswire data. In the literary domain, Glass and Bangay [11] did early work by adopting a rule-based method to identify speakers in fiction novels. Elson and McKeown [5] made important first step towards automatic quote attribution. They formulated the task as mention identification in which the goal was to link a quote to a mention of its speaker. They constructed a single feature vector for each pair of an utterance and a speaker candidate, and experimented with various WEKA

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-1614

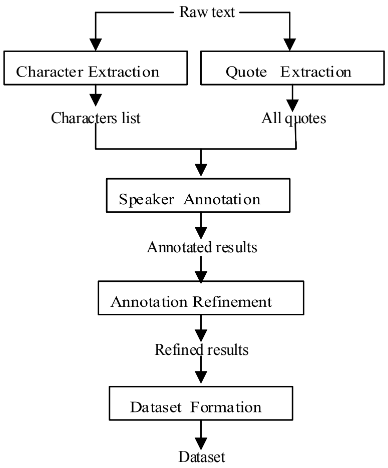

Figure 1: *The flowchart of dataset construction.*

classifiers and score combination methods. O'Keefe et al. [10] also treated quote attribution as mention identification, using a sequence labelling approach. Their approach was successful in the news domain but it failed to beat their baseline in the literary domain. This work quantitatively indicated that quote attribution in literature was fundamentally different from the task in newswire.

Text-To-Speech synthesis has made rapid progresses in recent years [15, 16]. The single speaker and reading style speech synthesis systems are able to achieve similar naturalness to human recordings [17]. For the application of synthesizing audio books, the expressiveness of synthetic speech still needs improvement. The speaker labels extracted for the quotes in novels can help speech synthesis systems to assign appropriate voices to different quotes and to improve the expressiveness of synthetic speech.

3. Dataset Construction

A famous Chinese novel, *World of Plainness*, was chosen to build our dataset. We understand that the research results based on this dataset may not be generalized to other novels. Considering the difficulty of identifying speakers by text analysis, focusing on one novel to start with is a reasonable choice. To build dataset containing multiple novels and to develop novel-independent speaker identifying algorithms will be our future work. This novel has over 1,040,000 Chinese word tokens and the percentage of tokens belonging to quotes is about 11.83%. In order to examine if this ratio is reasonable, we also collected a text corpus of 240 Chinese novels of 12 genres (20 novels per genre). The statistical results on this corpus show that this percentage is around 8% to 20% and varies with genres of novels.

Figure 1 shows the flowchart of constructing our dataset from raw text. This section will introduce the five main steps in this figure in detail.

3.1. Character Extraction

In this step, we extracted all characters and their aliases and combined them to construct a global character list. The global character list was used to detect mentions of characters in text. When annotating each quote manually, these mentions were utilized to make a list of character candidates for annotators. The global character list was also employed when generating the list of character candidates for model training. This step simplified the task by making it focus on finding the relationship between a quote and a speaker candidate rather than finding all possible characters automatically.

At the beginning, we planned to collect some comments from book review websites to find all characters. However, most comments just talk about 6-10 main characters and it is not enough to find all characters according to comments only. Then, we chose to extract all characters manually. There are total 126 characters who speak at least one quote. A character may has different aliases in this novel. The last name of a person is usually ignored in Chinese (e.g., "Shaoping" for "Shaoping Sun"). Another instance is that some nicknames are decided by social relations (e.g., "Sister Runye" for "Runye Tian"). In some cases, the position of a person is also a part of his or her nickname (e.g., "Director Tian“ for "Futang Tian"). Therefore, each item in the character list was defined as $\{$*character index, gender label, name_1, ...,name_N*$\}$, where *name_1, ...,name_N* denote *N* aliases of this character in the novel.

When extracting characters and their aliases, we also extracted each character's gender and all speech verbs appearing in this novel. Such information can help to extract features for speaker identification according to previous study [11].

3.2. Quote Extraction

The next step was to extract all quotes in the novel. Simple regular expressions were designed to achieve this goal. Every quote is surrounded by double quotation marks in the raw text of this novel. However, some onomatopoeia and specific words are also surrounded by double quotation marks. Fortunately, in the novel *World of Plainness*, these onomatopoeia and specific words don't contain punctuation. For example, there is no punctuation in "boom", while each quote ends with punctuation. Thus, we can filter out these words by regular expressions easily. Totally, 2,968 quotes were extracted from this novel.

3.3. Speaker Annotation

In this step, the speaker identity of each quote was labelled manually. Two annotators with rich experiences in labelling Chinese texts took part in the annotation.

The format of an instance showed to annotators was $\{$*quote,context,character list*$\}$, where context consisted of the center quote and 10 sentences before and after center quote respectively, and the characters list denoted the characters appearing in the context of altogether 21 sentences. This local character list was derived by matching aliases of all characters in the novel towards the context. We provided the local character list to annotators in order to help them find the correct speakers easily. However, annotators were not constrained to make decisions only based on the information within the context window considering that there was external information, such as the plots obtained when processing previous quotes, that can help annotators to find correct speakers. Preliminary experiment showed that using a context window of 21 sentences can achieve a satisfactory balance between the time consumption of annotation and the ratio of quotes which can be labelled successfully.

Two annotators were asked to label all the 2,968 instances using a webpage-based interface. Three choices were given to them. If they can find the speaker in the local character list, they just chose it directly. If they can determine the speaker but it was not contained by the local character list, they were required to write down the name of the speaker. A *NONE* label was assigned to the quote if annotators failed to attribute the center quote to a specific character.

Finally, the first annotator labelled 2,559 quotes with specific speakers among all 2,968 quotes, and the second annotator labelled 2,457 quotes. For each annotator, the remaining quotes were labelled as *NONE*. These two annotators gave the same labels to 2,803 quotes and the consistency rate was 94.44%.

3.4. Annotation Refinement

We assumed that if the two annotators assigned the same character to a quote, the label should be reliable and needed no further refinement. Examining the results of the speaker annotation step, 2,413 quotes satisfied this condition. Then, the first author of this paper examined the remaining quotes manually to determine the correct speakers of them. Finally, 2,682 quotes were labeled with a character in the global character list, and the remaining 286 quotes were labelled as *NONE*.

3.5. Dataset Formation

During manual speaker annotation, a context window of 21 sentences was used. Since annotators were not constrained to make decisions only based on the information within the context window, the quotes whose correct answer can't be found within the context window should be excluded from the final dataset. We evaluated 3 different window sizes (i.e., 5, 10, or 20 sentences before and after each quote respectively) and counted how many quotes whose speaker mentions appeared within the context window. Among the 2,682 quotes obtained after annotation refinement, the ratios of resolvable quotes were 92.3% 95.0% and 97.0% for window sizes of 11, 21, and 41 respectively. Finally, the context window of 21 sentences (i.e., 10 sentences before and after each quote) was chosen and this led to a final dataset of 2,548 quotes.

Each instance in the dataset was in the form of {*quote, context, index list of character candidates, correct character index*}. The index list of character candidates was generated by the same way used in Section 3.3. The average number of character candidates for each quote was 2.94. We split the dataset into training, development, and test sets with 2,000, 274, and 274 instances respectively. [1]

It should be noticed that the numbers of quotes spoken by different characters formed an unbalanced distribution in our dataset. Figure 2 shows the quote frequencies in descending order. We can see that almost 80% quotes belong to the first 30 characters. We believe that such long-tail distribution is common for other novels and this increases the difficulty of building models to identify the speakers of quotes in novels.

4. Baseline Methods and Results

4.1. Rule-Based method

Two baseline methods of speaker identification were implemented using our dataset. The first one was developed following the rule-based method proposed by Glass and Bangay [11]. This method first determines four items for each quote, Actor, Best, Next, and Last speaker. **Actor** denotes the possible speaker of the quote. For each character candidate, a score is calculated by combining its speech verb information, the distance between the center quote and the sentence which contains its mention, the count of its mentions within the context window, etc. Then, the character candidate with the highest score is chosen as Actor. **Best** means the character with most mentions within the context window. **Next** means the character with second most mentions within the context window. **Last speaker** stands for the speaker of previous quote decided by the rule-based method.

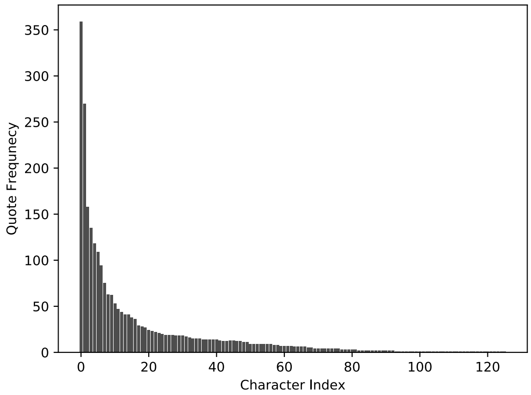

Figure 2: *The numbers of quotes corresponding to different characters in our dataset.*

Then, the rules designed by Glass and Bangay [11] are applied to find the speaker of the quote based on the four items and the list of character candidates. In our implementations, we revised some rules by considering the conversation pattern and punctuation information of a quote. Here, conversation pattern denotes if the sentence before the quote is also a quote. Punctuation information indicates if the sentence before the quote ends with a colon, or the sentence following the center quote ends with a period. The details are skipped here due to limited space.

4.2. Classifier-Based Method

The second method adopts a data-driven approach by training a classifier. For every character in the candidate list, linguistic features are extracted from the quote and context sentences. Then, these features are fed into a 2-layer multi-layer perceptron with 10 units at the hidden layer to output a matching score. The scores of all character candidates are further normalized by a softmax layer. The cross-entropy loss on training set is minimized to train the whole model. The features used in this method are listed as follows.

1. *quote_index* the line index of the quote in the whole novel.
2. *in_quote_count* the count of character mentions appearing within the quote sentences in the context. Here, quote sentences means all quote sentence, not only the center quote.
3. *out_quote_count* the count of character mentions appearing within the non-quote sentences in the context.
4. *character_in_quote* a binary flag indicating whether the character appears in the center quote. It is very rare that the speaker name of a quote appears within the quote.
5. *nsubj_count* the count of character mentions appearing as subject in the context.
6. *obj_count* the count of character mentions appearing as object in the context.

[1] https://github.com/chenjiaxiang/Chinese-dataset-for-speaker-identification.

Table 1: *Speaker identification accuracies (%) of using different methods on test set.*

method	rule	classifier	random
accuracy	53.77	58.66	35.12

7. *main_verb_count* the count of verbs which are associated with the character and are main verbs in dependency trees.
8. *speech_verb_count* the count of verbs appearing in the list of speech verbs. The fifth to eighth features are extracted based on dependency trees generated by the Stanford CoreNLP toolkit [18].
9. *gender* the gender of the character.
10. *female_count* the count of "she" appearing in the context.
11. *male_count* the count of "he" appearing in the context.
12. *conversation_pattern* a binary flag indicating if the sentence before the quote is also a quote.
13. *quote_pattern* a binary flag indicating whether the nearest mention sentence is a quote. Here, the nearest mention sentence means the nearest sentence to the quote which contains a mention of the character.
14. *sentence_distance* the distance between the nearest sentence which contains mention of candidate and the center quote.
15. *full_stop_pattern* a binary flag indicating whether the nearest mention sentence ends with a period.
16. *colon_pattern* a binary flag indicating whether the nearest mention sentence ends with a colon.

4.3. Results

The speaker identification accuracies of the two methods on test set are showed in Table 1. Random method means we choose a character randomly from the candidate list as the result for each instance in the test set. From this table, we can see that the data-driven approach outperformed the rule-based one by about 5% absolute accuracy. However, the overall accuracies of both methods were still low and improvements are necessary in future work.

4.4. Analysis

In order to further analyze our dataset and experimental results, the instances in the dataset are divided into three categories with #1 *explicit*, #2 *implicit*, and #3 *latent* labels respectively. Each instance is processed with following steps to get its category.

Step 1 If the sentence before the center quote ends with a colon, or the sentence following the center quote ends with a period, pass this instance to Step 2. Otherwise, it is classified as Category #3.

Step 2 If any verb in the sentence is a speech verb, its subject is determined according to the dependency tree. Otherwise, this instance is classified as Category #3.

Step 3 If the subject is the mention of a character, the instance is classified as Category #1. If the subject is pronoun, the instance is classified as Category #2. Otherwise, it is classified as Category #3.

For Category #1 and #2, category-specific methods are designed to obtain the results of speaker identification. For Category #1, the character mention is explicit, i.e., the subject found by Step 3 introduced above. For Category #2, a coreference resolution algorithm is applied to map the pronoun determined by Step 3 to a character mention within the context window. In our implementation, the Stanford CoreNLP toolkit [18] was utilized for coreference resolution.

Table 2: *Speaker identification accuracies (%) of using category-specific methods, rule-based method and classifier-based method on three categories of test instances.*

category	#1	#2	#3
specific	92.16	29.03	-
rule	84.31	35.48	52.08
classifier	65.95	45.75	57.28

Finally, the numbers of instances belong to these three categories were 480, 280, and 1788 respectively for our whole dataset. These numbers were 51, 31, and 192 for our test set. We evaluated the speaker identification accuracies of using the category-specific methods, the rule-based method, and the classifier-based method on these three categories of test instances. The results are shown in Table 2. We can see that the test instances of Category #1 can be solved well using the category-specific method introduced above. This is reasonable since their character mentions are explicit with speech verbs in surrounding sentences. In contrast, the accuracy of applying general coreference resolution algorithm to the test instances of Category #2 was far from satisfying. When using the rule-based and classifier-based methods introduced in Section 4.1 and 4.2 for identification, Category #1 achieved the best accuracies among the three categories but they were still not as good as the category-specific method. These two methods obtained better performance for Category #2 than coreference resolution. This implies that some task-dependent algorithms should be developed to dealing with this category. Comparing the rule-based method with the classifier-based method, we can see that the former achieved better accuracy on Category #1 but performed worse on the other two categories than the latter. More than 70% percent of instances in our dataset belong to Category #3. To further improve the classifier-based method for handling this category will be a task of our future work.

5. Conclusion

This paper have presented a Chinese dataset for identifying speakers in novels considering the lack of Chinese resources for this task nowadays. Two baseline speaker identification methods, i.e., a rule-based one and a classifier-based one, have been implemented using this dataset. The classifier-based one has achieved better accuracy on test set in our experiments. Some further analysis has also been conducted by dividing the instances into three categories and designing category-specific methods. To increase the size of the dataset by adding more novels and to introduce state-of-the-art neural network-based NLP models into this task will be our future work.

6. Acknowledgements

This work was supported by the National Key R&D Program of China (Grant No. 2017YFB1002202) and the National Nature Science Foundation of China (Grant No. 61871358). The authors would like to acknowledge iFlytek Research for help annotate the dataset.

7. References

[1] C. Scheible, R. Klinger, and S. Padó, "Model architectures for quotation detection," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1736–1745.

[2] B. Pouliquen, R. Steinberger, and C. Best, "Automatic detection of quotations in multilingual news," in *Proceedings of Recent Advances in Natural Language Processing*, 2007, pp. 487–492.

[3] H. Ji and D. Lin, "Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection," in *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, vol. 1, 2009.

[4] D. Bamman, T. Underwood, and N. A. Smith, "A bayesian mixed effects model of literary character," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 370–379.

[5] D. K. Elson and K. McKeown, "Automatic attribution of quoted speech in literary narrative." in *AAAI*, 2010.

[6] H. He, D. Barbosa, and G. Kondrak, "Identification of speakers in novels," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 1312–1320.

[7] G. Muzny, M. Fang, A. Chang, and D. Jurafsky, "A two-stage sieve approach for quote attribution," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, vol. 1, 2017, pp. 460–470.

[8] D. J. A. P. D. R. Hardik Vala, Stefan Dimitrov, "Annotating characters in literary corpora, a scheme, the charles tool,and an annotated novel," *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, 2016.

[9] S. Pareti, "A database of attribution relations." in *LREC*. Citeseer, 2012, pp. 3213–3217.

[10] T. O'Keefe, S. Pareti, J. R. Curran, I. Koprinska, and M. Honnibal, "A sequence labelling approach to quote attribution," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 790–799.

[11] K. Glass and S. Bangay, "A naive salience-based method for speaker identification in fiction books," in *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA'07)*, 2007, pp. 1–6.

[12] S. Pareti, T. O'Keefe, I. Konstas, J. R. Curran, and I. Koprinska, "Automatically detecting and attributing indirect quotations," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 989–999.

[13] E. Iosif and T. Mishra, "From speaker identification to affective analysis: A multi-step system for analyzing children's stories," in *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, 2014, pp. 40–49.

[14] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[15] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[16] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[17] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to human quality TTS with transformer," *arXiv preprint arXiv:1809.08895*, 2018.

[18] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010