PAPER

Fuzzy Matching of Semantic Class in Chinese Spoken Language Understanding

Yanling LI[†,††a)], ***Nonmember***, **Qingwei ZHAO**[†], ***Member***, ***and*** **Yonghong YAN**[†], ***Nonmember***

SUMMARY Semantic concept in an utterance is obtained by a fuzzy matching methods to solve problems such as words' variation induced by automatic speech recognition (ASR), or missing field of key information by users in the process of spoken language understanding (SLU). A two-stage method is proposed: first, we adopt conditional random field (CRF) for building probabilistic models to segment and label entity names from an input sentence. Second, fuzzy matching based on similarity function is conducted between the named entities labeled by a CRF model and the reference characters of a dictionary. The experiments compare the performances in terms of accuracy and processing speed. Dice similarity and cosine similarity based on TF score can achieve better accuracy performance among four similarity measures, which equal to and greater than 93% in F1-measure. Especially the latter one improved by 8.8% and 9% respectively compared to q-gram and improved edit-distance, which are two conventional methods for string fuzzy matching.

key words: *fuzzy matching, Conditional Random Field (CRF), Spoken Language Understanding (SLU), Named Entity Recognition (NER), similarity function*

1. Introduction

Voice search is the technology underlying many spoken dialog systems that provide users with the information they request with a spoken query [1]. With the improvement of Automatic Speech Recognition (ASR), the technology of spoken dialog systems has developed vigorously. Spoken dialog systems have been attracting extensive interest from the research and industrial communities since they provide a natural interface between human and hardware devices such as computer or mobile phone, which has such potential benefits as remote or hands-free access, ease of use, naturalness, and greater efficiency of interaction.

A typical spoken dialog system is composed of automatic speech recognition (ASR), and spoken language understanding (SLU), dialogue management (DM) and text-to-speech (TTS). The task of SLU is to map a user's utterance to the corresponding semantics. Thus, the performance of spoken dialog systems not only relies on the accuracy of recognition achieved by ASR, but also the semantic entity of sentence accomplished by SLU. SLU technologies in these systems range from understanding predetermined phrases through fixed grammars, extracting some predefined named entities, extracting users' intents for call classification, to combinations of users' intents and named entities [2].

The study of SLU surged in the 1990's, with the DARPA sponsored Air Travel Information System (ATIS) project [3], which was designed to provide flight information service. Traditional approaches of SLU can be divided into three categories, which are knowledge-based approaches, data-driven approaches and approaches that combine the above two approaches. Knowledge-based approaches in most SLU systems use grammar-based parsers to extract key semantic information, which translate a sentence into a parse tree [4]. Data-driven approaches include hidden Markov model (HMM), i.e. AT&T's CHRONUS [5]; model based on Probabilistic Context-Free Grammar (PCFG), i.e. BBN's hidden understanding model (HUM) [6]; model proposed by He and Young [7] based on hidden Vector State (HVS) and statistic machine translation model [8]. An approach combined with these two categories makes full use of advantage of them [9].

Although the study of SLU focuses only on specific domain, including music/video management [10], business and product reviews [11] etc., it still faces great challenges. One of them is robustness problem. Robustness problems may include three components. The first one is induced by spoken language characteristics, such as false start, self-correction, repetitions and hesitations, ellipsis, out-of-order structures and so on. The second one is words' variations induced by ASR. The third one is the incompleteness of key information, because users usually remember and say the first several words in the listing name but probably to forget or omit the words at the end. Two main approaches to improve the robustness of ASR errors have been proposed [12]. In the first approach, the word sequence hypothesized by the recognizer is decomposed into smaller units under the assumption that acoustically confusable words will have many units in common at the subword level. The second approach uses the recognizer to generate multiple candidate hypotheses from the recognizer rather than just one.

Our SLU system is limited in such domain, and mainly provides for searching services for TV station, website, app and media. This paper pays attention to the robust problems about variations of ASR and the incompleteness of key information induced by users. We propose a new approach to label semantic concepts, which can correct the error of key semantic class based on only part information

Manuscript received November 20, 2012.
Manuscript revised February 21, 2013.

[†]The authors are with Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing, 100190 China.

[††]The author is with College of Computer and Information Engineering, Inner Mongolia Normal University, Hohhot, 010022 China.

a) E-mail: liyanling@hccl.ioa.ac.cn

DOI: 10.1587/transinf.E96.D.1845

in a Chinese sentence. It is divided into two stages. First, a CRF model provides a preprocessor, which spots semantic class of a query and give initial class labels to semantic concepts. Second, if exact matching fails, which means there may be some errors about semantic concept, fuzzy matching can achieve an appropriate Chinese character string to substitute for the wrong ones. We also compare the performance based on some well-known similarity functions.

The structure of this paper is as follows: Sect. 2 described framework of spoken language understanding; Sect. 3 introduced related work about fuzzy matching; CRF to label named entity and several similarity functions are described in Sect. 4; our experiments and results are presented in Sect. 5 and we conclude the paper in Sect. 6.

2. System Architecture of SLU

The main function of SLU is to process text output from ASR and translate the input sentences into formal languages for meaning representation [13]. Our SLU system contains two components, which are semantic class labeling and semantic understanding shown in Fig. 1. Semantic class labeling is to label the key semantic concepts of a sentence, which includes preprocessor and exact/fuzzy match labeling. The goal of preprocessor is to spot the position of key semantic concepts, which is similar to named entity recognition (NER). Because the same class of semantic concepts will share the same contexts. We can solve this problem by the approaches used in NER. Then labeling key semantic concepts is conducted by exact matching with list names. If exact matching fails, it will enter fuzzy matching block. After key semantic concepts are corrected and labeled, semantic understanding transforms a labeled sentence into meaning representation and searches database to give results to users. The precision of semantic class labeling model will influence the performance of semantic understanding model.

Our SLU system is limited in specific domain i.e. TV station, website, app and media. Media component includes the name of people (actor/director/artist) and media (movie, TV series and song). For example, a sentence of "我想看喜羊羊这部电影 (I want to see movie 'Pleasant Goat')". However, in our database, there is only a movie named "喜羊羊与灰太狼 (Pleasant Goat and Wolf)". Thus, only if "喜羊羊 (Pleasant Goat)" is corrected by "喜羊羊与灰太狼 (Pleasant Goat and Wolf)", and mark the class label "movie_Name", it will be possible to find out the right movie name and feed back to users. Therefore, the output of semantic class labeling block is "我想看[喜羊羊与灰太狼]\movie_name 这部电影".

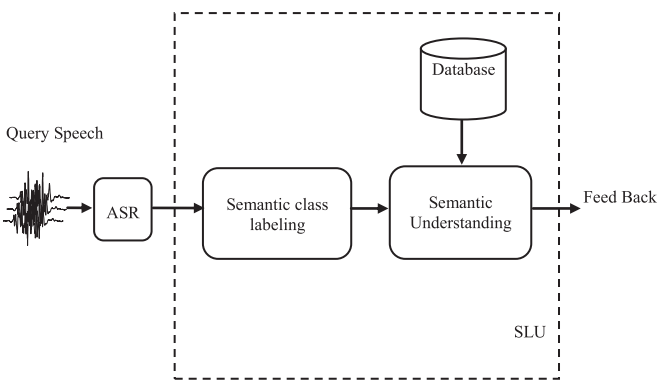

Fig. 1 Architecture of SLU.

3. Related Work about Fuzzy Matching

Fuzzy matching is also named approximate string matching [14]. String matching is a kind of pattern matching problem, which obtains the starting position of substring between an input sentence string and a reference string. Most of methods about approximate string matching based on dictionary or text utilized edit distance as similarity function [15], [16]. Previous work [17]–[19] measures the closeness between two tokens through the similarity between sets of substrings—called q-gram sets—of tokens instead of edit distance between tokens used in fuzzy match methods. Record matching is a well known problem of matching records that represent the same real-world entity and is important procedure in the data cleaning process [20], [21]. Record matching is identifying the customer record in a data warehouse from the customer information such as name and address in a sales record. Those methods are applied to match with determined two words or phrases directly, but not with keywords in a Chinese sentence.

To verify the effectiveness of our method, we will introduce two typical fuzzy matching methods to compare, which are q-gram distance and improved edit distance.

3.1 Q-Gram Distance

Q-gram distance is to find common substrings of fixed length q [22]. Q-gram is that given a string s and a positive integer q, the set of q-gram of s is the set of all size q substrings [18]. We adopt this method in our task to get the substring between the sentence and a reference string. Size of q is from 2 to the length of reference keyword. If there are overlapping substrings, keep the substring with maximum length and delete others. Once substring of sentence is obtained, the similarity measure is computed by Eq. (1).

$$Sim_{q\text{-}gram} = \frac{length(substring)}{length(keyword)} \tag{1}$$

3.2 Improved Edit Distance

Edit distance also named Levenshtein distance [19], which describes the substitution, deletion and insertion times when one string transforms into another string. It is computed based on dynamic programming algorithm. Through filling the whole matrix constructed by two strings, edit distance of two strings is the last element in matrix. Two strings are denoted as $A = (a_1, a_2, \ldots a_n)$ and $B = (b_1, b_2, \ldots b_m)$. We

adopt $C_{i,j}$ to denote each element of matrix and calculation method is as follows. Equations (2) and (3) is an initialization process. Equation (4) is the method to calculate subsequent element of matrix.

$$C_{i,0} = i \tag{2}$$

$$C_{0,j} = j \tag{3}$$

$$C_{i,j} = min(C_{i-1,j-1} + \delta(a_i, b_j), C_{i-1,j} + 1, C_{i,j-1} + 1) \tag{4}$$

Where $\delta(a_i, b_j)$ is zero for $a_i = b_j$, and 1 otherwise. According to the algorithm, matrix of edit distance of two strings such as "我想看富爸爸坏爸爸" and "穷爸爸富爸爸" is shown in Fig. 2. The edit distance between the two strings is 5.

Improved edit distance algorithm is similar to edit distance algorithm, the only difference is initializing the first row of the matrix with zeros. The algorithm causes matching of any string may start at any position in the sentence and filling overall matrix is the same with edit distance algorithm. Figure 3 exemplifies this algorithm applied to search the substring "穷爸爸富爸爸" in the sentence "我想看富爸爸坏爸爸". A minimum edit distance 2 in the last row of matrix is obtained. We detect the matching substring by tracing back along the generation path to the first row. At last, the start position and end position of substring, which has the minimum edit distance with "穷爸爸富爸爸", is acquired.

		我	想	看	富	爸	爸	坏	爸	爸
	0	1	2	3	4	5	6	7	8	9
穷	1	1	2	3	4	5	6	7	8	9
爸	2	2	2	3	4	4	5	6	7	8
爸	3	3	3	3	4	4	4	5	6	7
富	4	4	4	4	3	4	5	5	6	7
爸	5	5	5	5	4	3	4	5	5	6
爸	6	6	6	6	5	4	3	4	5	**5**

Fig. 2 Edit distance matrix between "我想看富爸爸坏爸爸" and "穷爸爸富爸爸".

		我	想	看	富	爸	爸	坏	爸	爸
	0	0	0	0	0	0	0	0	0	0
穷	1	1	1	1	**1**	1	1	1	1	1
爸	2	2	2	2	2	**1**	1	2	1	1
爸	3	3	3	3	3	2	**1**	2	2	1
富	4	4	4	4	3	3	2	**2**	3	2
爸	5	5	5	5	4	3	3	3	**2**	3
爸	6	6	6	6	5	4	3	4	3	**2**

Fig. 3 Improved Edit distance matrix between "我想看富爸爸坏爸爸" and "穷爸爸富爸爸".

The similarity measure is denoted by Eq. (5). "editdis" is the minimum edit distance between the keyword and substring of sentence.

$$Sim_{improved\text{-}editdis} = 1 - \frac{editdis}{length(keyword)} \tag{5}$$

4. Two Stages of Fuzzy Matching

Fuzzy matching methods of q-gram distance and improved edit distance need a sentence directly matching with the reference string in dictionary. Each character of sentence must participate in operation, which induces search space to be too large [19]. Therefore we add CRF probabilistic model to reduce the range of search. This method is divided into two stages described in Fig. 4. First, CRF model preprocesses the sentence, spots semantic class and give initial class labels to semantic concepts. Second, if exact matching fails, fuzzy matching will achieve a maximum Chinese character string to substitute for the wrong ones. We also compare the performance based on some of well-known similarity functions.

4.1 Named Entity Recognition Based on CRF

The NER task is, given a sentence, to segment which words are part of entities, and to classify each entity by type (person, organization, location, and so on). Wang and Acero [23] compared the use of CRF, perceptron, large margin, and minimum classification error (MCE) using stochastic gradient descent (SGD) for sequential labeling problem in the ATIS domain. CRF achieved best performance, though it was the slowest to train.

We use linear-chain CRFs to acquire named entities within limits of semantic class in specific domain. The definition is as follows. Let $W = (w_1, w_2, \ldots w_T)$ be some observed input data sequence, such as a sequence of words in a text document. Let $Y = (y_1, y_2, \ldots y_T)$ be some sequence of states. Linear-chain CRFs define the conditional probability of a state sequence given an input sequence to be [24]:

Fig. 4 Two stages of Semantic class labeling.

$$P_\Lambda(Y \mid W) = \frac{1}{Z(W)} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, W, t)\right) \quad (6)$$

Where $Z(W)$ is a normalization factor over all state sequences, f_k is an arbitrary feature function over its arguments, λ_k is a learned weight for each feature function and y_0 = start, to simplify some expressions.

Then the most probable label sequence can be achieved by maximizing the conditional probability, as shown in Eq. (7).

$$Y^* = \arg\max_Y P(Y \mid W) \quad (7)$$

In addition, decoding algorithm may employ Viterbi algorithm.

4.2 Similarity Function

After the process of NER, semantic class is labeled in a sentence. If named entities of semantic class exist in dictionary or database, corresponding results should be searched and feed back to users. However, in most cases, named entities of semantic class are not in dictionary or database for some errors. There may be three reasons: the recognition errors of CRF model, missing field of key information by users and the variations induced from ASR.

How can we utilize fragment information to correct those errors and feed right answers back? Fuzzy matching may be an indispensable procedure. Choosing which character string in dictionary to substitute for wrong named entity will depend on the similarity between the two named entities. To be compared, we introduce several similarity functions in our previous work [25] for fuzzy matching, and identify the highly similar characters to correct the errors. We use string A to denote named entity recognized by CRF, string B to denote named entities in dictionary.

(a) Cosine Similarity based on TF score

Term frequency-inverse document frequency (TF-IDF) is based on vector space model from the information retrieve domain [26]. We adopt vector space model to compare the similarity between two strings by calculating the cosine similarity of two vectors. To reduce the dimensionality of vector space model, a set of single Chinese character that appear in the string pair is used as a feature set, instead of using indexing character from a dictionary collection [27]. Then term frequencies are used to constitute the vector of string pair and cosine similarity is calculated by Eq. (8).

$$Sim(A, B)_{TF} = \frac{A \cdot B}{|A| \cdot |B|} \quad (8)$$

(b) Jaccard Similarity

Jaccard similarity is defined as the size of the intersection of the Chinese characters in the two strings compared to the size of the union of the characters in the two strings [28].

$$Sim(A, B)_{Jac} = \frac{A \cap B}{A \cup B} \quad (9)$$

(c) Dice Similarity

Dice Similarity is defined as Eq. (10) [28]. $A \cap B$ denotes the number of overlap characters in two strings.

$$Sim(A, B)_{Dice} = \frac{2 \cdot |A \cap B|}{|A| + |B|} \quad (10)$$

(d) Edit distance

Edit distance is described in Sect. 3.2. If two strings are denoted as $A = (a_1, a_2, \ldots a_n)$ and $B = (b_1, b_2, \ldots b_m)$, we can obtain the minimum edit distance by Eq. (2)(3)(4).

However, it is meaningless to compare edit distance directly. It should be normalized to be similarity measure. In Eq. (11), "editdis" represents edit distance between string A and B. The length of string A is n and the length of string B is m.

$$Sim(A, B)_{edit\text{-}distance} = 1 - \frac{editdis}{\max(n, m)} \quad (11)$$

5. Experiments

Experiments are composed of two parts, which are NER based on CRF and fuzzy matching based on similarity functions described above.

5.1 Data Set

The target application behind our work is a voice assistant with function of searching multimedia content. Suppose one needs to search some actor or director's movie, our system will provide the corresponding movie list. For this scenario, we collected and constructed 48905 Chinese sentences to train CRF model and 205 Chinese sentences to test. The test data covers domains including media, app, TV station and website.

In order to evaluate the performance of fuzzy matching method proposed in this paper, we choose 552 Chinese sentences in media domain containing errors to test whole fuzzy matching algorithm. The test data is constructed by some graduates, who analog users to interact with spoken dialog system. The data set is transcribed rather than output of ASR. Among the 552 sentences, about 200 sentences occurring errors with people's name, or media (movie and TV series) name. For example, a sentence of "请帮我找电影麦兜的故事 (Please help me find out 'McDull story')". "麦兜的故事" should be corrected by "麦兜故事". Another situation, there are two semantic concepts in a query such as "我想看周星驰的电影大话西游 (I want to see Xingchi Zhou's film 'Westward Journey')". "周星驰 (Xingchi Zhou)" and "大话西游 (Westward Journey)" should be key semantic concepts. Errors about one of

Table 1 Tag set and its corresponding explanation used in CRF.

Tag set	Explanation
PER-B	Chinese characters in head of Chinese people's name
PER-I	Chinese characters in middle or end of Chinese people's name
NAME-B	Chinese characters in head of names in domain
NAME-I	Chinese characters in middle or end of names in domain
O	others

them or both of them are likely to occur in the test data.

5.2 Experiment of NER

(a) Tag set and features in NER of our specific domain

In this paper, single Chinese character is chosen to be research object in NER [29]. Experiments of [30] shows that choosing single Chinese character to be research object is much better than choosing the tokens after Chinese word segmentation of spoken language data. Named entity in this paper involves the name of Chinese people (actors, directors and singers), Chinese media content (movie, TV series and song), app, TV station and website. In general, words and phrases which appear in similar context usually share similar semantics. Named entities in domain of Media, app, TV station and website in our searching service may share the same context, whose classes are ambiguous. For example, a query is "给我找一下 (help me find out) $name". $name is a variable, which may be substituted by any named entities about media, app, TV station and website. In such sentence pattern, we can't classify the $name into a determined class. Thus, we utilize CRF model to classify semantic class into three kinds, which are people's name, names in media, app, TV station and website (we call them "names in domain" for short) and others. Tag set includes five categories of labels in Table 1.

Considering the characteristic of Chinese people's name, which is composed of surname and lastname, we construct dictionaries of characters commonly used in Chinese surname and lastname. Furthermore, in order to extract Chinese people's name and names in domain more accurately, we collect unigrams and bigrams of Chinese characters, which respectively appear before and after people's name and names in domain. All of atomic templates are in Table 2.

Features of observation sequence are easily added into CRF model to describe dependence on context. We choose the range two to be observed window of context $(w_{-2}, w_{-1}, w_0, w_1, w_2)$. All atomic templates need to shift four seats, which are -2, -1, 1 and 2. These four labels are marked after each atomic template to represent the seats of atomic templates. Features are divided into two components. One is atomic template feature and their four offset seats features. The other is the combination features composed of atomic features. Features selection experiments

Table 2 Atomic template feature list.

No.	Atomic template	Explanation
p0	Curword	Current Chinese character
p1	PersonName	Chinese character of people's name including surname and lastname
p2	Surname	Chinese character of people's surname
p3	Lastname	Chinese character of people's lastname
p4	Prefixper_u	Unigram before people's name
p5	Prefixper_b	Bigram before people's name
p6	Suffixper_u	Unigram after people's name
p7	Suffixper_b	Bigram after people's name
p8	PrefixEntity_u	Unigram before names in domain
p9	PrefixEntity_b	Bigram before names in domain
p10	SuffixEntity_u	Unigram after names in domain
p11	SuffixEntity_b	Bigram after names in domain

Table 3 Combination features list.

No.	combination features
pp1	Curword_-1 && Curword_0
pp2	Curword_0 && Curword_1
pp3	PersonName_-1 && PersonName_0
pp4	PersonName_0 && PersonName_1
pp5	Curword_0 && PrefixEntity_u_-1
pp6	Curword_0 && Prefixper_u_-1
pp7	Curword_0 && Prefixper_b_-1 && Prefixper_b_-2
pp8	Curword_0 && PrefixEntity_b_-1 && PrefixEntity_b_-2
pp9	Curword_0 && Suffixper_b_1 && Suffixper_b_2
pp10	Curword_0 && SuffixEntity_b_-1 && SuffixEntity_b_-2

show that those features in Table 2 and Table 3 are effective in our task.

b) Evaluation of NER

In the experiments below, performances are reported in three metrics (for each Chinese entity): precision, recall and F1-measure.

$$precision = \frac{\#\ No.\ of\ correctly\ recognized\ entities}{\#\ No.\ of\ all\ recognized\ entities} \times 100\% \tag{12}$$

$$recall = \frac{\#\ No.\ of\ correctly\ recognized\ entities}{\#\ No.\ of\ all\ reference\ entities} \times 100\% \tag{13}$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{14}$$

The relationship between experiment results in NER and features are shown in Table 4. Features are added into CRF model in accordance with the number. Feature selection method is very naive, only by observing the performance of experiment. We can see the performance of names in domain and people's name rises above 90% when all of

Table 4 Feature and its corresponding results about people's name, names in domain and overall.

No.	Feature	People's name			Names in domain			overall		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
1	p0	70.30	49.65	58.20	70.73	24.79	36.71	70.42	38.46	49.75
2	p1	67.48	58.04	62.41	69.77	25.64	37.50	68.07	43.46	53.05
3	p2-p7	82.98	81.82	82.40	85.86	72.65	78.70	84.17	77.69	80.80
4	p8-p11	83.57	81.82	82.69	87.39	88.89	88.13	85.33	85.00	85.16
5	pp1-pp2	84.40	83.22	83.81	87.39	88.89	88.13	85.77	85.77	85.77
6	pp3-pp4	85.11	83.92	84.51	87.39	88.89	88.13	86.15	86.15	86.15
7	pp5	85.11	83.92	84.51	88.98	89.74	89.36	86.87	86.54	86.70
8	pp6	85.11	83.92	84.51	90.60	90.60	90.60	87.60	86.92	87.26
9	pp7-pp10	86.01	86.01	86.01	95.73	95.73	95.73	90.38	90.38	90.38

Table 5 The comparision of different fuzzy matching methods in accuracy.

Approach	precision(%)	recall(%)	F1(%)
Q-gram	86.51	84.78	85.64
Improved editdis	85.64	85.33	85.48
CRF+ TF	90.17	96.38	**93.17**
CRF+Jaccard	89.15	95.29	92.12
CRF + Dice	90.00	96.20	93.00
CRF + editdistance	89.49	95.65	92.47

Table 6 The comparision of different fuzzy matching methods in processing speed.

Approach	Total Time(s)	Time/sentence(s)
q-gram	68	0.123
Improved editdis	425	0.770
CRF+ TF	235	0.426
CRF+Jaccard	234	0.424
CRF + Dice	235	0.426
CRF + editdistance	250	0.453

features are added in CRF model, which means all of features are effective to our system. After all of features added in CRF model, the performance of overall F1-measure can arrive at 90.38%.

5.3 Experiments of Fuzzy Matching

To compare the performances in terms of accuracy and processing speed, we carry out experiments of conventional fuzzy matching methods and of the proposed methods in this paper. Precision, recall and F1-measure are used as metrics for evaluation of accuracy. The metrics for the processing speed include the total time of algorithm running and the average time of per sentence processing. Experiment results are shown in Table 5 and Table 6.

In Table 5, we give the performance of accuracy about several different fuzzy matching methods. Improved edit distance and q-gram distance have almost the same performance in F1-measure. Four similarity measures methods combined with CRF acquire better performance than both two conventional fuzzy matching methods, they all have more than 90% in F1-measure. Among them, Dice similarity and cosine similarity based on TF score can achieve better performance than the other two ones, that equal to and greater than 93%. Especially, the latter measure can obtain the best performance, which arrives at 93.17%. It improves F1-measure by 8.8% and 9% respectively compared to q-gram and improved edit distance.

We compare the processing speed of different fuzzy matching methods in Table 6. It can be seen that four similarity methods combined CRF model take almost the similar time. Q-gram distance takes much less time of all the methods and improved edit distance method costs almost twice of other four similarity methods based on CRF model. Q-gram method has advantages to deal with deletion mistakes rather than substitution and insertion mistakes of key semantic concept in a sentence. However, there are words' variation induced by ASR and mistakes made by users in the process of SLU, q-gram method will not achieve higher accuracy than similarity measure combined with CRF. In practical application, users can accept such a speed of fuzzy matching method about an average time of 0.4 s for handling a sentence.

6. Conclusions

In this paper, we propose a fuzzy matching method to improve the robustness of SLU based on CRF model and similarity measures. Four kinds of similarity measures are calculated between two strings, which are named entity recognized by CRF model and name in list. Our goal is to obtain the most appropriate one to correct named entities with some errors. Experiments show that Dice similarity and cosine similarity based on TF score, combined with CRF, can achieve better performance.

The contribution of this paper is providing a new and effective method to correct the named entity variations induced by multiple causes. Two conventional methods are compared, which are q-gram distance and improved edit distance. These two methods need each character of a sentence to compute with the string of dictionary. However, our method based on CRF and similarity functions only utilize the named entities recognized by CRF model to compare the similarity. And our method allows more than one target named entities occurred errors. It provides a prerequisite that the system can conduct semantic understanding by only a part of the key information of a user' s query.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 10925419, 90920302, 61072124, 11074275, 11161140319, 91120001) and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant Nos. XDA06030100, XDA06030500).

References

[1] Y.-Y. Wang, D. Yu., Y.-C. Ju, and A. Acero, "An introduction to voice search," IEEE Signal Process. Mag., vol.25, no.3, pp.28–38, May 2008.
[2] N. Gupta, G. Tur, D.H. Tur, and G. Riccardi, "The AT&T spoken language understanding system," IEEE Trans. Audio Speech Language Process., vol.14, no.1, pp.213–222, Jan. 2006.
[3] P.J. Price, "Evaluation of spoken language systems: The ATIS domain," Proc. DARPA Workshop on Speech and Natural Language, pp.91–95, June 1990.
[4] Y.-Y. Wang, "A robust parser for spoken language understanding," Proc. EUROSPEECH, pp.2055–2058, Budapest, Hungary, 1999.
[5] R. Pieraccini, E. Tzoukermann, Z. Gorelov, J.-L. Gauvain, E. Levin, C.-H. Lee, and J.G. Wilpon, "A speech understanding system based on statistical representation of semantics," ICASSP, pp.193–196, San Francisco, USA, 1992.
[6] S. Miller, R. Bobrow, R. Ingria, and R. Schwartz, "Hidden understanding models of natural language," Proc. Annu. Meeting Association for Computational Linguistics, pp.25–32, 1994.
[7] Y. He and S. Young, "Semantic processing using the hidden vector state model," Comput. Speech Language, vol.19, no.1, pp.85–106, 2005.
[8] K. Macherey, F.J. Och, and H. Ney, "Natural language understanding using statistical machine translation," Proc. EUROSPEECH, pp.2205–2208. 2001.
[9] Y.-Y. Wang, A. Acero, C. Chelba, B. Frey, and L. Wong, "Combination of statistical and rule-based approaches for spoken language understanding," Proc. ICSLP, pp.609–612, Denver, Colorado, 2002.
[10] S. Mann, A. Berton, and U. Ehrlich, "How to access audio files of large data bases using in-car speech dialogue systems," INTERSPEECH-2007, pp.138–141, 2007.
[11] G. Zweig, P. Nguyen, Y.-C. Ju, Y.-Y. Wang, D. Yu, and A. Acero, "The voice rate dialog system for consumer ratings," Proc. INTERSPEECH-2007, pp.2713–2716, 2007.
[12] M.L. Seltzer, Y.-C. Ju, I. Tashev, Y.-Y. Wang, and D. Yu, "In-car media search," IEEE Signal Process. Mag., vol.28, no.4, pp.50–60, 2011.
[13] W.-L. Wu, R.-Z. Lu, J.-Y. Duan, H. Liu, F. Gao, and Y.-Q. Chen, "Spoken language understanding using weakly supervised learning," Comput. Speech Language, vol.24, no.2, pp.358–382, 2010.
[14] M. Kiwi, G. Navarro, and C. Telha, "On-line approximate string matching with bounded errors," Theor. Comput. Sci., vol.412, no.45, pp.6359–6370, 2011.
[15] N. Koudas, S. Sarawagi, and D. Srivastava, "Record linkage: similarity measures and algorithms," Proc. 2006 ACM SIGMOD Intl. Conf. on Management of Data, pp.802–803, June 2006.
[16] G. Navarro, R. Baeza-Yates, E. Sutinen, and J. Tarhio, "Indexing methods for approximate string matching," IEEE Data Engineering Bulletin, vol.24, no.4, pp.19–27, 2001.
[17] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, "Robust and efficient fuzzy match for online data cleaning," SIGMOD Conference 2003, pp.313–324, 2003.
[18] W. Cohen and J. Richman, "Learning to match and cluster entity names," Proc. SIGKDD, Edmonton, July 2002.
[19] S. Chaudhuri, V. Ganti, and D. Xin, "Mining document collections to facilitate accurate approximate entity," Proc. VLDB Endowment, vol.2. no.1, pp.395–406, Aug. 2009.
[20] A. Arasu, S. Chaudhuri, and R. Kaushik, "Learning string transformations from examples," PVLDB, vol.2, no.1, pp.514–525, 2009.
[21] A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios, "Duplicate record detection: A survey," IEEE Trans. Knowl. Data Eng., vol.19, no.1, pp.1–16, Jan. 2007.
[22] N. Gonzalo, "A guided tour to approximate string matching," ACM Comput. Surv., vol.33, no.1, pp.31–88, March 2001.
[23] Y.-Y. Wang and A. Acero, "Discriminative models for spoken language understanding," Proc. ICSLP, Pittsburgh, PA, Sept. 2006.
[24] A. McCallum, "Efficiently inducing features of conditional random fields," Proc. 19th Conference in Uncertainty in Artificial Intelligence, pp.403–410, Acapulco, Mexico, 2003.
[25] Y.-L Li and Y.-H. Yan, "New similarity measures for automatic short answer scoring in spontaneous non-native speech," International Conference on Automatic Control and Artificial Intelligence, in Xiamen, China, pp.3677–3681, 2012.
[26] P. Achananuparp, X.-H Hu, and X.-J. Shen, "The evaluation of sentence similarity measures," Proc. 10th International Conference on Data Warehousing and Knowledge Discovery, pp.305–316, Berlin, Germany, 2008.
[27] K. Richard, K. Angelo, and T. Mayya, "Automated assessment of short free-text responses in computer science using latent semantic analysis," Proc. 16th Annual Joint Conference on Innovation and Technology in Computer Science Education (ITiCSE), pp.158–162, Darmstadt, Germany, 2011.
[28] P. Malakasiotis and I. Androutsopoulos, "Learning textual entailment using SVMs and string similarity measures, Proc. Annual Meeting of the Association for Computational Linguistics (ACL), pp.42–47, 2007.
[29] X.-D. Zhu, M. Li, J.-F. Gao, and C.-N. Huang, "Single character Chinese named entity recognition," Proc. 2nd SIGHAN Workshop on Chinese Language, pp.125–132, Sapporo, Japan, 2003.
[30] C.-C. Bao, W.-Q. Xu, and Y.-H. Yan, "Recognizing named entities in spoken Chinese dialogues with a character-level maximum entropy tagger," Proc. Interspeech-2008, pp.1145–1148, 2008.

Yanling Li received the M.S. degrees in Communication and Information System from Soochow University. Now she is a Ph.D. candidate in Key Laboratory of Speech Acoustics and Content Understanding at Institute of Acoustics, Chinese Academy of Sciences. His research interests include spoken language understanding and natural language processing.

Qingwei Zhao received PhD in Signal & Information Processing from Tsinghua University, 1999. Before joining Key Laboratory of Speech Acoustics and Content Understanding at Institute of Acoustics, he was the senior researcher and project manager in Intel China Research Center, manager of core technology department of Tech-Lan Voice Technology Co. Ltd.

Yonghong Yan received the B.E. degree at the Electronic Engineering Department of Tsinghua University in 1990, and Ph.D. in Computer Science and Engineering from Oregon Graduate Institute of Science & Engineering in 1995. From 1995 to 1998, he worked in OGI as Assistant Professor, Associate Director and Associate Professor of the Center for Spoken Language Understanding, and from 1998 to 2001 he worked as the Principal Engineer of Intel Microprocessors Research lab, Director and Chief Scientist of Intel China Research Center. In 2002 he returned to China to work for Chinese Academy of Sciences. He is a professor and director of Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences. His research interests include large vocabulary speech recognition, speaker/language recognition and audio signal processing. He has published more than 100 papers and holds 40 patents.