SPOKEN DOCUMENT RETRIEVAL BY DISCRIMINATIVE MODELING IN A HIGH DIMENSIONAL FEATURE SPACE

Takanobu Oba[†‡]*, Takaaki Hori*[†]*, Atsushi Nakamura*[†] *and Akinori Ito*[‡]

[†]NTT Communication Science Laboratories, NTT Corporation, Japan
[‡] Graduate School of Engineering, Tohoku University, Japan
{oba.takanobu, hori.t, nakamura.atsushi}@lab.ntt.co.jp, aito@spcom.ecei.tohoku.ac.jp

ABSTRACT

This paper proposes discriminative modeling in a high dimensional feature space for spoken document retrieval (SDR). To estimate the parameters of a high dimensional model properly, a large quantity of data is necessary, but there is no such large corpus for document retrieval. This paper employs two approaches to overcome this problem. One is a reranking approach. A baseline system first gives each document a score and then the score is compensated by employing a high dimensional model. The other approach is automatic query generation. A large number of queries are automatically generated and used for parameter estimation. Our experimental result shows that our proposed method can greatly improve SDR performance.

Index Terms— Spoken document retrieval, Discriminative model, Linear model

1. INTRODUCTION

In recent years, it has become easier to record spoken documents and store them in computers. The quantity of spoken document data continues to increase rapidly. As a result, spoken document retrieval (SDR), which is a task that involves finding documents relevant to a user's query, has been gaining in importance.

In text or spoken document retrieval, the relevance of a document to query is estimated based on a predefined distance measure such as cosine distance in a certain feature space [1, 2] or a generative statistical language model [3, 4, 5, 6]. The problem with these approaches is that the models do not take into account the relationship between the query and other documents when calculating the relevance of the document to the query.

Today, discriminative models such as linear models are frequently used in many tasks including natural language processing, and often perform better than generative models, or can improve the performance by using both models together [7, 8, 9, 10, 11]. This trend naturally leads us to use discriminative models in document retrieval. In previous research on document retrieval, some weak learners were prepared and merged through the weighted sum of their inference scores [12, 13, 14]. These methods can be regarded as a linear model that has the form of the inner product of a model parameter vector and a feature vector. For example, in a discriminative approach for document retrieval proposed by Nallapati et al. [12], some different types of scores are respectively accumulated over terms that appear both in a query and a document. Each of the scores is related to term frequency (tf), a combination of tf and inverse document frequency (idf). Finally, the accumulated scores are merged where the weights are trained using a discriminative learning method. Meng et al. have employed accumulated probabilities over multiple co-occurrence terms with word, sub-word and character [14]. These methods are, thus, designed to be used with a low dimensional feature space.

Recent advances in machine learning technology have enabled us to estimate parameters properly even in a high dimensional feature space, when given a large quantity of data. A variety of features that is useful for document retrieval becomes available by the use of a linear model in high dimensional feature space.

This paper proposes a discriminative linear model in a high dimensional feature space for SDR. Separate statistics of terms, e.g., words, sub-words, etc., rather than an accumulated value over multiple terms, are directly employed as feature elements. For example, sub-word features would be effective for the problem of out-of-vocabulary (OOV). Confidence measure features would be able to mitigate the effects of speech recognition errors.

There are two problems as regards achieving an SDR with a high dimensional linear model. One is the lack of training data, which are sets consisting of a query and corresponding relevant document labels. A large quantity of training data is necessary to estimate parameters in a high dimensional feature space properly. However, it is expensive to make queries and their relevant document labels manually, and there is no corpus that includes a large number of training data. The other problem relates to index size. Many features are obtained from each document in the proposed framework. Thus, the index table tends to be large.

To resolve the first problem, i.e. the difficulty of parameter estimation when there is a lack of training data, we employ two methods. One is a reranking approach. Documents are first ranked based on a conventional baseline document retriever, and then reranked by a discriminative linear model through combining the two types of scores. The other is the automatic generation of training queries. A language model is made from a document and used to generate queries. The document that is used to make the language model is regarded as a pseudo relevant document for discriminative training.

The second problem, namely that of index size, might not be serious if many computers are available in parallel or if features are dynamically extracted from each document (recognition result in SDR) as in [14]. The second problem is dealt with to expand the availability of our proposed method. We employ a model shrinkage technique for linear models to make the index table compact. This is a method for removing parameters whose impact is small. The index table becomes small by removing features that correspond to the removed parameters.

As a first step, the reranking approach is evaluated in terms of SDR performance, where there are few OOV words and speech recognition errors. Our experimental results show that our proposed

method greatly outperforms a conventional tf-idf based baseline system, which measures the relevance of a document to a query by the cosine distance of their tf-idf vectors, and can realize a small index table by shrinking the model.

2. LINEAR MODEL BASED RERANKER

Let a document be D and a query be Q. And let $f_0(D,Q)$ be the similarity (score) between D and Q measured by using a baseline system. Given a parameter vector $\mathbf{a}$, reranking is undertaken based on the following criterion.

$$a_0 f_0(D,Q) + \mathbf{a}^\top \mathbf{f}(D,Q) \tag{1}$$

$\mathbf{f}(D,Q)$ is a feature vector and a_0 is a scaling constant.

Each element of $\mathbf{f}(D,Q)$ is typically the count of an n-gram in D that activates only when the n-gram also appears in Q. Let the count of an n-gram x_k in a document D be $c(x_k,D)$. Also, $c(x_k,Q)$ is the count of x_k in Q. The k-th element of the feature vector $f_k(D,Q)$ is $c(x_k,D)$ if $c(x_k,Q) > 0$, otherwise 0. In this paper, we use word unigrams, part-of-speech (POS) unigrams, character uni-, bi-, trigrams within each word, and phoneme uni-, bi-, trigrams within each word.

The purpose of training is to estimate the value of $\mathbf{a}$. a_0 is decided using development data in this work, although a_0 can also be trained. Document retrieval is often regarded as a binary classification problem in a discriminative training scenario [12]. Suppose that $r(D,Q)$ denotes whether or not D is a relevant document for Q. $r(D,Q) = 1$ if they are relevant, and $r(D,Q) = -1$ otherwise. Given a set of pairs consisting of a document and a query, and their reference labels r, the parameter vector $\mathbf{a}$ must be estimated so that the sign of $\mathbf{a}^\top \mathbf{f}(D,Q)$ corresponds to the sign of $r(D,Q)$ for most pairs of (D,Q).

In this work, we chose the log maximum entropy model for document retrieval. Two parameter vectors $\mathbf{a}_{+1}$ and $\mathbf{a}_{-1}$ are estimated so as to minimize

$$-\sum_{(D,Q)} \log \frac{\exp(\mathbf{a}_{r(D,Q)}^\top \mathbf{f}(D,Q))}{\sum_{r' \in \{+1,-1\}} \exp(\mathbf{a}_{r'}^\top \mathbf{f}(D,Q))} + \frac{||\mathbf{a}_{+1}||_2^2 + ||\mathbf{a}_{-1}||_2^2}{C}. \tag{2}$$

And then parameter vector $\mathbf{a}$ is regarded as $\mathbf{a}_{+1} - \mathbf{a}_{-1}$ in the reranking stage. $||x||_2^2$ is L2-norm and C is a constant. We also used the L-BFGS algorithm [15] to solve the minimization problem.

Note that parameter vector $\mathbf{a}$ does not have any arguments, i.e. the same $\mathbf{a}$ value is commonly used for any set of (D,Q). Discriminative training estimates the $\mathbf{a}$ value so that the relevance or irrelevance of each set of (D,Q) is correctly predicted.

3. AUTOMATIC QUERY GENERATOR

To generate queries automatically, we employed two types of back-off n-gram language models. One is the background language model P_{B}, which is trained using the whole document collection. The other is a document specific language model P_{D}. Queries are generated randomly based on the linear interpolated model, $P_{\lambda,\mathrm{D}} = \lambda P_{\mathrm{D}} + (1-\lambda) P_{\mathrm{B}}$. The query length is also decided randomly according to the interpolated model. Hence, we assume that each document is divided into sentences or sentence-like units before the language models are trained.

The background model includes all the terms. In contrast, the specific model P_{D} includes only some of the terms. Generally, queries include terms that do not appear in the relevant documents. The background model is used to simulate this characteristic of queries.

This automatic query generation is undertaken to obtain a large number of training samples. A variety of types of queries should be generated. The use of a specific λ value is contrary to this idea. Therefore, we employed several values, 0.9, 0.8, 0.7, 0.6 and 0.5.

As mentioned in the introduction, when a query is generated according to $P_{\lambda,\mathrm{D}}$, document D is regarded as relevant to the query. The other documents are regarded as irrelevant to the query. In other words, with each query, one document is the positive sample for discriminative training and the others are negative samples. Such an imbalance generally prevents proper model training. To avoid this problem, we randomly chose one negative sample (document) with each query.

4. MODEL SHRINKAGE

This section describes a method for eliminating redundant parameters from a simple linear model, which has the form of the inner product of a model parameter vector and a feature vector. While there are pruning methods for back-off n-gram language models [16, 17], this method is for linear models.

Consider converting an n-dimensional original linear model $\mathbf{a} \in \mathbf{R}^n$ to an m-dimensional model $\hat{\mathbf{a}}_m \in \mathbf{R}^m$ $(m < n)$ that provides results that are as similar as possible. Assume that $\mathbf{a}$ is converted according to linear matrices R_m and B as

$$\hat{\mathbf{a}}_m = R_m^\top B \mathbf{a}. \tag{3}$$

$R_m = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_m]$ is a matrix designed to permute elements in order of importance and to choose only the m top elements, while removing the other elements. $\mathbf{r}_k$s' are n-dimensional orthogonal bases, and only one element of $\mathbf{r}_k$ is 1 and all of the other elements are zero. B is an $n \times n$ diagonal matrix to scale the values of the parameter elements of $\mathbf{a}$. The diagonal element vector of B is denoted as $\mathbf{b}$.

Suppose that $\mathbf{a}$ is converted so as to minimize the following square error

$$\sum_{\mathbf{f} \in \mathcal{D}} (\mathbf{a}^\top \mathbf{f} - \hat{\mathbf{a}}_m^\top R_m^\top \mathbf{f})^2 \tag{4}$$

(approximately), where $\mathcal{D}$ is a data set. To obtain a simple analytical solution, $\mathbf{f}$ is assumed to be sparse. As a result, the square error is approximately converted into

$$\epsilon_n(B) = \sum_{\mathbf{f} \in \mathcal{D}} ||F\mathbf{a} - FB\mathbf{a}||^2, \tag{5}$$

when $m = n$. F is a diagonal matrix whose diagonal is $\mathbf{f}$.

We also assume that $\mathbf{b}$ is sparse since the purpose of model shrinkage is to convert most elements of $\mathbf{a}$ to 0. Introducing an L1-norm, the solution of $\epsilon_n(B) + c|\mathbf{b}|$ is given as

$$b_k = \max\left(0, 1 - \frac{c}{2a_k^2 \sum_{\mathbf{f} \in \mathcal{D}} f_k^2}\right). \tag{6}$$

c is a constant and we describe how to decide its value.

On the other hand, $\epsilon_n(B)$ can be rewritten as

$$\epsilon_n(B) = \epsilon_{n-1}(B) + a_k^2 b_k (2 - b_k) \sum_{\mathbf{f} \in \mathcal{D}} f_k^2 \tag{7}$$

$$= \epsilon_{n-1}(B) + \left\{ q_k + \frac{-c^2}{4q_k} \right\} \tag{8}$$

$$q_k = a_k^2 \sum_{\mathbf{f} \in \mathcal{D}} f_k^2. \tag{9}$$

The term in brackets in the recurrence formula (8) can be regarded as

having an impact when we remove one element from $\mathbf{a}$. Therefore, the elements of $\mathbf{a}$ should be permuted in the decreasing order of this term. Since this term is a monotonic increasing function with $q_k(> 0)$, the elements of $\mathbf{a}$ should be permuted in order of q_k. We represent a parameter element index in this order as K_j.

Assume that the c value is decided so that the top m elements of $\hat{\mathbf{a}}_n$ have nonzero values and the others are zero. In this case, $c = 2q_{K_{m+1}}$, hence

$$b_{K_j} = 1 - \frac{q_{K_{m+1}}}{q_{K_j}} \tag{10}$$

for $j \leq m$.

As the result, our proposed shrinkage method merely requires us to calculate q_k for all k using a data set $\mathcal{D}$, and then to sort them in the decreasing order, and the parameter elements of $\mathbf{a}$ is also sorted in this order, converting as

$$\hat{a}_{K_j} = a_{K_j}\left(1 - \frac{q_{K_{m+1}}}{q_{K_j}}\right) \tag{11}$$

where the shrinkage model size m is given. In this work, the data set $\mathcal{D}$ consists of feature vectors that are used for the discriminative training of a model for SDR.

5. EXPERIMENT

We used the CSJ-Spoken document retrieval test collection [18] for our experiments. The CSJ (Corpus of Spontaneous Japanese) [19] is a Japanese lecture corpus, containing speech data and their transcriptions, and this document retrieval collection contains 39 queries and their relevant document indexes to 2702 spoken documents in the CSJ. The queries are in sentence form and designed manually so that each query is relevant to certain parts of documents, seeing all the 2702 documents. The relevant segments vary in length from short to long.

The 39 queries were divided into groups of 9 and 30. The former set was a development set to decide the a_0 value. To show that our proposed method provides a better result without special efforts, the hyperparameters, a_0 and the regularization constant C in equation (2), were roughly decided. The a_0 value was chosen from 1, 10, 100, 500 and 1000. The C value was set to 1.

Our baseline SDR system measures the similarity between a document D and a query Q by the cosine distance of word unigram tf-idf vectors. Although the test collection also contains 50-best recognition results for the 2702 documents to eliminate the effects of speech recognition quality, 1-best hypotheses were used for our experiments.

Using our proposed automatic query generation method, we generated 50 queries with each document. This process was run 5 times with different λ values, 0.9, 0.8, 0.7, 0.6, 0.5. Hence, the total number of queries for training was $50 \times 2702 \times 5$. The language models were trained using 1-best sentences.

Note that discriminative training does not require indexes over entire documents, but instead requires feature elements with a nonzero value, i.e. terms that occur in both a document and a query. Hence, discriminative training can be performed using a machine with a relatively small memory. For phoneme features, a morpheme analyzer was used to add phonemes to each word. It was employed for both written queries and spoken recognized documents.

Table 1 compares the baseline SDR performance with that of our proposed method in terms of 3 types of criteria, mean average precision (MAP), R-precision and normalized discounted cumulative gain (nDCG) [20] for the top 5 ranks. The values in the brackets denote the performance for the development (9 queries) set. a_0 is 100 under all the conditions in table 1.

Table 1. SDR performance. Baseline vs w/ Discriminative model. Performance in bracket is for development set (9 queries).

	MAP	R-prec.	nDCG@5
Baseline	0.32 (0.23)	0.31 (0.21)	0.40 (0.17)
+DLM(word)	0.43 (0.32)	0.43 (0.32)	0.52 (0.32)
+POS	0.42 (0.33)	0.39 (0.32)	0.53 (0.37)
+character	0.43 (0.38)	0.41 (0.36)	0.54 (0.40)
+phoneme	0.40 (0.36)	0.38 (0.33)	0.52 (0.39)

Table 2. Insensitivity for scaling constant. MAP values for development set (9 queries) and evaluation set (30 queries) with different values of scaling constant a_0.

a_0	1	10	100	500	1000
9 queries	0.19	0.22	0.32	0.27	0.25
30 queries	0.20	0.23	0.42	0.37	0.34

SDR performance was greatly improved when we applied the reranking approach using the discriminative model with word unigram count features. While we used tf-idf for the baseline, tf and idf are usually memorized separately. Since the term 'count' means 'term frequency (tf)', this discriminative model utilizes information that is used in the baseline system. This means that the index size is unchanged even if the discriminative model with word unigram features is used for reranking the baseline result.

Additional use of POS, character and phoneme features did not greatly affect SDR performance. This experiment was not designed for problems arising from OOV words and speech recognition errors. These features might be effective under such conditions.

Table 2 shows relationships between a_0 and MAP. $a_0 = 100$ was best with both the small set (9 queries) and the large set (30 queries). $a_0 = 500$ also provided a better result than the baseline (see table1). There may be better values around 100. This experimental result indicates that the reranking approach outperforms the baseline, even if the a_0 value is decided roughly.

Next, we evaluated the size of the index table needed to store all the documents. The index table size largely depends on the data structures used to hold the indexes. Hence, the index table size is measured as the number of feature elements with a nonzero value over the 2702 documents. For example, if a certain word appears in 100 documents, the index size for that word is 100.

Usually, multiple (n-best) recognition sentences or lattices are used for SDR. In such a situation, we do not yet know what kind of features are effective for high dimensional discriminative models. The purpose of this experiment is to show the trend as to how many indexes are required depending on feature types.

For this purpose, we use the above 4 types of features, although the additional use of POS, character and phoneme features does not improve SDR performance under our experimental conditions. POS simulates a macro feature, for example, that indicates whether or not a word is a content word. Character n-grams simulate sub-words. Phoneme n-grams are examples of short time term features.

Table 3 shows MAP values and index sizes for some shrunken models. The MAP value is given at the top of each cell and the index size is given below it. 'Model size' indicates the number of parameter elements with a nonzero value. Full models have at least 27,000 parameter elements.

Table 3. MAP and index size for shrunken models. Baseline index size is about 2,062k.

model size	500	1000	2000	full
DLM(word)	0.38	0.43	0.42	0.43
	0	0	0	0
+POS	0.38	0.41	0.41	0.42
	89k	148k	188k	199k
+character	0.36	0.40	0.43	0.43
	288k	535k	923k	1,572k
+phoneme	0.29	0.35	0.39	0.40
	793k	1,439k	2,430k	4,821k

The index sizes in table 3 denote the number of indexes that are needed in addition to the baseline indexes. Since word unigrams are commonly used for the baseline system, the index sizes for word models are 0. The baseline index size was about $2m$.

Models with 2000 dimensions performed equally well as the full models. This means that most features are unnecessary and they can be removed by using our proposed model shrinkage method. POS features increased the index size slightly. Even when the full model was used, the index size was small. Thus, these kinds of features are easy to use for SDR. Character n-gram features increased the index size at a certain level. However, by shrinking the model, the index size was reduced to about half of the baseline index size with the 2000 dimensional model and one-fourth of the baseline index size with the 1000 dimensional model. Therefore, sub-word features could be used to mitigate the OOV problem without storing a large number of indexes. In contrast, phoneme features greatly increased the index size, even if the model was shrunk to a small one. Hence, these kinds of features must be carefully used, if a small index size is required.

6. CONCLUSION

A high dimensional modeling method for SDR was proposed in this paper. To overcome the problem of a lack of training data, we employed the reranking approach and an automatic query generation method. In addition, we employed a model shrinkage method to realize a compact index table. Our experimental results show that our proposed method can improve SDR performance, although the baseline system is simple and the evaluation set is small. In future work, we will employ our proposed method with a state-of-the-art SDR system, using multiple recognition hypotheses, and evaluate it using a large test set, assuming the presence of out-of-vocabulary words and speech recognition errors.

7. REFERENCES

[1] Gerard Salton, Anita Wong, and Chung-Shu Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, 1975.

[2] Jonathan Mamou, David Carmel, and Ron Hoory, "Spoken document retrieval from call-center conversations," in *Proceedings of ACM SIGIR*, 2006, pp. 51–58.

[3] Jay M. Ponte and W. Bruce Croft, "A language modeling approach to information retrieval," in *Proceedings of SIGIR*, 1998, pp. 275–281.

[4] W.Bruce Croft, "Language models for information retrieval," in *Proceedings of the 19th International Conference on Data Engineering*, 2003, pp. 3 –7.

[5] John Lafferty and Chengxiang Zhai, "Document language models, query models, and risk minimization for information retrieval," in *Proceedings of ACM SIGIR*, 2001, pp. 111–119.

[6] Xinhui Hu, Ryosuke Isotani, and Satoshi Nakamura, "Spoken document retrieval using topic models," in *Proceedings of the 3rd International Universal Communication Symposium*, 2009, pp. 400–403.

[7] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of Machine Learning*, 2001, pp. 282–289.

[8] Michael Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of EMNLP*, 2002, pp. 1–8.

[9] Franz Josef Och and Hermann Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of ACL*, 2002, pp. 295–302.

[10] Brian Roark, Murat Saraclar, and Michael Collins, "Discriminative n-gram language modeling," *Computer Speech and Language*, vol. 21, no. 2, pp. 373–392, 2007.

[11] Takanobu Oba, Takaaki Hori, and Atsushi Nakamura, "Round-robin discrimination model for reranking ASR hypotheses," in *Proceedings of Interspeech*, 2010, pp. 2446–2449.

[12] Ramesh Nallapati, "Discriminative models for information retrieval," in *Proceedings of SIGIR*, 2004, pp. 64–71.

[13] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon, "Adapting ranking SVM to document retrieval," in *Proceedings of SIGIR*, 2006, pp. 186–193.

[14] Chao-Hong Meng, Hung-Yi Lee, and Lin-Shan Lee, "Improved lattice-based spoken document retrieval by directly learning from the evaluation measures," in *Proceedings of ICASSP*, 2009, pp. 4893–4896.

[15] Dong C. Liu and Jorge Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[16] Andreas Stolcke, "Entropy-based pruning of backoff language models," in *Proceedings of DARPA News Transcription and Understanding Workshop*, 1998, pp. 270–274.

[17] Jianfeng Li, Haifeng Wang, Dengjun Ren, and Guohua Li, "Discriminative pruning of language models for Chinese word segmentation," in *Proceedings of the Association for Computational Linguistics*, 2006, pp. 1001–1008.

[18] Tomoyosi Akiba, Kiyoaki Aikawa, Yoshiaki Itoh, Tatsuya Kawahara, Hiroaki Nanjo, Hiromitsu Nishizaki, Norihito Yasuda, Yoichi Yamashita, and Katunobu Itou, "Construction of a test collection for spoken document retrieval from lecture audio data," *Journal of Information Processing*, vol. 17, pp. 82–94, 2009.

[19] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara, "Spontaneous speech corpus of Japanese," in *Proceedings of ICLRE*, 2000, pp. 947–952.

[20] Kalervo Järvelin and Jaana Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions of Information Systems.*, vol. 20, pp. 422–446, 2002.