A Neural-Network-Based Approach to Identifying Speakers in Novels

Yue Chen$^{1,}$, Zhen-Hua Ling$^{1,}$, Qing-Feng Liu$^{1,2}$

$^{1}$National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei, P. R. China  
$^{2}$iFLYTEK Co., Ltd., Hefei, P. R. China

cy321@mail.ustc.edu.cn, zhling@ustc.edu.cn, qfliu@iflytek.com

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 audiobooks. However, existing approaches stick with manual features and traditional machine learning classifiers, which constrain the accuracy of speaker identification. In this paper, we propose a method to tackle this challenging problem with the help of deep learning. We formulate speaker identification as a scoring task and build a candidate scoring network (CSN) based on BERT. Candidate-specific segments are put forward to eliminate redundant context information. Moreover, a revision algorithm is designed utilizing the speaker alternation pattern in two-party dialogues. Experiments have been conducted using the dataset built on the Chinese novel World of Plainness. The results show that our proposed method reaches a new state-of-the-art performance with an identification accuracy of 82.5%, which outperforms the baseline using manual features by 12%.

Index Terms: speaker identification, BERT, dialogue analysis, speech synthesis

1. Introduction

Identifying speakers in novels is an important task for many downstream applications, such as assigning appropriate voices to utterances when producing audiobooks [1, 2], and creating scripts based on novels [3]. As dialogues serve as an important mean of shaping characters in literature, automatic identification of speakers can also be useful for text mining tasks like extracting the social network of characters [4, 5].

Existing speaker identification methods can be divided into rule-based ones and machine-learning-based ones. Rule-based methods [6, 7] focused on designing linguistic rules so as to make decisions among speaker candidates. These methods heavily relied on the knowledge of developers and usually resulted in low accuracy and poor generalization ability. On the other hand, machine-learning-based methods [8–11] utilized manually-labelled training data to build classifiers, such as support vector machines (SVM) [12] and multi-layer perceptrons (MLP) [13], to determine the speakers of quotes. However, all these studies still adopted hand-crafted linguistic features, which were designed heuristically and failed to describe the input text in a comprehensive way.

In recent years, numerous studies in natural language processing (NLP) have demonstrated that neural-network-based feature extraction is far more effective than manual feature engineering [14, 15], since it can jointly optimize the feature extractor and the classifier, and can therefore learn globally optimal features. Deep neural architectures like recurrent neural networks (RNN) [16] and convolutional neural networks (CNN) [17] have been popularly applied to various NLP tasks and outperformed the traditional machine learning models. More recently, pretrained language models, such as BERT [18] and GPT [19], have achieved state-of-the-art performance on many shared tasks, owing to their attention-based architectures and knowledge obtained from a massive amount of pretraining data.

Therefore, this paper proposes a neural-network-based approach to solve the task of identifying speakers in novels. We formulate this task as a scoring problem and build a candidate scoring network (CSN) to calculate the score for each candidate speaker. In consideration of the limited size of annotated data, the pretrained architecture for natural language understanding, BERT [18], is adopted as the foundation of CSN. To reduce redundant context information, candidate-specific segments are extracted from quote and context sentences, and are sent into BERT to derive the representations of candidates, contexts, and quotes respectively. Then, these representations are concatenated and fed into an MLP scorer to obtain the score for each candidate speaker. The candidate with the highest score is determined as the speaker of the quote. To further improve model performance, a revision algorithm based on the confidence measures given by CSN is designed utilizing the speaker alternation pattern (SAP) in two-party dialogues. Experimental results show that our proposed method achieved an accuracy of 82.5% on the World of Plainness dataset which outperformed the machine-learning-based baseline method using manual features by 12%.

The main contributions of this paper are twofold. First, we build a candidate scoring network (CSN) based on BERT to tackle the problem of speaker identification in novels. Second, we design a revision algorithm utilizing the confidence measures given by CSN and the speaker alternation pattern in two-party dialogues.

2. Methodology

2.1. Task Definition

In our task, an instance for speaker identification is a textual segment consisting of a number of sentences. We denote it as $I = s_{-w_s} \oplus \cdots \oplus s_{-1} \oplus qs \oplus s_1 \oplus \cdots \oplus s_{w_s}$ and $\oplus$ is the concatenation operation. $qs$ is the quote sentence whose speaker needs to be identified. $\{s_{-w_s}, \cdots, s_{-1}\}$ are the $w_s$ context sentences on the left of $qs$, and $\{s_1, \cdots, s_{w_s}\}$ are the ones on the right. $w_s$ is the single-sided context window size.

In addition to instances, a name list is provided which contains the characters that occur throughout the novel and a varying number of aliases for each character. For each instance in the dataset, the true speaker of the quote sentence has been

---

The source code of this paper can be obtained from https://github.com/YueChenkki/CSN-SAPR
annotated manually. Further, we assume that at least one alias of the true speaker in the name list should appear within the context sentences of the instance. Otherwise, the instance should be discarded from the dataset.

Given these prerequisites, our task is to determine the speaker of the quote sentence in an instance given the name list of the novel.

2.2. Framework of Proposed Method

The flowchart of our proposed method is shown in Fig. 1 which consists of three main steps. First, an instance is sent into the nearest mention location (NML) module together with the name list to determine a candidate set for this instance and to extract a candidate-specific segment from the instance for each candidate. Second, each candidate-specific segment is fed into the candidate scoring network (CSN) to produce its score. At last, a speaker alternation pattern (SAP)-based revision is applied to the quote sentences that belong to two-party conversations before finally determining the predicted speakers. The details of these three steps will be introduced in the following subsections.

2.3. Nearest Mention Location (NML)

This module first generates a set of candidate speakers for the input instance. For each character in the name list, if any one of its aliases appears within the context sentences of the instance, this character is added to the candidate set.

For each candidate speaker, there may be more than one occurrence (i.e., mentions) of its aliases. Intuitively, we assume that the mention nearest to the quote sentence carries the most evidence about whether the candidate is the true speaker. Therefore, we locate the nearest mention for each candidate which has the least number of words between itself and the quote sentence. Assuming that the nearest mention locates in the context sentence \( s_{nm} \) for a candidate speaker, its candidate-specific segment (CSS) is defined as

\[
CSS = \begin{cases} 
  s_{nm} \oplus \cdots \oplus s_{-1} \oplus q_{s}, & \text{if } nm < 0, \\
  q_{s} \oplus s_{1} \oplus \cdots \oplus s_{nm}, & \text{if } nm > 0.
\end{cases}
\]

(1)

The purpose of extracting candidate-specific segments is to exclude redundant context sentences which may not contribute to judge the correctness of a candidate speaker. In addition to candidate-specific segment, we also define candidate-specific context as \( CSS \backslash q_{s} \), i.e., removing quote from the candidate-specific segment.

2.4. Candidate Scoring Network (CSN)

Since the number of candidates for an instance varies, we regard speaker identification as a scoring task instead of a standard classification task. That is, a candidate scoring network (CSN) is built which assigns a score for every candidate speaker of an instance, and the candidate with the highest score is the result.

The structure of our proposed CSN is shown in Fig. 2. We first adopt the widely used pretrained language model, BERT [18], to generate contextualized representations. At the encoding stage, we feed the candidate-specific segment into BERT and obtain a sequence of hidden states with the same length as the input token sequence. The hidden states corresponding to the nearest candidate mention, the quote sentence, and the candidate-specific context are denoted as \( H_{wcm} \), \( H_{qs} \), and \( H_{csn} \) respectively. Then, each of \( H_{wcm} \), \( H_{qs} \) and \( H_{csn} \) goes through a max-pooling layer and produces the embedding vector for candidate representation, quote sentence representation, and context representation respectively. These three fixed-length representations are concatenated to form the feature vector of this candidate. The feature vector is then fed into an MLP scorer with tanh output activation to produce a score within \((-1, 1)\).

At the training stage, a margin ranking loss is adopted since our goal is to assign high scores to true speakers and low scores to distractors. For an instance \( I \), its true speaker \( c_{1} \) is paired with another candidate \( c_{2} \) to form a positive-negative example pair. Then, the loss of this pair is calculated as

\[
L(I, c_{1}, c_{2}) = \max \{0, csn(I, c_{1}) - csn(I, c_{2}) + mgn\},
\]

(2)

where \( csn(I, c) \) denotes the score of candidate \( c \) calculated by the CSN and \( mgn \) is a hyper-parameter that controls the ideal score margin between the two candidates. During training, the parameters in CSN are optimized to minimize the overall loss on the training set.

2.5. Speaker-Alternation-Pattern-Based Revision

Continuous multi-turn conversations are common in novels and most of them occur between two speakers. However, the context sentences of a quote in a conversation may seriously overlap with other quotes in the conversation, and thus the CSN may not be able to learn useful context representations for scoring candidates. Therefore, this module aims to revise the speaker identification results of CSN for two-party conversations utilizing the speaker alternation pattern (SAP). SAP means that the speaker of the \( n^{th} \) utterance in a two-party conversation is usually the speaker of the \( (n + 2)^{th} \) utterance, but is not the speaker of the \( (n + 1)^{th} \) utterance \([6, 9, 20]\). Based on this pattern, once the speaker of an utterance in the dialogue is determined, the speakers of the rest utterances can be easily inferred.

In this paper, we automatically detect two-party conversations in our dataset following two conditions. First, a conversation should be composed of continuous quote sentences without interrupting context sentences. Let \( q_{s1} \oplus \cdots \oplus q_{sm} \) denote a conversation with \( M \) quotes and \( M \geq 2 \). Second, a two-party conversation should have two dominant candidate speakers. Let \( c_{1} \) denote the candidate speaker which has the \( n^{th} \) highest mention frequency \( f_{1} \) in the \( 2m \) context sentences of the conversation. This conversion is considered as a two-party one if \( f_{2} \geq f_{3} + th \), where \( th \) is a pre-set threshold. Thus, \( c_{1} \) and \( c_{2} \) become the two candidate speakers of all quotes in this context as \( CSS \backslash q_{s} \), i.e., removing quote from the candidate-specific segment.
SAP-based revision algorithm is shown in Algorithm 1. The complete conversation. At the test stage, if the quote of an instance is in a two-party conversation, SAP-based revision is applied. Its basic idea is to first determine the speaker of the first quote or the last quote in a two-party conversation based on their confidence measures considering that the context sentences of these two quotes have the least overlap with other quote sentences. For the $m$-th quote in a conversation, its confidence measure is calculated as

$$q_{m}.cm = |csn(I_m, c_1) - csn(I_m, c_2)|,$$  
(3)

where $I_m$ is the instance with $q_m$ as the quote sentence, $c_1$ and $c_2$ are the two candidate speakers of the conversation. A higher confidence measure indicates more reliable scores given by the CSN. Then, the speakers of the other quotes in the conversation are determined one by one according to SAP. The complete SAP-based revision algorithm is shown in Algorithm 1.

### Algorithm 1 SAP-based revision

**Input:** a conversation of $M$ quotes $q_1 \oplus \cdots \oplus q_M$, candidate speakers $c_1$ and $c_2$.

$q_{1}.cm = |csn(I_1, c_1) - csn(I_1, c_2)|$;
$q_{M}.cm = |csn(I_M, c_1) - csn(I_M, c_2)|$;  

If $q_{1}.cm > q_{M}.cm$ then

$q_{1}.speaker = \arg \max_{c \in \{c_1, c_2\}} csn(I_1, c);$  

for $i = 2; i \leq M; i + +$ do

if $q_{i-1}.speaker = c_1$ then

$q_{i}.speaker = c_2$;

else

$q_{i}.speaker = c_1$;

end if

end for

else

$q_{M}.speaker = \arg \max_{c \in \{c_1, c_2\}} csn(I_M, c);$  

for $i = M - 1; i \geq 1; i --$ do

if $q_{i+1}.speaker = c_1$ then

$q_{i}.speaker = c_2$;

else

$q_{i}.speaker = c_1$;

end if

end for

end if

Table 1: The numbers of instances in our dataset.

<table>
<thead>
<tr>
<th></th>
<th>explicit</th>
<th>implicit</th>
<th>latent</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>393</td>
<td>220</td>
<td>1387</td>
<td>2000</td>
</tr>
<tr>
<td>Development</td>
<td>44</td>
<td>31</td>
<td>223</td>
<td>298</td>
</tr>
<tr>
<td>Test</td>
<td>44</td>
<td>29</td>
<td>225</td>
<td>298</td>
</tr>
</tbody>
</table>

The Chinese BERT-base model released by Google Research\(^3\) was adopted to build our CSN. For a text input, BERT output a contextualized representation and the embedding size was 768. The MLP scorer in CSN had a hidden layer of 100 units with tanh activation. The margin in Eq. (2) was set as $mgn=1.0$. The mention frequency threshold in SAP-based revision was $3$.  

\(^2\)https://github.com/YueChenkkk/Chinese-Dataset-Speaker-Identification  
\(^3\)https://github.com/google-research/bert

### 3. Experiments

#### 3.1. Dataset

The dataset built by Chen et al. [10] was adopted. This dataset was constructed based on the famous Chinese novel World of Plainness with human annotations. Its context window size (i.e., $ws$) was 10. The name list was collected manually and contained 125 roles that occurred throughout the novel, and each role had a varying number of 1-5 aliases. We extended the original dataset by making additional annotations and obtained 2596 instances in total\(^2\). We kept their original temporal order in the novel, and divided them into a training set with 2000 instances, a development set with 298 instances, and a test set with 298 instances. Following the taxonomy described in Chen’s paper [10], we further divided the subsets into 3 categories named explicit, implicit and latent respectively. For a brief introduction, an instance in explicit or implicit can find the subject of a speech verb within the adjacent two sentences of the center quote. If that subject is a mention of a candidate in the name list, the instance belongs to explicit. If the subject is a pronoun, the instance belongs to implicit. The latent category holds instances that belong to neither explicit nor implicit. The numbers of instances in all categories are listed in Table 1.

#### 3.2. Settings

The Chinese BERT-base model released by Google Research\(^3\) was adopted to build our CSN. For a text input, BERT output a contextualized representation and the embedding size was 768. The MLP scorer in CSN had a hidden layer of 100 units with tanh activation. The margin in Eq. (2) was set as $mgn=1.0$. The mention frequency threshold in SAP-based revision was 1.0.
3.3. Experimental Results

The speaker identification accuracies of our proposed method on the test instances of three categories are shown in the third row of Table 2. In this table, Random Guess means selecting the speaker from candidates randomly. Manual+MLP is the baseline method described in Chen et al. [10] which was based on manually-designed linguistic features and an MLP scorer. We can see that our proposed method outperformed the baseline method described in Chen et al. [10] which was based on manually-designed linguistic features and an MLP scorer.

The learning rate and batch size were set as 16 and 2e-5. The optimal iteration was determined based on the development set performance.

Ablation studies were conducted to further analyze the effectiveness of the modules in our proposed method. The first ablation was to remove the SAP-based revision and the results are shown in the fourth line of Table 2. By comparing the third and the fourth rows, we can see that this ablation didn’t affect the performance of explicit and implicit categories, and degraded the accuracy of the latent category. The reason is that the quote sentences of explicit and implicit instances in two-party conversations were usually the first or last quotes, thus they were not aimed to be revised. Actually, there were 20 latent instances in the test set which changed their decisions after the SAP-based revision. Among them, 16 revisions amending the wrong decisions of CSN, while 4 instances were incorrectly revised given the correct decisions of CSN. Furthermore, Table 3 shows the average confidence measures given by CSN for the first and last quotes in two-party conversations. It can be seen that the first and last quotes in conversations had significantly higher confidence measures than the other quotes ($p=0.0027$ in $t$-test). This is consistent with our assumption for designing the SAP-based revision algorithm that the quotes in the middle of conversations may not have as reliable scores as the first and last quotes in conversations.

The second ablation further discarded candidate-specific segments at the encoding step, which means it directly fed the whole instance into BERT to derive the representations of candidate, context, and quote sentence. We name it candidate-irrelevant encoding. After this ablation, the average character-level length of BERT input declined from 534.4 to 125.7 which was close to that of using candidate-specific segments (111.7). For each instance, its candidate set was the same as that of $ws=10$. If a candidate didn’t have any mention in the truncated context window, a score of zero was assigned to him or her by default. Finally, this approach yielded an overall accuracy of 0.698 without SAP-based revision, which was still much lower than that of using candidate-specific segments. These results indicate the necessity of filtering out redundant context information and modeling different context sentences for scoring different candidates in our neural-network-based approach.

3.4. Case Study

Fig. 3 presents a test instance with the nearest mentions of its 4 candidates are underlined and numbered. The quote and other context sentences are omitted.

![Case Study](image)

Finally, this approach yielded an overall accuracy of 0.698 without SAP-based revision, which was still much lower than that of using candidate-specific segments. These results indicate the necessity of filtering out redundant context information and modeling different context sentences for scoring different candidates in our neural-network-based approach.

4. Conclusion

In this paper, we have proposed a deep-learning-based method to tackle the problem of identifying speakers in novels. Compared with conventional methods using manually designed linguistic features, our BERT-based model can utilize textual inputs more effectively and lead to better accuracy of speaker identification. In the future, more sophisticated mechanisms will be explored to model the interaction between the quote and its context. Besides, the approaches based on other techniques such as speaker modeling in dialogues [22, 23] and social network extraction [4, 5] are also worth further investigation.
5. References


