INTERSPEECH 2009 BRIGHTON

An indexing weight for voice-to-text search

Chen Liu

Applied Research and Technology Center, Motorola, Schaumburg, IL 60196, USA

Chen.Liu@Motorola.com

Abstract

The TF–IDF (term frequency–inverse document frequency) weight is a well-known indexing weight in information retrieval and text mining. However, it is not suitable for the increasingly popular voice-to-text search, as it does not take into account the impact of voice in the search process. We propose a method for calculating a new indexing weight, which is used as guidance for selection of suitable queries for voice-to-text search. In designing the new weight, we combine prominence factors from both the text and acoustic domains. Experimental results show significant improvement in the average search success rate with the new indexing weight.

Index Terms: voice-to-text search, indexing weight, inter-phoneme distance, speech recognition

10.21437/Interspeech.2009-566

1. Introduction

The TF-IDF (term frequency-inverse document frequency) weight is long established in term weighting and document ranking [1]. It is a statistical metric used for evaluating how important a word is to a document in a collection of documents or corpus. Queries designed with TF-IDF enable a fast retrieval of the documents. Use of TF-IDF for determining document queries was recently examined through experiments by Ramos [2]. However, it is not equally suitable for the increasingly popular voice-to-text search, as it does not take into account the impact of voice in the search process. For example, there are words that tend to result in low voice recognition success due to their pronunciations that are easily confused with other words in the corpus. The recognition task is very challenging for a large corpus with abundant distinct words.

In this paper, we propose a method for calculating a new indexing weight. The weight is used as guidance for selection of suitable queries for voice-to-text search. Specifically a database can be organized with a minimal set of index words for each document, optimized for voice-to-text search. This is especially useful for implementation on embedded devices where limited resource is available. Another typical use of the weight is for evaluating and comparing the efficiency of a voice-to-search system, with known optimal query words.

In designing the new weight, we combine the discriminatory factors from both the text and acoustic domains. The conventional TF-IDF is used in the text domain (Sec. 2.1), while in the acoustic domain we propose a concept of pronunciation prominence. It is established based on an inter-word pronunciation distance, which is in turn derived from an inter-phoneme distance (IPD). There are many approaches to estimation of the inter-phone distance. Dependent on the availability of real speech data, one family is data driven [3][4][5] and another family is phonetic based [6]. We describe both approaches in Sec. 2.2. Our experimental results, given in Sec. 3, show significant improvement in the success rate of search by using the new indexing weight.

2. Method

2.1. Prominence metric in the text domain

Any state-of-the-art prominence metrics in the text domain can be utilized, and we adopt the conventional TF-IDF in the study. The documents in the corpus undergo preprocessing such as garbage cleaning, punctuation cleaning, stemming, and stopword filtering. Each document is converted into a word vector. The word vectors are used for calculation of TF (term frequency) and IDF (inverse document frequency). The TF-IDF formula used in the experiment is one of the most general versions. Specifically, the term frequency is the normalized count of a term t_m within a particular document d_q

$$\mathrm{TF}_{mq} = \frac{n_{mq}}{\sum_k n_{kq}} \tag{1}$$

where n_{mq} is the number of occurrences of the term t_m in document d_q, and the denominator is the number of occurrences of all terms in document d_q. The inverse document frequency of a term t_m is

$$\mathrm{IDF}_m = \ln \frac{|D|}{|\{d_q : t_m \in d_q\}|} \tag{2}$$

where $|D|$ is the total number of documents in the corpus, while the denominator represents the number of documents where the term t_m appears. The TF-IDF weight is

$$(\mathrm{TF\text{-}IDF})_{mq} = \mathrm{TF}_{mq} \cdot \mathrm{IDF}_m \tag{3}$$

which measures how important a term t_m to a document d_q in the corpus.

2.2. Metrics of pronunciation prominence

Our pronunciation prominence metric can be obtained from either the data-driven based inter-phoneme distance (IPD) or phonetic based IPD.

2.2.1. Data-driven IPD

In this approach, we assume that a certain amount of speech data is available in advance for a phonemic recognition test. Thus the phonemic confusion matrix can be derived from the result of speech recognition using phoneme-loop grammar. If the phonemic inventory is denoted as $\{p_i \mid i = 1,\ldots,I\}$, where I is the total number of phonemes in the inventory. Each element in the confusion matrix is denoted by $C(p_j \mid p_i)$, which represents the number of instances of a phoneme p_i being recognized as p_j. In a general sense, the pause and

silence models can be included in the phonemic inventory. Hence the confusion matrix can also provide information about deletion (when p_j = pause or silence) or insertion (when p_i = pause or silence) of each phoneme. The tendency of a phoneme p_i being recognized as p_j can be defined as

$$d(p_j \mid p_i) = \frac{C(p_j \mid p_i)}{\sum_{j'=1}^{I} C(p_{j'} \mid p_i)} \tag{4}$$

Note that this quantity characterizes closeness between two phonemes p_i and p_j to some extent, but it is not a distance measure in a strict sense because it is not symmetric, i.e.,
$d(p_j \mid p_i) \neq d(p_i \mid p_j)$.

2.2.2. Phonetic-based IPD

This approach estimates the inter-phoneme distance solely from the phonetic knowledge. Characterization of quantitative relationship between phonemes in a purely phonetic domain has been well researched [6][7][8]. We adopt the phonetic distance metric recently developed by Liu and Melnar [9]. It represents each phoneme with a vector with each of its elements corresponding to a distinctive phonetic feature, i.e.,

$$\mathbf{f}(p_i) = [v_i(l)]^T, \qquad l = 1, \ldots, L, \; i = 1, \cdots, I \tag{5}$$

where the vector contains a total L elements or features, each element taking value of either one for a feature presence or zero for a feature absence. Recognizing the difference of features in contribution to the phonemic distinction, the features are modified with a weight factor. The weight is derived from the relative frequency of each feature in the language. Let $c(p_i)$ denote the occurrence count of a phoneme p_i, then the frequency of each feature l contributed by the phoneme p_i is $c(p_i)v_i(l)$, and the frequency of each feature l contributed by all the phonemes is $\sum_{i=1}^{I} c(p_i)v_i(l)$. The weights derived from all the phonemes in the language are

$$\mathbf{W} = \mathrm{diag}\{w(1), \cdots, w(l), \cdots, w(L)\} \tag{6}$$

where the weight for each specific feature l is

$$w(l) = \frac{\sum_{i=1}^{I} c(p_i)v_i(l)}{\sum_{l'=1}^{L}\sum_{i=1}^{I} c(p_i)v_i(l')} \qquad l = 1, \cdots, L \tag{7}$$

where diag(vector) represents a diagonal matrix with elements of the vector as the diagonal entries. The estimated phonemic distance between the two phonemes p_i and p_j, is calculated as

$$d(p_j \mid p_i) = \left\| \mathbf{W}[\mathbf{f}(p_i) - \mathbf{f}(p_j)] \right\|_1 = \sum_{l=1}^{L} w(l) \left| v_i(l) - v_j(l) \right| \tag{8}$$

where $i = 1, \cdots, I$, and $j = 1, \cdots, I$. The distance between a phoneme and silence or pause is artificially defined in this approach, specifically

Penalty for deletion: $$d(\mathrm{sil} \mid p_i) = \underset{j}{\mathrm{avg}}\, d(p_j \mid p_i) \tag{9}$$

Penalty for insertion: $$d(p_j \mid \mathrm{sil}) = \underset{i}{\mathrm{avg}}\, d(p_j \mid p_i) \tag{10}$$

2.2.3. Inter-word pronunciation confusability

In this study, the inter-word pronunciation confusability takes a form of inter-word pronunciation distance. In estimating the possibility of a word t_m to be confused in pronunciation by another word t_n, we design a modified version from the well-known Levenshtein distance [10]. The Levenshtein distance is a metric to measure edit distance between two strings. Originally the distance is given by the minimum number of operations needed to transform one string into another, where an operation could be an insertion, deletion, or substitution of a single character. In the modified version, we measure the Levenshtein distance between the pronunciations, i.e., strings of phonemes, of any two words t_m and t_n. Moreover, the insertion, deletion, or substitution of a phoneme p_i is associated with a punishing cost Q. The Levenshtein distance between two pronunciation strings P_{t_m} and P_{t_n} is

$$D(t_n \mid t_m) = \mathrm{LD}(P_{t_m}, P_{t_n}; Q(p_j \mid p_i) : p_i \in P_{t_m}, p_j \in P_{t_n}) \tag{11}$$

where LD stands for Levenshtein distance and can be realized with a bottom-up dynamic programming algorithm [11][12]. Clearly the distance is a function of the pronunciation strings of the two words to be compared as well as a cost Q. The cost can be represented either by the data-driven inter-phoneme distance in Eq. (4) or the phonetic-based inter-phoneme distance Eq. (8), that is,

$$Q(p_j \mid p_i) = d(p_j \mid p_i) \tag{12}$$

We refer to $D(t_n \mid t_m)$ as a tendency or possibility of word t_m recognized as t_n, instead of using the term likelihood, since it is not a probability quantity.

2.2.4. Word pronunciation prominence

The pronunciation prominence (PP), or robustness, of word t_m is defined to characterize the degree of distinction of the word in pronunciation with respect to the rest of the words in the corpus, specifically,

$$R_m = \underset{t_n \in S(t_m)}{\mathrm{avg}} \; D(t_n \mid t_m) - D(t_m \mid t_m) \tag{13}$$

The first term in Eq. (13) measures the average tendency of word t_m to be confused by a group of acoustically most similar words, $S(t_m)$, thus

$$D(t_n \mid t_m) \leq D(t_{n'} \mid t_m), \qquad \forall t_n \in S(t_m), \; \forall t_{n'} \notin S(t_m) \tag{14}$$

In our experiment, the size of $S(t_m)$ is controlled to be top five most confusing words for each t_m. Please note that in the data-driven approach, $D(t_m \mid t_m)$ is not necessarily equal to zero due to the existing speech recognition error. For some poorly recognized words t_m, R_m might be even below zero. In this case we set $R_m = 0$. Depending on the version of IPD used, we have data-driven PP and phonetic-based PP, respectively.

The pronunciation prominence quantity can be enhanced through some nonlinear transformation,

$$\mathrm{PP}_m = \mathscr{T}(R_m) \tag{15}$$

The enhancing function $\mathscr{T}$ can take various forms. In this study, we use power function, thus

$$\mathrm{PP}_m = (R_m)^r \tag{16}$$

The power parameter r is a natural number greater than zero, and used to enhance the pronunciation prominence relative to the existing TF-IDF as shown next. In the experiment we will show that $1 \leq r \leq 5$ will normally suffice our need.

2.3. An indexing weight for voice-to-text search

Combining the pronunciation prominence factor PP_m in the acoustic domain with the TF-IDF in the text domain we obtain a TF-IDF-PP weight

$$(\text{TF-IDF-PP})_{mq} = \text{TF}_{mq} \cdot \text{IDF}_m \cdot \text{PP}_m \quad (17)$$

It embodies the overall prominence of a term t_m to a document d_q in the corpus, evaluated in both the text and acoustic domains. Obviously, for a flat pronunciation prominence factor that is constant across terms, the weight reduces to the normal TF-IDF.

Apart from TF-IDF, the pronunciation prominence factor can also be integrated into other kinds of indexing weights, e.g., TF-DV (term frequency-discrimination value) described in [1].

3. Experiments

3.1. Derivation of confusion matrix

As indicated in Sec. 2.2.1, for the data-driven approach, some speech recognition tests are needed beforehand in order to derive the phonemic confusion matrix. For speech recognition, we use a context-independent acoustic model set containing 3-state HMMs for American English. The features are regular MFCC including 13 cepstral coefficients, 13 first-order cepstral derivative coefficients, and 13 second-order derivative cepstral coefficients. The speech recognition test for determining the confusion matrix is independent of the voice-to-text search test and the only exception is that they use the same speech recognizer and acoustic models. The test data has been manually transcribed beforehand. Speech recognition is run using phoneme-loop grammar on 72,000 isolated words from 1,200 adult, native speakers of American English, including half female and half male speakers. The database contains 6,600 phonetically rich, distinct words.

3.2. Data

We randomly pick 500 pieces of emails from the Enron Email Data Set for the voice-to-text search test. The email headers, nonalphabetical characters as well as punctuations are filtered out. The texts are further screened by a stopword list consisting of 818 words. After the cleaning and filtering process the 500 documents contain total 52,488 words with 8,358 unique words.

3.3. Test metrics

In the voice-to-text test, the distinct words, found in the email data, spoken by 40 speakers are recognized. The recognition is run with a word-loop grammar. The word recognition accuracy $A(t_m)$ is obtained for each word t_m. Therefore, the probability to conduct a successful search of a document d_q can be estimated by

$$A(d_q) = \prod_{t_m \in T(d_q)} A(t_m) \quad (18)$$

Note the multiplication is conducted on a top subset $T(d_q)$ of the word list of the document d_q. The subset $T(d_q)$ is made of the top-ranked words in a sorted word list of document d_q, and its members are just enough for exclusively locating the document. In other words, the number of terms in $T(d_q)$ is equal to the number of steps it takes to find the document d_q if the terms are fed to the search engine in sequence. Average search success rate across all the documents in the corpus can be obtained as

$$A = \operatorname*{avg}_q A(d_q) \quad (19)$$

Namely, the average search success rate measures the rate of successfully reaching a correct target document in a corpus after numerous repeated, speaker-independent, voice-to-text search trials. For each of such trials, just enough search terms are given to the search engine, which would guarantee to reach the target document exclusively if it were purely a text-input search. The average number of search terms needed per document, which measures the average search steps needed when keywords are inputted in sequence, is derived from

$$\operatorname*{avg}_q \left| T(d_q) \right| \quad (20)$$

In terms of the indexing weight used, the word list for each document is sorted in three different ways: TF-IDF only, TF-IDF-PP with PP obtained using data-driven approach, and TF-IDF-PP with PP obtained using phonetic-based approach. There are also some variants to the latter two indexing weights; namely, the importance of the factor PP is adjusted with respect to TF-IDF by changing the value of the power parameter r in Eq. (16). The results are presented in Tables 1 and 2 and discussed in the next section.

3.4. Results

In Table 1 is the search performance with keywords selected using TF-IDF weight and TF-IDF-PP weight where PP is derived with a data-driven IPD.

Table 1. *Result of search tests using keywords selected based on TF-IDF only or TF-IDF-PP where PP (pronunciation prominence) is derived using data-driven approach.*

Indexing weight	r value	Average number of search terms	Average search success rate (%)
TF-IDF	0	2.30	78.29
TF-IDF-PP	1	2.25	80.81
	2	2.25	82.13
	3	2.27	82.88
	4	2.28	83.16
	5	2.29	83.20
	6	2.31	83.20

The results show that the average search success rate improves significantly with TF-IDF-PP weight relative to TF-IDF alone. The benefit increases with the parameter r, i.e., an enhancement of prominence, while it saturates when r is big, e.g., r>5. Generally, by using the new indexing weight an average five percentage point increase in the success rate of search can be obtained.

Note that TF-IDF may not necessarily guarantee the minimal search terms necessary for locating a document since IDF for each term is obtained *globally*. This can be better explained in an equivalent search scenario where the search keywords are fed to the search engine in sequence. The number of necessary search steps in the latter scenario is equal to the number of search terms necessary for pinpointing a document in the former scenario. Clearly, the searches after

the first step are *local* search, and the globally obtained IDF may not render the best search queries. The PP factor, however, tends to bias toward words with large number of syllables such as "microeconomic", "renegotiate", since understandably they are less likely to be acoustically confused. Those words may not have high TF value, which is believed to be somehow related to the main subject of the document, but most of them often turn out to be very unique to the document, and thus help locate the document faster.

To illustrate the influence on the success rate of search caused by the inconsistency in average number of search terms, we made some approximate estimation. By assuming 90% word accuracy, the average word accuracy of our speech recognizer in the experiment, a change of the average number of search terms from 2.30 to 2.25 would have only resulted in an increase from 78.29% to 78.47% in the average search success rate. Therefore, we can say that the improvement in the average search success rate with the TF-IDF-PP weight is largely contributed by the use of acoustically more robust terms as keywords.

By using the pronunciation prominence factor derived from phonetic knowledge, we obtain similar improvement in success rate of search, as shown in Table 2.

Table 2. *The same as Table 1. except that PP is derived using phonetic-based approach.*

Indexing weight	*r* value	Average number of search terms	Average search success rate (%)
TF-IDF	0	2.30	78.29
TF-IDF-PP	1	2.20	80.66
	2	2.14	81.83
	3	2.14	82.09
	4	2.15	82.11
	5	2.16	82.08

The improvement with phonetic-based PP is slightly smaller than the result with data-driven PP shown in Table 1. This is easily understandable. The TF-IDF-PP obtained using the data-driven approach is specific to a certain speech recognizer, particularly the acoustic models used. Therefore, the improvement is more significant when the same speech recognizer and acoustic models are used in the voice-to-text search. The TF-IDF-PP obtained using the phonetic-based approach, however, is general as it is not associated with the performance of any speech recognizer, and furthermore it can be calculated without need for any speech data and speech recognizer. As a benefit, the index weight is portable.

3.5. Discussion

In the new indexing weight, both TF-IDF and PP play indispensable roles in ensuring a high success rate of search. Specifically, PP improves the query selection based on TF-IDF alone. In our experiment, the word accuracy spreads from about 65% all the way up to 99%. Understandably, with the same number of search terms, the final success rate of search can vary greatly depending on the specific terms chosen. The role of TF-IDF is to help constrain the average number of search terms. Even if the individual words selected all have high recognition accuracy, a significant impact would be placed on the success rate of search when the number of search terms is out of control.

4. Conclusions

We developed an indexing weight for voice-to-text search. Compared with the existing TF-IDF that focuses on the text information alone, it provides measurement for selection of queries by taking account of information in both text domain and acoustic domain. This strategy renders a better choice of quires for the voice-to-text search. Our experiment shows five percentage point improvement in the success rate of search with the new weight than the TF-IDF weight.

5. References

[1] Salton, G., Automatic text processing, Addison-Wesley, 1988.

[2] Ramos, J., "Using TF-IDF to determine word relevance in document queries," iCML-03, 2003.

[3] Bahlmann, C. and Burkhardt, H., "Measuring HMM similarity with the Bayes probability of error and its application to online handwriting recognition," ICDAR 01, 406-411, 2001.

[4] Anguita, J. and Hernando, J., "Inter-phone and inter-word distances for confusability prediction in speech recognition," Procesamiento del lenguaje natural, 33: 33-40, 2004.

[5] Tan B. T., Gu, Y., and Thomas, T., "Word confusability measures for vocabulary selection in speech recognition," ASRU 1999, 185-188.

[6] Chomsky, N. and Halle, M., The sound pattern of English, Harper & Row, New York, 1968.

[7] Kessler, B., "Phonetic comparison algorithms," Transactions of the Philological Society, 103, 243-260, 2005.

[8] IPA, Handbook of the International Phonetic Association, Oxford University Press, 1999.

[9] Liu, C. and Melnar, L., "An automated linguistic knowledge-based cross-language transfer method for building acoustic models for a language without native training data," Interspeech-2005, 1365-1368, 2005.

[10] Levenshtein, V. I., "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics Doklady, 10(8): 707–710, 1966.

[11] Bellman, R., Dynamic programming, Dover Publications, 2003.

[12] Wagner R.A. and Fischer M. J., "The string-to-string correction problem," Journal of the ACM, 21(1): 168-173, 1974.