Spoken Term Detection and Spoken Content Retrieval: Evaluations on NTCIR-11 SpokenQuery&Doc Task

Sz-Rung Shiang
National Taiwan University
No 1, Sec 4, Roosevelt
Road,Taipei, 10617 Taiwan
b97901031@ntu.edu.tw

Po-Wei Chou
National Taiwan University
No 1, Sec 4, Roosevelt
Road,Taipei, 10617 Taiwan
botonchou@gmail.com

Lang-Chi Yu
National Taiwan University
No 1, Sec 4, Roosevelt
Road,Taipei, 10617 Taiwan
b99901132@ntu.edu.tw

ABSTRACT

In this paper, we report out experiments on NTCIR-11 SpokenDoc&Query task for spoken term detection (STD) and spoken content retrieval (SCR). In STD, we consider acoustic feature similarity between utterances over both word and sub-word lattices to deal with the general problem of open vocabulary retrieval with queries of variable length. In SCR, we modify term frequency using expected term frequency in the vector space model (VSM) to deal with the errors in the speech recognition. In addition, we utilize three techniques to improve the relevance of the first-pass retrieval, that is, pseudo relevance feedback called Rocchio algorithm, query expansion using recurrent neural network language model (RNNLM), and lecture slide similarity feedback using random walk. Experiment results are shown for each task to indicate the improvement of the techniques we apply.

Categories and Subject Descriptors

H3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms

Algorithms, Performance, Experimentation

Keywords

NTICR-11, Spoken term detection, spoken content retrieval, dynamic time warping, graph based re-ranking.

Team Name

R531

Subtasks

Spoken Term Detection and Spoken Content Retrieval.

1. INTRODUCTION

There are generally two processing stages in the task of spoken term detection (STD). First, the audio is first transcribed into lattices, and then the retrieve engine will search through the lattices based on the user's query and return a list of relevant spoken utterances. When out-of-vocabulary (OOV) words are presented in the query, we need subword-based approaches because the aforementioned word lattices would be inadequate. Since the same words may have similar pronunciation thus similar acoustic feature sequences, acoustic feature similarity between utterances are sometimes useful in task of STD. We use both word and sub-word approaches for the first-pass retrieval, and then consider word/subword-based acoustic feature similarity for re-ranking.

In spoken content retrieval(SCR), a major problem is that some errors in ASR would eliminate relevance of the documents and the queries. To deal with that, instead of recognizing one-best transcriptions using ASR, an alternative way called expected term frequency is applied. It calculates the word appearance probability on lattice and induces recognition confidence into term frequency (TF). In other words, the frequency that a word appearing once in transcriptions is a probability score instead of 1. With expected term frequency, false negative, that the correct words not appearing in the recognition result, would be more likely to be included in the transcriptions. In addition, due to the diversity of words, some synonyms share the same semantics but are considered indepently and individually in vector space model (VSM). To tackle this problem, more words with semantic relationships should be taken into the queries. A generally used method is query expansion. We leverage the word representation from recurrent neural network (RNN) to model the semantic relationships between words, and add the releted word into queries. Moreover, to improve the relevance of SCR task, we leverage the first-pass relevance score from VSM, and further re-rank it using pseudo relevance feedback (PRF) and score propagation through slide segments similarity. With re-ranking, more words not in the original queries but actually with semantic similarity can be used to reformulate the new queries.

The rest of paper is structured as follow. The spoken term detection (STD) task, including the graph based re-ranking with acoustic similarity and corresponding evaluations, is described in section 2. The spoken content retrieval (SCR) task and its experiments are described in section 3. Conclusions are provided in section 4.

2. SPOKEN TERM DETECTION TASK

In this framework for spoken term detection, the utterances in the corpus were first transcribed into word or subword lattices by a ASR system. Every time when the user enters a new query (Q), the retrieve engine will search through the lattices and return a first-pass list xi ranked by the relevance score, which will be discussed later in 2.1. After that, the acoustic feature similarity $S(x_i, x_j)$ from the dynamic time warping (DTW) is used to re-rank the first-pass list to obtain the final retrieval result.

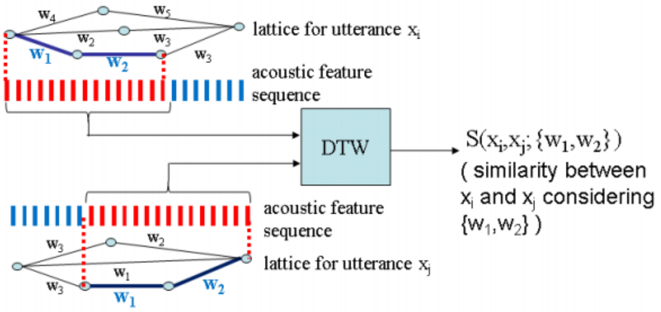
w4
w5
w2
w3
w1
w2
w3
lattice for utterance xi
acoustic feature sequence
DTW
S(xi,xj;{w1,w2})
(similarity between xi and xj considering {w1,w2})
acoustic feature sequence
w3
w2
w1
w3
w1
w2
lattice for utterance xj

Figure 1: An example of computing $S(x_i, x_j; \{w_1, w_2\})$, acoustic feature similarity between x_i and x_j considering the 2-gram $\{w_1, w_2\}$.

2.1 First-Pass Retrieval

The relevance score $R(x_i)$ used to obtain the first-pass list can be derived from either word or sub-word lattices. Relevance scores from word lattices are usually more accurate than those from subword lattices, but to be able to deal with out-of-vocabulary (OOV) queries, we need the latter. Given a query Q with more than one word, $Q = \{w_j, j = 1, 2, ..., N\}$, where w_j is the j-th word and N is the number of words in query Q, we can compute the relevance scores $R(x_i)$ for utterance x_i by counting the expected term frequencies. That is, we calculate the expected count $E_{k,n}$ for each possible n-gram $W_{k,n} = \{w_k, ..., w_{k+n-1}\}, k = 1, 2, ...N - n + 1$ as in (1b).

$$E_{k,n,x_i} = E[W_{k,n}|x_i] \tag{1a}$$

$$= \frac{\sum_{u \in W(x_i)} P(x_i, u) C(u, W_{k,n})}{\sum_{u \in W(x_i)} P(x_i, u)}, \tag{1b}$$

Then we combine those counts for all such n-grams to produce a score $R_n(x_i, Q)$ as in (2a), and finally obtain the relevance score of utterance x_i and query Q by integrating all those $R_n(x_i, Q)$ with weight a_n, as in (2b).

$$R_n(x_i, Q) = \sum_{k=1}^{N-n+1} E_{k,n,x_i} \tag{2a}$$

$$R(x_i, Q) = \sum_{n=1}^{N} a_n \cdot R_n(x_i, Q), \tag{2b}$$

2.2 Acoustic Feature Similarity

The acoustic feature similarity $S(x_i, x_j)$ between two retrieved utterances x_i and x_j is computed by dynamic time warping (DTW), which will later be used in two re-ranking methods to obtain second-pass retrieval results.

For every query Q and each n-gram $W_{k,n} = \{w_k, ..., w_{k+n-1}\}$ in Q, dynamic time warping (DTW) distance is first performed between the acoustic feature sequences corresponding to the subpaths in the lattices of x_i adn x_j for word hypotheses. An example is shown in Fig.1. This gives a DTW distance $d(x_i, x_j; W_{k,n})$ between x_i and x_j considering the n-gram $W_{k,n}$ in the query. The similarity $S(x_i, x_j; W_{k,n})$ can be easily converted from the DTW distance as in (3).

$$S(x_i, x_j; W_{k,n}) = 1 - \frac{d(x_i, x_j; W_{k,n}) - d_{min}}{d_{max} - d_{min}}, \tag{3}$$

Task	Word	Syllable	Word + Syllable
Baseline	0.4236	0.0359	0.4362
PRF	0.4256	0.0215	0.3969

Table 1: MAP Results of SQSTD

where d_{max} and d_{min} are the largest and smallest values of $d(x_i, x_j; W_{k,n})$ for all pairs of utterances in the first-pass list (i.e. we negate the distance and normalize it to the range of 0 to 1). Using the same approach as in (2a) and (2b), we integrate all such n-grams and then combine them together as the following:

$$S_n(x_i, x_j) = \sum_{k=1}^{N-n+1} S(x_i, x_j; W_{k,n}) \tag{4a}$$

$$S(x_i, x_j) = \sum_{n=1}^{N} b_n \cdot S_n(x_i, x_j), \tag{4b}$$

where b_n is another set of weighting parameters. The computation of $S(x_i, x_j)$ based on sub-word units is exactly the same as those based on words, except replacing each word w_i by a sub-word unit s_i.

2.3 Re-ranking & Second-pass

To obtain the second-pass retrieval results, we use pseudo-relevance feedback (PRF) and graph-based re-ranking as our re-ranking methods. Details of these 2 algorithms can be found in [4].

2.4 Experiments

The retrieval results are measured by mean average precision (MAP) and shown in the Table.1. We found that the pseudo-relevance feedback improve the performance of word-based retrieval but not the overall performance. Similar results can also be found in graph-based re-ranking. One reason to explain this phenomenon is that the performance of baseline result of syllable-based approach is too worse to be reliable. Therefore, the re-ranking methods, which are heavily based on the hypothesis regions obtained from the baseline results, does not improve the performance. The overall performance also depends on the weighting between word and syllable, which is chosen to be 0.1 in our experiments. A smarter and better way to choose the weighting parameter is needed.

3. SPOKEN CONTENT RETRIEVAL TASK

In the spoken content retrieval (SCR) task, we mainly focus on Slide-Group-Segment (SGS) subtask that demands for a relevance list of slide segments according to the query. We introduce expected term frequency into vector space model (VSM) and apply query expansion and two re-ranking techniques on the first-pass retrieval results. In this section, we first describe the expected term frequency, then explain the query expansion and re-ranking techniques to improve the relevance score, and show the evaluations of this task.

3.1 Expected Term Frequency and Vector Space Model

Spoken documents are prone to have more errors and information loss than text documents do. To adapt to the characteristic, we used lattices as input instead of one-best

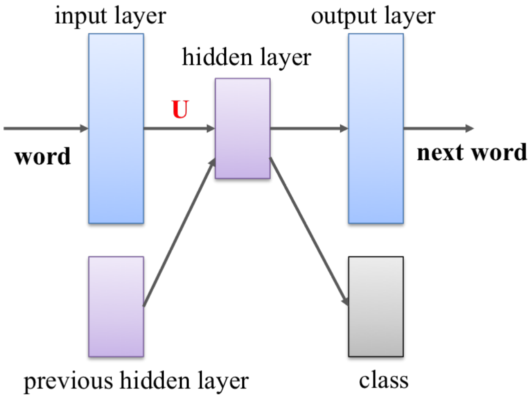

Figure 2: Illustration of recurrent neural network language model (RNNLM), where U is the word representation matrix we used to comprehend semantic relationships between words.

recognition transcriptions in order to consider the uncertainty of speech recognition. On the lattice graph, each path from start node to end node is a possible recognition result, and a path probability can be calculated through normalizing the score of acoustic model and language model. With a lot of paths (possible recognition results) with probability, the word appearance probability can be counted as the total probability of the paths across it on lattice, and that is the expected term frequency [3]. Here we apply the expected term frequency to replace simple term frequency (TF). That is to say, the frequency of word appearing once on lattice counts a probability score from 0 to 1 rather than 1. Higher probability of word appearance indicates that the words are more likely to be correctly recognized. With expected term frequency, false negative recognition errors can be more likely to be included in the VSM, thus preventing releted word from being excluded by ASR. We use term frequency-inverse document frequency (TF-IDF) in VSM with TF replaced by expected term frequency, and we filter out the words with probability less than 0.05.

After constructing TF-IDF for each slide-group segment and query, we calculate the relevance score through cosine similarity [8] as (5).

$$Relevance(segment, Q) = \frac{vec(segment) \cdot vec(Q)}{\|vec(segment)\| \, \|vec(Q)\|}, \quad (5)$$

where Q is the query and vec is the vector representation in TF-IDF/VSM. The segments with high score indicate high relevance to the queries, and vice versa.

3.2 Query Expansion

To include more semantic related words into queries, we apply word representation using recurrent neural network language model (RNNLM) [5],[6] on the query expansion part. The $L \times H$ weight matrix from the input layer to the hidden layer as described in Fig.2, where L is the lexicon size and H is the hidden layer size, can be taken as word representation matrix. In the word representation matrix, each word has a feature vector with the same dimension as hidden layer size H. It is noted that we set hidden layer size H as 80 in the following experiments.

For each word in the original query, we compute the similarity to other words in lexicon using cosine similarity:

$$Sim(w_1, w_2) = \frac{Fea(w_1) \cdot Fea(w_2)}{\|Fea(w_1)\| \, \|Fea(w_2)\|}, \quad (6)$$

where w_1 and w_2 are specific words and $Fea(\cdot)$ denotes the feature vector for word. We take the a similarity score threshold as 0.5 or up to 10 similar words for each word. That is to say, we only take at most 10 expansion words for each word in query. After extracting the similar words for query expansion, we add them to TF-IDF/VSM, and then calculate the relevance score through cosine similarity as (5).

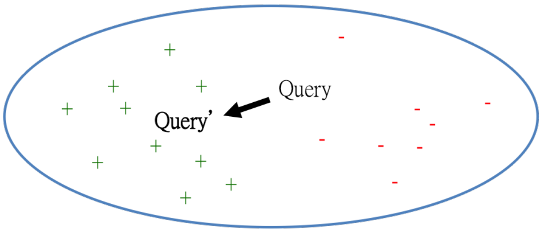

Figure 3: Illustration of rocchio pesudo relevance feedback. Query' is the modified query, "+" is the relevant document, and "-" is the irrelevant document.

3.3 Pseudo Relevance Feedback

In the first-pass retrieval result using the original query, some relevant documents could not be retrieved because of word diversity. Words in these documents do not equal to those in the query, even they have some similarity in semantics; therefore, the performance would degrade. To include more words related to queries in order to improve the retrieval performance, we applied Rocchio algorithm [2], which makes the words in first pass relevance documents to be included into the original query.

It is hypothesized that the documents retrieved first time using the original query are good enough, and some cooccurring words in the documents but not in the query probably have semantic relationships to the query words. In addition, words appearing in the most irrelevant documents (those with lowest score in the first-pass retrieval) may be totally off-topic of the intention of the query; therefore, we can also put negative weights on these words to modify query. That is to say, we not only make modified query close to the words in the first pass relevant documents, but also far away from the irrelevant documents in Rocchio relevance feedback, as shown in Fig. 3. It is also called pseudo relevance feedback because actually there is not any answer or human effort to judge the relevance of the documents as feedback to reformulate query; instead, we just assume that documents with highest/lowest score reveal the relevance/irrelevance to some extent. The new query is formulated as below:

$$\vec{Q_m} = (a \cdot \vec{Q_0}) + (\frac{b}{|D_r|} \cdot \sum_{D_j \in D_r} \vec{D_r}) - (\frac{c}{|D_{nr}|} \cdot \sum_{D_k \in D_{nr}} \vec{D_k}), \quad (7)$$

where $\vec{Q_m}$ and $\vec{Q_0}$ denotes vector space representation, such as TFIDF in this task, of the modified query and the original query correspondingly, D_r represents the relevant documents set and D_{nr} represents the irrelevant documents set. Parameters a, b and c are weights to balance the original

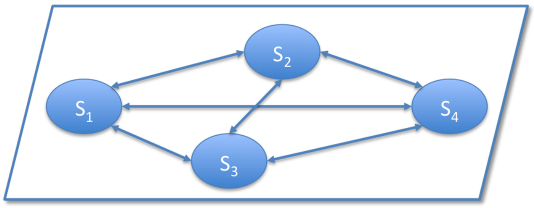

Figure 4: An example of random walk graph, where each node is a slide, and each two nodes have a edge with score (neighboring score).

query and the relevance feedback documents. Here we set $a = 1$, $b = 0.8$, $c = 0.1$, $|D_r| = |D_{nr}| = 5$.

3.4 Lecture Similarity Feedback

In this task, neighboring segments or slides sometimes share the similar idea of the lecture. The speaker probably talks about the same topic in one slide; therefore, these segments share the similar content even the words in segments are different. Moreover, sometimes speaker only mentions terms in the beginning of the lectures, and they would use pronouns instead afterwards. As a result, though some segments of slides share the similar idea, they fail to show the relevance of the segments to the queries using VSM. To capture this characteristic, we think up a way to propagate the relevance score to the neighboring segments using random walk [7] re-ranking techniques. In random walk, score propagates through the graph composed of nodes and edges as shown in Fig. 4. Each node here denotes one segment in slide, and each edge between two nodes has a weight to show the neighboring score as (8).

$$Score(s_i, s_j) = \frac{1}{|i-j|^2}, \tag{8}$$

where i and j are the index of the segments of a certain slide. If two segments s_i and s_j are not in the same slide, the neighboring score would be 0.

After constructing the graph, we propagate the first-pass retrieval score of each segment to the neighboring segment using random walk:

$$F^{(t+1)} = (1-\alpha) \cdot F^{(0)} + \alpha \cdot L^T \cdot F^{(t)}, \tag{9}$$

where $F(t)$ denotes the t-th iteration of the score vector of segments, $F(0)$ denotes the initial score vector of the segments (here we used first-pass relevance score) ,L is the normalized edge matrix of neighboring score as (8), and α is a parameter to balance the initial score and the graph propagation with value between 0 and 1. We iteratively update the score as (9) until the score vector converges as (10), and we take the convergent score vector as the re-ranking relevance score. In addition, higher the score, higher the relevance score.

$$F^{(T)} = (1-\alpha) \cdot F^{(0)} + \alpha \cdot L^T \cdot F^{(T)}. \tag{10}$$

Through the score propagation, nodes would give and take some score from their neighboring nodes; therefore, they would more likely to have similar score. In other words, the segment with more neighboring segments with high score are more likely to have high score, and vice versa.

	One-best	Expected TF
(a) Raw	0.1056	0.1294
(b) +Hiragana filter	0.1066	0.1351
(c) Dictionary form	–	0.1537
(d) +Hiragana filter	–	0.1543

Table 2: MAP Results using one-best transcriptions and expected term frequency from lattice.

	One-best	Expected TF
(a) Baseline	0.1056	0.1294
(b) Rocchio	0.1165	0.1360
(c) RNNLM query expansion	0.1058	0.1294
(d) Lecture similarity	0.1079	0.1319

Table 3: MAP Results of re-ranking techniques and query expasion on both one-best transcriptions and expected term frequency from lattice.

3.5 Experiments

In the SCR subtask, we used the word level recognition results with match model. More details about the corpus and task can be found in the overview paper of NTCIR-11[1]. We first extract some stopwords from the top 100 words with highest product of term frequency (TF) and document frequency (DF), and then we remove the stopwords from our recognition results. In the Japanese decoder, both the pronunciation and dictionary form are given; therefore, we take both the raw results and dictionary form results as comparison. The RNNLM word representation used in this task is trained on one-best transcriptions of all the lectures in NTCIR-11 SpokenQuery&Doc corpus with lexicon size as 6542 words and hidden layer size as 80. Mean average precision (MAP) is used as the evaluation metrics.

In Table. 2, we show MAP results of first-pass retrieval using both one-best recognition transcripts and expected term frequency from lattice. It is noted that the one-best means that we assign 1 for each word appearing in the transcriptions. Row (a) and (c) show the results on raw form and dictionary form correspondingly. "Hiragana filter" in both row (b) and row (d) means that we only extract the words that not all elements are hiragana, because most of the important words in the lectures are characters in Chinese or katakana. It is shown that the expected term frequency can effectively improve the performance over 2-3 percent on MAP, since more possible words are considered into the queries of the retrieval task. Moreover, the usage of hiragana filter and dictionary form also make significant improvement.

In Table. 3, we compare the results of query expansion, Rocchio algorithm, and lecture similarity feedback with the baseline, and we conduct the experiments on the raw form recognition results using expected term frequency in VSM. Baseline shown in row (a) is the same as the one in Table. 2. We set α in random walk for lecture similarity feedback as 0.1 in this task. Promising results reveal that Rocchio algorithm as row (b) and lecture similarity feedback as row (d) can make progress on both one-best and lattice recognition results. However, results of RNNLM query expansion only show little increasing MAP. There are two possible reasons why RNNLM query expansion has limited improvement. First, the training corpus for RNNLM here is too small, thus insufficient and imbalance data lead to bad

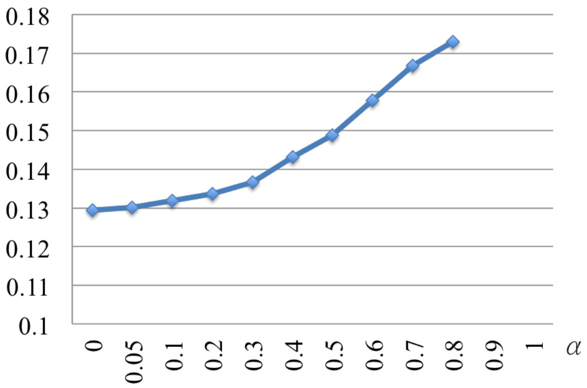

Figure 5: MAP results of different α in random walk for lecture similairty feedback. The horizontal axis is α and the vertical axis is MAP evaluation.

performance of word representation. Another reason may be that recognition errors in ASR induce confusing word representation; therefore, some words for query expansion actually do not have such semantic relationships at all.

In Fig.5, MAP results with different parameter α in random walk for the lecture similarity feedback are shown. Experiment is conducted on the raw form recognition results with VSM using expected term frequency. As the parameter α increasing, the MAP performance gets improvement, which shows the neighboring segments in slides share the similar idea. It is noted that the performance with $\alpha = 1$ equals to the first-pass retrieval result.

4. CONCLUSION

In STD, though the combination of word and sub-word can be helpful in performance in some cases, a further study on the algorithm to determine the weighting parameter is required. In SCR, we conduct the preliminary experiments and achieve improvement over the results with the modified VSM, query expansion, and re-ranking techniques using Rocchio algorithm and lecture similarity feedback. Greater improvement is possible if all of the methods can be combined or jointly considered.

5. REFERENCES

[1] T. Akiba, H. Nishizaki, H. Nanjo, and G. J. F. Jones. Overview of the ntcir-11 spokenquery&doc task. In *Proceedings of NTCIR-11, Tokyo, Japan*, 2014.

[2] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 292–300, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[3] H. Lee, S. Shiang, C. Yeh, Y. Chen, Y. Huang, S. Kong, and L. Lee. Spoken knowledge organization by semantic structuring and a prototype course lecture system for personalized learning. *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 22(5):881–896, 2014.

[4] H.-y. Lee, P.-w. Chou, and L.-S. Lee. Open-vocabulary retrieval of spoken content with shorter/longer queries considering word/subword-based acoustic feature similarity. In *INTERSPEECH*, 2012.

[5] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. H. Cernocky. Rnnlm - recurrent neural network language modeling toolkit. IEEE Automatic Speech Recognition and Understanding Workshop, December 2011.

[6] T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. In *SLT*, 2012.

[7] E. Minkov and W. W. Cohen. Improving graph-walk-based similarity with reranking: Case studies for personal information management. *ACM Trans. Inf. Syst.*, 29(1):4:1–4:52, Dec. 2010.

[8] A. Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43, 2001.