Extracting Relationship of Meeting Minutes Generated by Speech Recognition System

Motoki Ito, Seikoh Nishita

Abstract—A minutes generation system by speech recognition automatically records minutes generated from voices in the meeting. In case where generated minutes are not strictly managed, the minutes possibly include error words caused by the speech recognizer. Such error words makes information retrieval on minutes difficult. To address the problem, this paper proposes a technique to extract relationship of minutes generated by speech recognition systems. Our technique is based on “collective entity resolution in relational data”. This paper also reports an experimental evaluation of our technique. The experimental result suggests effectiveness of the technique for minutes texts including error words.

Index Terms—Speech recognition, meeting minutes, text mining, entity resolution

I. INTRODUCTION

As a recent progress of speech recognition technology, minutes generation systems by speech recognition are increasingly introduced into formal/informal assemblies, meetings and seminars. The minutes generation system by speech recognition automatically records minutes generated from voices in the meeting. As a principled basis, the system involves mis-recognition: there are possibly incorrectly recognized words (error words) in the minutes, because speech recognition may fail to identify voices. Therefore, the system requires scribes who manipulate the system console to correct the error words in formal meetings. On the other hand, if there is no correction (this may happen in the use of informal meetings), the error words are left in the minutes, leading to worse performance of information retrieval on the minutes.

To address the problem, this paper proposes a technique based on collective entity resolution (CER)[1] to extract relationship of minutes possibly including error words caused by mis-recognition of speech recognition systems. Given a set of the minutes texts and a pair of texts in the set, our technique extracts a relationship of the pair of texts by mis-recognition. Figure 1 shows our motivative example of a set of texts generated by the speech recognition.

Fig. 1. Text examples generated by speech recognition, underlined words are mis-recognition.

In this week, I would like to hold a party to sing a fun song.

In this week, I investigated about the proximity sensor and “smart at phone” refers to the same word. In order to link an error word to its spoken word, we leverage CER, which links words by combination of attribute similarity of words (phonemes in this paper), and co-occurrence information.

II. KEY OBSERVATION

Keyword extraction is a popular technique at a stage prior to similarity calculation of a pair of texts in general. However, the keyword extraction does not work fine for texts generated by the speech recognition, since a keyword extracted may be an error word. Figure 1 shows our motivative example of a set of texts generated by the speech recognition.

The underline in the text denotes an error word with a parenthesized correct spoken word. The text 1 and 2 describe same topic on smart phones. Therefore, we expect that the similarity of text 1 and 2 is relatively higher. However, there is a common word, “fun song”, in both text 1 and 3. If they are found as keywords, the similarity of the text 1 and 3 would be falsly higher. In order to address the problem, we focus the characteristics of error words and text generated by the speech recognition:

- An error word and its spoken word are similar with each other in respect of phonemes, since the system recognizes a word by given voice and phonemes.
- An error word and its spoken word are similar with each other, when co-occurrence words of them are also similar with each other.

In figure 3, the word “smart phone” in text 1 and “smart at phone” in text 2 have similar phonemes. Moreover, they have both keyword “proximity sensor” as co-occurrence. These information may give a reason that the word “smart phone” and “smart at phone” refers to the same word. In order to link an error word to its spoken word, we leverage CER, which links words by combination of attribute similarity of words (phonemes in this paper), and co-occurrence information.

In this week, I would like to hold a party to sing a fun song.

Text1

Text2

Text3

Next week, I would like to hold a party to sing a fun song.
III. COLLECTIVE ENTITY RESOLUTION

For brief explanation of CER, we give an example that is illustrated in paper[1]. The following is three descriptions in a census record:

1) Jonathan Doe is married to Jeanette Doe, and he has dependents, Jim and Jason Doe,
2) Jon Doe is married to Jean Doe,
3) and J.Doe has dependents, Jim, Jason and Jackie Doe.

Entity resolution in this example is a task to assign a real world entity (a person described in the record) to each reference (a name appearing in the description). Since the census record possibly includes duplicated descriptions, any pair of names like ‘J.Doe’ and ‘Jon Doe’ may refer to the same person. To solve the entity resolution, CER constructs a reference graph (Fig.2) from the descriptions as its first step. The reference graph is composed of names appearing in the description as nodes, and co-occurrence information as hyper-edges. Second, CER forms an entity graph(Fig.3), whose nodes are clusters representing real world entities (people, subject of census). Each cluster is a collection of names that all refer to the same person.

The entity resolution algorithm of CER is a greedy agglomerative clustering algorithm, which consists of three steps, blocking, bootstrapping and merging clusters; the blocking step finds potential resolution candidates for each reference, the bootstrapping step makes initial small clusters and ‘fun song’ in text2 are separated into two clusters. In order to apply CER to solve the problem in this paper, we need following consideration:

- reference, entities and hyper-edges for the problem of automatically generated minutes,
- method of eliminate common references as stopwords.
- similarity of references \( r_1 \) and \( r_2 \) used in the blocking step, \( \text{sim}_L(r_1, r_2) \),
- similarity of \( r_1 \) and \( r_2 \) used in the bootstrapping step, \( \text{sim}_S(r_1, r_2) \),
- and attribute based similarity of \( r_1 \) and \( r_2 \) used in the merging clusters step, \( \text{sim!}(r_1, r_2) \).

We note that CER uses both co-occurrence-based and attribute-based similarity, and the latter only requires the concrete definition for each application of CER.

Fig. 2. A reference graph for the census record

IV. THE RELATIONSHIP EXTRACTION OF AUTOMATICALLY GENERATED MINUTES

This section proposes a technique to extract relationship of minutes automatically generated by the speech recognition. As an assumption of the problem, we suppose that input data is a collection of texts, each of which is a minute of one theme in a meeting. Figure 1 shows an example collection of three texts. The underline in the text denotes an error word with a parenthesized correct spoken word.

The goal of the problem is to obtain similarities for all pairs of texts in the given collection. The following is stages of the technique.

1) Stopword elimination for all texts in the collection
2) Entity Resolution by CER
3) Similarity calculation for a given pair of texts.

A. Stopword elimination

Stopword elimination is a stage prior to apply CER. For the stopword elimination, we employ one of two simple algorithms with a morphological analyzer. Each of them, first, obtains a set of all noun words from the given texts by the morphological analysis, and second, it eliminates stopwords from the set. The two algorithms are distinguished with each other by a condition deciding stopwords

- \( \text{Freq} \) regards commonly appearing words as the stopwords. For each noun word, it counts the number of texts where the noun appears, and if the number is more than a threshold, it eliminates the noun as a stopword.
- \( \text{Ti} \) regards unimportant noun words with respect to TF-IDF as stopwords.

B. Entity resolution by CER

As an application of CER, our technique regards a keyword appearing in texts of the collection as a reference, a spoken-word for the keyword as an entity, and co-occurrence in each text as a hyper-edge. Figure 4 and 5 illustrates a reference graph and its entity graph for the texts of Fig.1 respectively. We note that the error word, ‘smut at phone’ and the correctly recognized word, ‘smart phone’ are included in the same cluster in Fig.5, although ‘fun song’ in text1 and ‘fun song’ in text2 are separated into two clusters. In order to implement clustering like Fig.5, we need to define similarities of references in CER.

Preliminary to the definition of the similarities, we introduce symbols and denotations for data structure in CER: The
symbol \( r \) denotes a reference for a keyword appearing in the given texts. The reference \( r \) has three attributes: \( r.k \) is the
keyword itself, \( r.p \) is the phoneme of the keyword, and \( r.t \) is the text where the keyword \( r.k \) appears. We note that any
reference \( r \) is distinguished with another reference \( r' \) by its
texts \( r.t \neq r'.t \), even if the same keyword \( r.k = r'.k \). The
symbol \( c \) denotes a cluster in CER. The symbol \( t \) denotes
one of the given text. The term \( t.R \) denotes the set of the
references whose text attributes is \( t \). The term \( t.C \) denotes
the set of the clusters, each of which has a reference whose text
is \( t \).

\[
\text{t.C} = \{ c \mid r \in c, r \in t.R \}
\]

We may uses subscripts \( i \) and \( j \) for all symbols to denote
two independent data.

As described in Section III, the definition of the similarities
for the blocking, bootstrapping and merging clusters steps
are required for application of CER. First, We focus on the
similarity \( \text{sim}_L \) in the blocking step and the attribute based
similarity \( \text{sim}_A \) in the iterative merging cluster step. Every
keyword in the given texts is possibly an error word. Since
voices are the source of both an error word and a correctly
recognized word in the speech recognition, the source voices
are similar with each other. Our technique uses the phonemes
of the keywords for the attribute based similarity. We define
the similarity \( \text{sim}_A(r_i,r_j) \) and \( \text{sim}_L(r_i,r_j) \) using the edit
distance of the keyword and the phoneme of the references
\( r_i \) and \( r_j \):

\[
\text{sim}_L(r_i,r_j) = 1 - \frac{\text{edist}(a,b)}{\max(|a|,|b|)}
\]

The term \( \max(x,y) \) denotes the greater value of the values
\( x \) and \( y \). The term \( \text{cost}(a,b) \) denotes the edit distance of
character sequences \( a \) and \( b \). The expression 2 normalizes
the edit distance of sequence to the range of 0.0 to 1.0; The
value 1.0 means that \( a \) and \( b \) are exactly the same character
sequences. The expression 1 obtains the edit distances with
respect to the keyword and its phoneme, then combines the
distances with the factor \( \beta \).

Since CER is an agglomerative clustering algorithm, once
two clusters are merged, there is no way to divide them again.

\[
\text{sim}_A(r_i,r_j) = (1-\beta) \times \text{edist}(r_i,r_j) + \beta \times \text{edist}(r_i,p,r_j,p)
\]

The following is the definition of \( \text{sim}_A \) which is defined with \( \text{sim}_A \) and combination factor the \( \alpha \).

\[
\text{sim}(c_i,c_j) = (1-\alpha) \times \text{sim}_A(c_i,c_j) + \alpha \times \text{sim}_R(c_i,c_j)
\]

\[
\text{sim}_R(c_i,c_j) = \max\{\text{sim}_R(r_i,r_j)|r_i \in c_i, r_j \in c_j\}
\]

\[
\text{sim}_R(c_i,c_j) = \frac{|\text{Nbr}(c_i) \cap \text{Nbr}(c_j)|}{|\text{Nbr}(c_i) \cup \text{Nbr}(c_j)|}
\]

\[
\text{Nbr}(c) = \{ r' . c | r \in c, r' \in r.t.R, r'.e \neq c \}
\]

C. Similarity calculation

Given pair of texts \( (t_i,t_j) \) in the collection, similarity
calculation obtains a degree of similarity of the texts using the
result of CER. We employ one of following two algorithms
for the calculation.
**Jac** is a Jaccard coefficient of two cluster sets for the text $t_i$ and $t_j$:

$$JacSim(t_i, t_j) \equiv \frac{|t_i \cap t_j|}{|t_i \cup t_j|}$$

The denominator of the right-hand side of the equation is the number of clusters, each of which has references appearing in $t_i$ or $t_j$. On the other hand, the numerator is the number of clusters, each of which has references appearing in both $t_i$ and $t_j$. Therefore, the right-hand side of the equation means the ratio of the common clusters of the two texts. This algorithm is based on observation that any pair of texts is likely similar with each other when the clusters connected with a pair of hyper-edges for the texts are highly overlapped in the entity graph.

**Cos** is based on an improvement of a well-known relationship-extraction technique, i.e. the combination of the keyword extraction[2] and the cosine similarity. As mentioned in Section II, error words arisen by the speech recognition possibly disserve the keyword extraction. In order to reduce the influence of the error words, this algorithm rewrites input texts $t_i$ and $t_j$, according to the result of CER; first, it selects a representative reference $r$ from each cluster $C$ in a random manner, then second, it rewrites every occurrence $r_i.k$ for a reference $r_i \in C$ in text $t_i$ and $t_j$ into $r.k$. Figure 6 shows the result of the algorithm **Cos** for the texts in Figure 5. The underline indicates that the word is re-written. Since the word “smart phone” and “smut at phone” are in single cluster C6 in Figure 5, the algorithm rewrites these words to randomly selected representative (“smart phone” in this case). As a final step of **Cos**, the standard cosine similarity is calculated with rewritten texts. A feature vector of text $t_i$ is defined as follows:

$$\vec{t}_i = (w_{i1}, w_{i2}, \ldots, w_{in})$$

where $w_{ij}$ is the importance of keyword $k_j$ in text $t_i$, which is obtained by the keyword extraction. Then, the similarity of text $t_i$ and $t_j$ is defined as follows:

$$CosSim(t_i, t_j) = \frac{\vec{t}_i \cdot \vec{t}_j}{|\vec{t}_i| \cdot |\vec{t}_j|}$$

V. IMPLEMENTATION

We developed a prototype of the proposal technique targeted on meeting minutes written in Japanese. The prototype is composed of Java and Perl programs. The keyword extraction and the cosine similarity calculation in **Cos** are implemented as a Perl program with Term Extractor[2]. The other part of the prototype is implemented as a Java program that leverages Mecab[4] as a Japanese morphological analyzer, Apache Lucene[5] for edit-distance calculation, and ICU4J[6] for translation of phonemes from noun words. Since Japanese words consist of mixture of phonograms (named Hiragana and Katakana) and ideograms (named Kanji character), translation of phonemes from Japanese word is not trivial task. Given Japanese noun word, transliterator class in ICU4J obtains the Roman alphabet (named romaji), which we regard as a phoneme of the word.

VI. EXPERIMENTAL EVALUATION

This section describes an experiment for the evaluation of our technique.

We prepared three set of 20 texts as experimental data. In order to generate the texts, we, first, recorded voice data by reading aloud parts of following books in Japanese:

- **Book1**: technical reports of software engineering
- **Book2**: a book on relationship between engineering science and math
- **Book3**: a instruction book on data structure and algorithms in Java programs
- **Book4**: a textbook on teaching about engineering

We recorded five voice data per a book, totally 20 voice data. The average of the number of noun phrases appearing in a voice data is 199. The number of noun phrases appearing in voice data of two or more books is 9.93 % of the all noun phrases. Second, we generated three sets of 20 texts by applying the speech recognition software “AmiVoice@SP2[7]” three times while changing a dictionary as follows:

- **Dict 1**: a multipurpose dictionary with large vocabulary words
- **Dict 2**: a multipurpose dictionary with small vocabulary words
- **Dict 3**: a dedicated dictionary on politics and economics

Since the four books are all concerned with engineering and science, the speech recognition with dict 1 is expected to be high accuracy. On the other hand, the other two dictionaries are expected to degrade the speech recognition. Therefore, a set of texts is generated under highly accurate speech recognition with dict 1, and the other two sets are under relatively inaccurate speech-recognition.

We leveraged four proposal techniques by switching two stopword eliminations (**Freq** and **TI**) and two similarity calculations (**Jac** and **Cos**). In addition, for comparison to existing technique, we used standard technique (named Cos) of relation extraction, that is, the combination of the keyword extraction and the cosine-similarity calculation applied to the three sets of text including error words. We assume that every technique judges any pair of texts are similar with each other, if and only if the similarity obtained is equals to or more than a threshold, which we define as the average of similarities of all pairs in the given set of texts.
In contrast to judgement, we regard that any pair of texts should be similar with each other, if and only if both of them are generated from same text. We used F-measure as evaluation index of each similarities.

A. Evaluation of the techniques

Table I illustrates the experimental result. We note that we obtained “idealized score of F-measure” 0.54, by applying Cos to the correct set of texts that are correctly written all parts in books corresponding to the sets of the experimental texts.

The accuracy of input data affects the standard technique Cos, which is relatively low score with inaccurate data sets. In contrast to Cos, four proposed techniques keeps scores, even if the dataset is inaccurate.

With respect to the stopword elimination, scores of F-measure of Ti- are greater than Freq- in all cases. On the aspect of the similarity calculation, scores of -Cos are greater than one of -Jac in all cases. Totally, the technique of TiCos obtains scores around 0.45 that is close to the idealized score 0.54. This result shows that our technique efficiently extracts relationship of texts including error words caused by speech recognition, especially with the stopword elimination by TF-IDF, and with similarity calculation by the standard keyword extraction and the cosine similarity calculation to rewritten text according to the collective entity resolution.

VII. Conclusion

This paper proposed a technique to extract relationship of minutes generated by speech recognition system. Our technique is based on the collective entity resolution. Our proposal technique is combined a technique of stopword elimination and a technique of similarity calculation using the cluster generated by CER. We evaluated our technique by using the experimental data generated by a speech recognizer in order to find the best combination of each steps in our technique. According to the experimental result, the best combination is composed of the stopword elimination using TF-IDF and the cosine similarity calculation with rewritten texts using the cluster generated by CER. Moreover, the experimental result suggests that our technique extracts relationship of minutes texts including error words arisen by the speech recognition more effectively than the standard technique of the keyword extract and the cosine similarity calculation.

REFERENCES


<table>
<thead>
<tr>
<th>methods of calculating similarities</th>
<th>Cos</th>
<th>FreqJac</th>
<th>FreqCos</th>
<th>TiJac</th>
<th>TiCos</th>
</tr>
</thead>
<tbody>
<tr>
<td>accurate dataset using Dict1</td>
<td>0.38</td>
<td>0.36</td>
<td>0.40</td>
<td>0.43</td>
<td>0.44</td>
</tr>
<tr>
<td>inaccurate dataset using Dict2</td>
<td>0.31</td>
<td>0.34</td>
<td>0.41</td>
<td>0.44</td>
<td>0.46</td>
</tr>
<tr>
<td>inaccurate dataset using Dict3</td>
<td>0.32</td>
<td>0.39</td>
<td>0.41</td>
<td>0.43</td>
<td>0.45</td>
</tr>
</tbody>
</table>