Extracting Relationship of Meeting Minutes Generated by Speech Recognition System using Entity Resolution

Seikoh Nishita, Motoki Itoh

***Abstract*—A minutes generation system by speech recognition automatically records minutes generated from voice data in the meeting. If generated minutes were not strictly managed, the minutes would possibly include error words caused by the speech recognition. 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 the speech recognition. 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 the speech recognition automatically records minutes generated from voice data in the meeting. As a principled basis, the system involves *mis-recognition*: there are possibly incorrectly recognized words (error words) in the minutes, because the 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 be 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 the mis-recognition of the speech recognition. 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 similarity calculation. We note that our technique uses only texts generated by the system; intermediate data structures like phonemes and conversion candidates during the speech recognition is not leveraged.

Section II illustrates key observation of our study. Section III briefly describes CER. Section IV proposes our technique. Section V and VI report implementation of the technique and an experimental result respectively. Section VII concludes the paper.

Manuscript received July 15, 2016.

Seikoh Nishita is with Department of Computer Science in Faculty of Engineering of Takushoku University (e-mail:snishita@cs.takushoku-u.ac.jp).

Motoki Itoh was with Information and Design Science Engineering Graduate School of Takushoku University. He is now with Excite Japan Co.,LTD.

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 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 falsely higher.

In order to address the problem, we focus on 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 Fig.1, the word "smart phone" in text 1 and "smut at phone" in text 2 have similar phonemes. Moreover, they have both keyword "proximity sensor." These information may give a reason that the word "smart phone" and "smut at

Text1

In this week, I tried to implement a fun song(function) of application that shows alert using proximity sensor of smart phone.

Text2

In this week, I investigated about the proximity sensor of smut at phone(smart phone) .

Text3

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

Fig. 1. Texts including error words generated by speech recognition

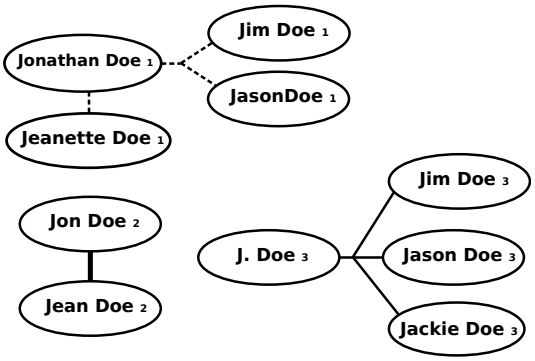

Fig. 2. A reference graph for the census record

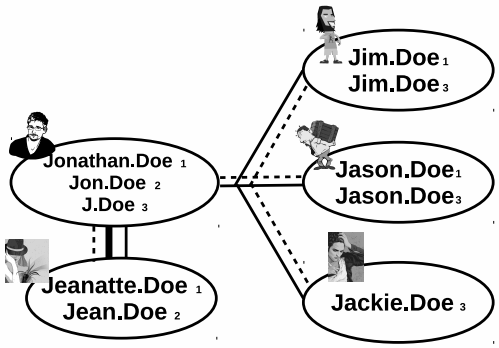

Fig. 3. An entity graph for the census record

phone" refers to the same word. In order to link an error word to its spoken word, we leverage CER, which links words according to combination of attribute similarity of words (phonemes in this paper), and co-occurrence information.

III. Collective Entity Resolution

For brief explanation of CER, we give an example that is illustrated in the original paper[1]. The following are 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). 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, and the bootstrapping step makes initial clusters, each of which has the small number of references. As the final step, the merging-clusters step iteratively merges similar clusters. In order to apply CER to solve the problem in this paper, we need following definitions:

- references, entities and hyper-edges for the problem of automatically generated minutes,
- a method to filter out common words as stop words,
- the similarities $\mathsf{sim}_{\mathsf{L}}(r_1, r_2)$ and $\mathsf{sim}_{\mathsf{S}}(r_1, r_2)$ used in the blocking step and the bootstrapping step respectively, for given references r_1 and r_2, and
- the attribute-based similarity $\mathsf{sim}_{\mathsf{A}}(r_1, r_2)$ used in the merging-clusters step.

Note that the merging-clusters step uses both the co-occurrence-based and the attribute-based similarities, and the latter only requires the concrete definition for each application of CER.

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. In order to illustrate our technique, we use texts in Fig. 1 as an example collection.

The goal of the problem is to obtain similarities for a pairs of texts in the given collection. Our technique consists of following three stages.

1) Stop-word elimination for all texts in the collection,
2) entity resolution by CER, and
3) similarity calculation for a given pair of texts.

A. Stop-word elimination

Stop-word elimination is a stage prior to apply CER. Since clustering of CER uses co-occurrence information, commonly appearing words in the minutes texts affect the clustering results. This is the reason why our technique filters out the commonly appearing words i.e. *stop words* before CER.

For the stop-word elimination, we employ one of two simple algorithms with a morphological analysis. Each of them, first, obtains a set of all noun words from the given texts by the morphological analysis, and second, it filters out stop words from the set. The two algorithms are distinguished with each other by the condition of filtering.

- For each noun word, algorithm Freq counts the number of texts where the noun appears. If the number is more than a threshold, it filters out the noun as a stop word.
- Algorithm Ti filters out a noun word, if tf-idf value of the noun is less than a threshold.

B. Entity resolution by CER

The stop-word elimination leaves keywords that represent characteristics of texts in the given collection. As an application of CER, our technique regards each of the keywords as a reference, a spoken-word for the keyword as an entity, and co-occurrence in each text as a hyper-edge. Figure 4

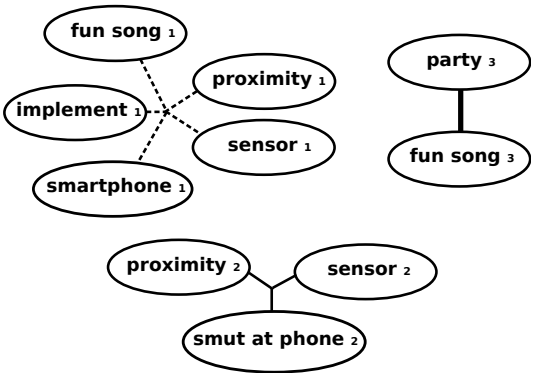

Fig. 4. A reference graph for the meeting minutes

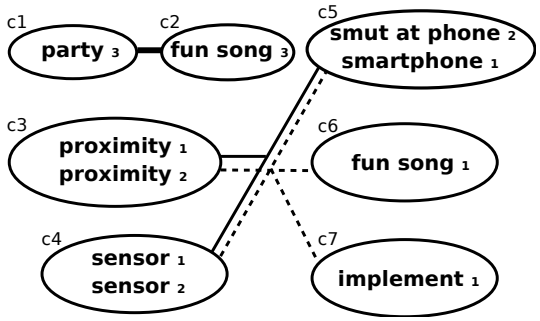

Fig. 5. An entity graph for the meeting minutes

and 5 respectively illustrates a reference graph and its entity graph for the texts of Fig.1. 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 text 1 and 'fun song' in text 2 are separated into two clusters. In order to implement clustering like Fig.5, definition of similarities is required.

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 appears. It is notable that any reference r is distinguished with another reference r' by its texts $r.t \neq r'.t$, even if $r.k = r'.k$. The symbol c denotes a cluster in CER. The symbol t denotes one of the given texts. The term $t.R$ denotes the set of the references whose text attributes is t. The term $t.C$ denotes the set of clusters, each of which has a reference whose text is t.

$$t.C = \{c \mid r \in t.R,\ r \in c\} \tag{1}$$

All symbols may have subscripts of i and j in order to denote independent data.

It is noteworthy that a cluster in CER is a set of references, and any pair of clusters in the entity graph are mutually exclusive. Therefore, any reference r has only one cluster c such that $r \in c$ at any instance of the merging-clusters step.

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 in the blocking step and the attribute based similarity 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 similarities as a combination of edit distance of the keyword and the phoneme of the references r_i and r_j:

$$\begin{aligned}&\mathsf{sim_L}\,(r_i, r_j)\\ &\equiv \mathsf{sim_A}\,(r_i, r_j)\\ &\equiv (1-\beta) \times \mathsf{edist}\,(r_i, r_j) + \beta \times \mathsf{edist}\,(r_i.p, r_j.p)\\ &\qquad\qquad (0 \leq \beta \leq 1)\end{aligned} \tag{2}$$

$$\mathsf{edist}\,(a, b) \equiv 1.0 - \frac{\mathsf{cost}\,(a, b)}{\mathsf{max}\,(|a|, |b|)} \tag{3}$$

$\mathsf{sim_L}(r_i, r_j)$ denotes the similarity of the reference r_i and r_j used in the blocking step, and $\mathsf{sim_A}(r_i, r_j)$ denotes the attribute-based similarity in the merging clusters step. The term $\mathsf{max}\,(x, y)$ denotes larger value in the two arguments x and y. The term $\mathsf{cost}\,(a, b)$ denotes the edit distance of the 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 is exactly same as b. The expression (2) obtains the edit distances with respect to the keyword and its phoneme, and combines them with the factor β.

Since CER is an agglomerative clustering algorithm, once two clusters are merged, there is no way to divide them again. Therefore, the bootstrapping step should make small clusters with scrupulous care by a precise calculation of the similarity $\mathsf{sim_S}$. We define $\mathsf{sim_S}$ as the combination of the attribute-based similarity $\mathsf{sim_A}$ and co-occurrence information as follows:

$$\begin{aligned}&\mathsf{sim_S}\,(r_i, r_j)\\ &\equiv (1-\alpha) \times \mathsf{sim_A}\,(r_i, r_j) + \alpha \times \mathsf{coo}\,(r_i, r_j)\\ &\qquad\qquad (0 \leq \alpha \leq 1)\end{aligned} \tag{4}$$

$$\mathsf{coo}\,(r_i, r_j) \equiv \frac{|r_i.t.R \cap r_j.t.R|}{|r_i.t.R \cup r_j.t.R|} \tag{5}$$

The term $\mathsf{sim_A}\,(r_i, r_j)$ means the similarity with respect to the edit distance defined at the expression (2). The term $\mathsf{coo}\,(r_i, r_j)$ means co-occurrence information: the overlap ratio of co-occurred keywords in $r_i.t.R$ and $r_j.t.R$. The expression (4) combines these similarities with the factor α, which represents the ratio of importance of the two types of the similarities, the attribute-based similarity and the co-occurrence based similarity.

The iterative merging clusters step decides if two cluster should be merged. These decision utilize the similarity $\mathsf{sim}\,(c_i, c_j)$ of clusters c_i and c_j. The following is the

definition of sim:

$$\mathsf{sim}(c_i, c_j) \equiv (1-\alpha)\times\mathsf{sim_A}(c_i, c_j) + \alpha\times\mathsf{sim_R}(c_i, c_j)$$
$$\mathsf{sim_A}(c_i, c_j) \equiv \mathsf{max}\{\mathsf{sim_A}(r_i, r_j) \mid r_i \in c_i, r_j \in c_j\}$$
$$\mathsf{sim_R}(c_i, c_j) \equiv \frac{|\mathsf{Nbr}(c_i) \cap \mathsf{Nbr}(c_j)|}{|\mathsf{Nbr}(c_i) \cup \mathsf{Nbr}(c_j)|}$$
$$\mathsf{Nbr}(c_i) = \{c_j \mid r_i \in c_i,\ r_j \in r_i.t.R,\ r_j \in c_j,\ c_j \neq c_i\}$$

The term $\mathsf{Nbr}(c_i)$ is the set of "neighbor" clusters of the given cluster c_i; a cluster c_j is a neighbor of c_i, if and only if there is a text where some reference $r_i \in c_i$ and $r_j \in c_j$ both appear. The term $\mathsf{sim_R}(c_i, c_j)$ means the co-occurrence based similarity of the cluster c_i and c_j. The similarity $\mathsf{sim}\,(c_i, c_j)$ is defined with $\mathsf{sim_A}$, $\mathsf{sim_R}$ and the combination factor α that is also used in definition of $\mathsf{sim_S}$. We adjust the threshold of sim with the factors α and β in $\mathsf{sim_A}$ during the experiment described in Section VI.

The three steps in CER uses the similarity functions $\mathsf{sim_L}$, $\mathsf{sim_S}$, and sim with some thresholds for the decisions. The similarity $\mathsf{sim_L}$ in the blocking step is utilized to decide potential resolution candidates for each keyword. Since the blocking step should detects all candidates for every keyword thoroughly, we set relatively low threshold for $\mathsf{sim_L}$. On the other hand, the similarity $\mathsf{sim_S}$ in the bootstrapping step is used to decide if any two keywords should be put into the same initial cluster. As mentioned above, once two clusters are merged in the iterative step, there is no way to divide them again. Therefore, the decision in the bootstrapping step should be performed carefully. As a result, we set relatively high threshold for $\mathsf{sim_S}$.

C. Similarity calculation

Given pair of texts (t_i, t_j) in the given collection, similarity calculation analyzes an entity graph in CER, and obtains a degree of the similarity for the pair. 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 :

$$\mathsf{Jac}(t_i, t_j) \equiv \frac{|t_i.C \cap t_j.C|}{|t_i.C \cup t_j.C|}$$

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 for the two texts. This algorithm is based on the 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, for every reference $r' \in c$, if $r'.t$ is the text t_i or t_j, the algorithm rewrites in the keyword $r.k$ from all occurrence $r'.k$ in the text $r'.t$. Figure 6 shows a result of rewrite by the algorithm Cos from the texts in Fig.1 with the entity graph of Fig. 5. The underline indicates that the word is rewritten. Since the word "smart phone" and "smut at phone" are in single cluster C6 in the entity graph, the algorithm rewrites these words in randomly selected representative ("smart phone" in this case).

Text1 (rewritten)

In this week, I tried to implement a function of application that shows alert using proximity sensor of smart phone.

Text2 (rewritten)

In this week, I investigated about the proximity sensor of smart phone .

Text3 (rewritten)

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

Fig. 6. Texts rewritten by the algorithm Cos

As a final step of Cos, it calculates the standard cosine similarity of the rewritten texts. A feature vector of text t_i is defined as follows:

$$\vec{t_i} \equiv (w_{i1}, w_{i2}, \ldots, w_{in})$$

where w_{il} is the importance of keyword k_l 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:

$$\mathsf{Cos}(t_i, t_j) = \frac{\vec{t_i} \cdot \vec{t_j}}{|\vec{t_i}| \cdot |\vec{t_j}|}$$

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 Extract[2]. The other part of the prototype is implemented as Java programs that leverage Mecab[5] as a Japanese morphological analyzer, Apache Lucene[6] for edit-distance calculation, and ICU4J[7] 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.

TABLE I
F-MEASURE OF THE SIMILARITY CALCULATIONS

dataset	dictionary	vocabulary	FreqJac	TiJac	FreqCos	TiCos	**Cos**
Da	multipurpose	large	0.36	0.40	0.43	0.44	0.38
Db	multipurpose	small	0.34	0.41	0.44	0.46	0.31
Dc	dedicated	politics and economics	0.39	0.41	0.43	0.45	0.32
Di	-	-	-	-	-	-	0.54

A. Experiment to evaluate relation-extraction techniqes

In order to evaluate the proposal technique, we perform an experiment with relation-extraction techniques including the proposal ones in this paper. This section describes the experiment itself, the techniques, and the datasets used in the experiment.

As a first step of the experiment, we categorize all texts in a given dataset, where the categories provide the definition of the similarity of texts, i.e. we regard any pair of texts in a category as similar with each other. Hence, a similarity value obtained by a relation-extraction technique is reasonable with respect to the categorization, if the technique obtains relatively high similarity value for every pair in a category, and low similarity value for every pair straddling on two categories.

In order to simplify the evaluation, given a relation-extraction technique and a dataset, we introduce a threshold defined as the average of similarities obtained by the technique for all pairs in the given dataset. Using the threshold, we assume that the technique judges any pair of texts are similar with each other, if and only if the similarity value obtained is equals to or more than the threshold. As evaluation index of the techniques, we use F-measure.

We leveraged five relation extraction techniques in the experiment; four of them are all proposal ones in this paper by switching two stop-word eliminations (Freq- and Ti-) and two similarity calculations (-Jac and -Cos). In contrast, the other is a standard technique (named **Cos**) of relation extraction, that is, the combination of the keyword extraction and the cosine-similarity calculation. It is notable that **Cos** would be easily influenced by error words caused by the speech recognition as mentioned in Section II.

As a dataset, we, first, recorded voice data of meetings in our laboratory, and generated meeting minutes from the voice data by a speech recognizer. However, it is difficult to categorize meeting minutes, because there are a lot of aspects in discussion in a meeting. Moreover, a pair of texts in the minutes are similar with each other in some aspect, but not similar in other aspect. Therefore, we stopped using the meeting minutes as the dataset in the experiment.

Instead of the meeting minutes, we recorded voice data by reading aloud parts of some books, and prepared texts generated from the voice data. In addition, we regard the books as categories; we define a pair of texts are similar with each other, if both texts are generated from same book. We selected four Japanese books of following topics:

- technical reports on software engineering,
- relationship between engineering and mathematics,
- data structure and algorithms in Java programs, and
- engineering education.

We read aloud five parts per a book and recorded totally 20 voice data. There were about 4,000 noun phrases appearing among all of the parts, and 9.93% of them appeared in two or more books. Second, we made following four dataset Da, Db, Dc and Di of texts: the first three of them are generated from the voice data by applying the speech recognition software "AmiVoice®SP2[8]", and the other is entered manually from the parts of the books skipping the speech recognition.

Da: generated with a multipurpose dictionary with large vocabulary
Db: generated with a multipurpose dictionary with small vocabulary
Dc: generated with a dedicated dictionary on politics and economics
Di: input manually from the parts of the books.

All of the datasets have 20 texts for the parts we recoded. The dataset Di is the ideal dataset, because it has exactly same texts as ones in the books. The other three datasets possibly have error words caused by the speech recognition. Especially, the dataset Db and Dc are expected to have more error words than Da, since the four books are all concerned with engineering and science, and the speech recognition with a dictionary with large vocabulary is expected to have high accuracy.

B. Experimental results and evaluation

Table I illustrates the experimental result. Note that we obtained "the ideal score of F-measure", 0.54, as the result of **Cos** with the ideal dataset Di.

The scores of the standard technique **Cos** with the inaccurate datasets are relatively low. This results suggests that the accuracy of input data affected **Cos**. In contrast to **Cos**, four proposed techniques keeps scores, even if the dataset is inaccurate.

With respect to the stop-word elimination, F scores 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 ideal score 0.54. This result shows that our technique efficiently extracts relationship of texts including error words caused by the speech recognition.

VII. CONCLUSION

This paper proposed a technique to extract relationship of minutes generated by a speech recognition system. Our technique is based on the collective entity resolution. Our proposal technique is combined a technique of stop-word elimination and a technique of similarity calculation using clusters generated by CER. We evaluated our technique by an experimental dataset 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 stop-word 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 texts including error words more effectively than the standard technique of the keyword extract and the cosine similarity calculation.

REFERENCES

[1] I.Bhattacharya and L.Getoor, "Collective Entity Resolution in Relational Data," *ACM Trans. Knowledge Discovery from Data*, vol.1, no.1, 2007.

[2] H.Nakagawa, H.Yumoto, T.Mori, "Term Extraction Based on Occurrence and Concatenation Frequency, " *Journal of Natural Language Processing*, Vol.10, no.1, pp.27–45, January, 2003.

[3] M.Itoh, S.Nishita, "Extracting Relationship of Meeting Minutes Generated By Speech Recognition System using Collective Entity Resolution, " *4th ICT International Student Project Conference*, 1A1, May, 2015, (CD-ROM).

[4] M.Itoh, S.Nishita, "Extracting Relationship of Meeting Minutes Generated by Speech Recognition System," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2016, IMECS 2016, 16-18 March, 2016, Hong Kong, pp.269–273.

[5] "MeCab Yet Another Part-of-Speech and Morphological Analyzer," Accessed July 1, 2016, http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html.

[6] "Apache Lucene Core: A high-performance, full-featured text search engine library," Apache Software Foundation, Accessd July 1, 2016, http://lucene.apache.org/core/.

[7] "ICU International Components for Unicode," Accessed July 1, 2016, http://site.icu-project.org/.

[8] "AmiVoice SP2: A speech recognition software," Advanced Media, Inc., Accessed July 1, 2016, http://sp.advanced-media.co.jp/.