MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Word Particles Applied to Information Retrieval

Evandro Gouvea, Bhiksha Raj

TR2009-018 May 2009

Abstract

Document retrieval systems conventionally use words as the basic unit of representation, a natural choice since words are primary carriers of semantic information. In this paper we propose the use of a different, phonetically defined unit of representation that we call "particles". Particles are phonetic sequences that do not possess meaning. Both documents and queries are converted from their standard word-based form into sequences of particles. Indexing and retrieval is performed with particles. Experiments show that this scheme is capable of achieving retrieval performance that is comparable to that from words when the text in the documents and queries are clean, and can result in significantly improved retrieval when they are noisy.

European Conference on information retrieval

Word Particles Applied to Information Retrieval

Evandro B. Gouvêa and Bhiksha Raj

Mitsubishi Electric Research Labs
201 Broadway, Cambridge, MA 02139, USA

Abstract. Document retrieval systems conventionally use words as the basic unit of representation, a natural choice since words are primary carriers of semantic information. In this paper we propose the use of a different, phonetically defined unit of representation that we call "particles". Particles are phonetic sequences that do not possess meaning. Both documents and queries are converted from their standard word-based form into sequences of particles. Indexing and retrieval is performed with particles. Experiments show that this scheme is capable of achieving retrieval performance that is comparable to that from words when the text in the documents and queries are clean, and can result in significantly improved retrieval when they are noisy.

1 Introduction

Information retrieval systems retrieve documents given a query. Documents are typically sequences of words indexed either directly by the words themselves, or through statistics such as word-count vectors computed from them. Queries, in turn, comprise word sequences that are used to identify relevant documents.

The increasing availability of automatic speech recognition (ASR) systems has permitted the extension of text-based information retrieval systems to systems where either the documents [1] or the queries [2] are spoken. Typically, the audio is automatically or manually transcribed to text, in the form of a sequence or graph of words, and this text is treated as usual.

In all cases, the basic units used by the indexing system are words. Documents are indexed by the words they comprise, and words in the queries are matched to those in the index.

Word-based indexing schemes have a basic restriction, which affects all forms of document retrieval. The key words that distinguish a document from others are often novel words, with unusual spelling. Users who attempt to retrieve these documents will frequently be unsure of the precise spelling of these terms. To counter this, many word based systems use various spelling-correction mechanisms that alert the user to potential misspelling, but even these will not suffice when the user is basically unsure of the spelling. Spoken document/queries pose a similar problem. ASR systems have finite vocabulary that is usually chosen from the most frequent words in the language. Also, ASR systems are statistical machines that are biased *a priori* to recognize *frequent* words more accurately than rare words. On the other hand the key distinguishing terms in any document are, by nature, unusual, and among the least likely to be well-recognized

by an ASR system or to even be in its vocabulary. To deal with this, the spoken audio from the document/query is frequently converted to phoneme sequences rather than to words, which are then matched to words in the query/document.

Another cause for inaccuracies in word-based retrieval is variations in morphological forms between query terms and the corresponding terms in documents. To deal with this, words are often reduced to pseudo-word forms by various forms of stemming [3]. Nevertheless, the remaining pseudo words retain the basic semantic identity of the original word itself for purposes of indexing and retrieval. In other words, in all cases words remain the primary mechanism for indexing and retrieving documents.

In this paper we propose a new indexing scheme that represents documents and queries in terms of an alternate unit that we refer to as *particles* [4] that are not words. Particles are phonetic in nature – they comprise *sequences of phonemes* that together compose the actual or putative pronunciation of documents and queries. Both documents and queries, whether spoken or text, are converted to sequences of particles. Indexing and retrieval is performed using these particle-based representations. Particles however are not semantic units and may represent parts of a word, or even span two or more words. Document indexing and retrieval is thus effectively performed with semantics-agnostic units which need make no sense to a human observer.

Our experiments reveal that this indexing mechanism is surprisingly effective. Retrieval with particle-based representations is at least as effective as retrieval using word-based representations. We note that particle-based representations, being phonetic in nature, may be expected to be more robust than word-based representations to misspelling errors, since misspellings will often be phonetic and misspelt words are pronounced similarly to the correctly spelled ones. Our experiments validate this expectation and more: when the documents or queries are corrupted by errors such as those that may be obtained from misspelling or mistyping, retrieval using particles is consistently more robust than retrieval by words. For spoken-query based systems in particular, particle-based retrieval is consistently significantly superior to word-based retrieval, particularly when the queries are recorded in noise that affects the accuracy of the ASR system.

The rest of the paper is organized as follows. In Sections 2 and 3 we describe particles and the properties that they must have. In Section 4 we describe our procedure to convert documents and queries into particle-based representations. In Section 5 we explain how they are used for indexing and retrieval. In Section 6 we describe our experiments and in Section 7 we present our conclusions.

2 Particles as Lexical Units

Particle-based information retrieval is based on our observation that the language of documents is, by nature, phonetic. Regardless of the origin of words they are basically conceptualized as units of language that must be *pronounced*, *i.e.* as sequences of sounds. This fact is particularly highlighted in spoken-document or spoken-query systems where the terms in the documents or queries are actually spoken.

The pronunciations of words can be described by a sequence of one or more phonemes. Words are merely groupings of these sound units that have been deemed to carry some semantic relationship. However, the sound units in an utterance can be grouped sequentially in any other manner than those specified by words. This is illustrated by Table 1.

Table 1. Representing the word sequence "The Big Dog" as sequences of phonemes in different ways. The pronunciation of the word "The" is */DH IY/*, that for "Big" is */B IH G/*, and for "Dog" it is */D AO G/*.

/DH IY/	/B IH G/	/D AO G/	
/DH IY B/	/IH G D/	/AO G/	
/DH/	/IY B IH/	/G D/	/AO G/

Here we have used the word sequence "The Big Dog" as an example. The pronunciations for the individual words in the sequence are expressed in terms of a standard set of English phonemes. However, there are also other ways of grouping the phonemes in the words together. We refer to these groupings as *particles* and the corresponding representation (*e.g.* */DH IY B/ /IH G D/ /AO G/*) as a *particle based representation*.

This now sets the stage for our formal definition of a particle. We define particles as sequences of phonemes. For example, the phoneme sequences */B AE/* and */NG K/* are both particles. Words can now be expressed in terms of particles. The word "BANK" can be expressed in terms of the two particles in our example as "BANK" → */B AE/ /NG K/*. Particles may be of any length, *i.e.* they may comprise any number of phonemes. Thus */B/*, */B AE/*, */B AE NG/* and */B AE NG K/* are all particles. Particles represent contiguous speech events and cannot include silences. Thus, the particle */NG K AO/* cannot be used in the decomposition of the word BANGKOK, if the user has spoken it as */B/ /AE/ /NG/* <pause> */K/ /AO/ /K/*.

The reader is naturally led to question the choice of phonemes as the units composing particles. One could equally well design them from the characters of the alphabet. We choose phonetic units for multiple reasons:

- As mentioned earlier, words are naturally phonetic in nature. The commonality underlying most morphological or spelling variations or misspellings of any word is the pronunciation of the word. A good grapheme-to-phoneme conversion system [5] can, in fact, map very different spellings for a word to similar pronunciations, providing a degree of insensitivity to orthographic variations.
- In spoken-document and spoken-query systems, recognition errors are often phonetic in nature. Since it is our goal that the particle-based scheme also be effective for these types of IR systems, phonetic representations are far more meaningful than character-based ones.

We note, however, that particles are not syllables. Syllables are prosodically defined units of sound that are defined independently of the problem of representing documents for retrieval. Rather, as we explain in the following sections,

our particles are derived in a *data driven* manner that attempts to emphasize the uniqueness of documents in an index.

3 Requirements for Particle-based Representations

Several issues become apparent from the example in Table 1. a) Any word sequence can be represented as particle sequence in many different ways. Clearly there is great room for inconsistency here. b) The total number of particles, even for the English language that has only about 40 phonemes, is phenomenally large. Even in the simple example of Table 1, which lists only three of all possible particle-based representations of "The Big Dog", 10 particles are used. c) Words can be pronounced in many different ways.

The key to addressing all the issues lies in the manner in which we design our set of *valid* particles, which we will refer to as a *particle set*, and particle-based representations of word sequences.

3.1 Requirement for Particles

Although any sequence of phonemes is a particle, not all particles are *valid*. The particle set that we will allow in our particle-based representations is limited in size and chosen according to the following criteria:

- The length of a particle (in terms of phonemes it comprises) is limited.
- The size of the particle set must be limited.
- The particle set must be *complete*, *i.e.* it must be possible to characterize all key terms in any document to be indexed in terms of the particles.
- Documents must be distinguishable by their particle content. The distribution of particle-based keys for any document must be distinctly different from the distribution for any other document.

The reasons for the conditions are obvious. For effective retrieval particle-based representations are intended to provide keys that generalize to documents pertaining to a given query better than word-based keys, particularly when the text in the documents or queries is noisy. By limiting particle length, we minimize the likelihood of representing word sequences with long particles that span multiple words, but do not generalize. Limiting the size of the particle set also improves generalization – it increases the likelihood that documents pertaining to a query and the query itself will all be converted to particle based representations in a similar manner. Clearly, it is essential that any document or query be convertible to a particle-based representation based on the specified particle set. For instance, a particle set that does not include any particle that ends with the phoneme /*G*/ cannot compose a particle-based representation for "BIG DOG". Completeness is hence an essential requirement. Finally, while the most obvious complete set of particles is one that simply comprises particles composed from individual phonemes, such a particle set is not useful. The distribution of phonemes in any document tends towards the overall distribution of phonemes in the English language, particularly as the size of the document increases and documents cannot be distinguished from one another. It becomes necessary to include larger particles that include phoneme sequences such that the distribution of the occurrence of these particles in documents varies by document.

3.2 Particle-based Representations

As mentioned earlier, there may be multiple ways of obtaining a particle-based representation for any word sequence using any particle set. Consequently, we represent any word sequence by *multiple* particle-based representations of the word sequence. However, not all possible representations are allowed; only a small number that are likely to contain particles that are distinctive to the word sequence (and consequently the document or query) are selected. We select the allowed representations according to the following criteria:

- Longer particles comprising more phonemes are preferred to shorter ones.
- Particle-based representations that employ fewer particles are preferable to those that employ more particles.

Longer particles are more likely to capture salient characteristics of a document. The second requirement reduces the variance in the length of particles in order to minimize the likelihood of non-generalizable decompositions, *e.g.* comprising one long highly-document specific particle and several smaller non-descript ones.

We have thus far laid out general principles employed in selecting particles and particle-based representations. In the following section we describe the algorithm used to actually obtain them.

4 Obtaining Particle Sets and Particle-based Representations

Our algorithm for the selection of particle sets is not independent of the algorithm used to obtain the particle-based representation or *particlization* of text strings – we employ the latter to obtain the former. Below we first describe our particlization algorithm followed by the method used to select particle sets.

4.1 Deriving Particle-Based Representation for a Text String

Our procedure for particlizing word sequences comprises three steps, whereby words are first mapped onto phoneme sequences, a graph of all possible particles that can be discovered in the corresponding phoneme sequence is constructed, and the graph is searched for the N best particle sequences that best conform to the criteria of Section 3.1. We detail each of these steps below.

Mapping Word Sequences to Phoneme Sequences

We replace each word by the sequence of phonemes that comprises its pronunciation, as shown in Table 2. The pronunciation of any word is obtained from a pronunciation dictionary. Text normalization may be performed [6] as a preliminary step. If the word is not present in the dictionary even after text normalization, we obtain its pronunciation from a grapheme-to-phoneme converter (more commonly known as a pronunciation guesser). Most speech synthesizers, commercial or open source, have one.

If the word has more than one pronunciation, we simply use the first one. The strictly correct solution would be to build a word graph where each pronunciation is represented by a different path, and then mapping this graph to a

particle sequence; however, if mapping of words to phoneme sequences is consistently performed, multiplicity of pronunciation introduces few errors even if the text is obtained from an speech recognizer.

Table 2. *Mapping the word sequence "SHE HAD" to a sequence of phonemes. "SHE" is pronounced as "/SH/ /IY/" and "HAD" is pronounced as "/HH/ /AE/ /D/".*

SHE HAD	→	/SH/ /IY/ /H/ /AE/ /D/

Composing a Particle Graph

Particles from any given particle set may be discovered in the sequence of phonemes obtained from a word sequence. For example, Table 3 shows the complete set of particles that one can discover in the pronunciation of the word sequence "SHE HAD" from a particle set that comprises every sequence of phonemes up to five phonemes long.

Table 3. *Particles constructed from the phone sequence /SH/ /IY/ /HH/ /AE/ /D/ obtained from the utterance "she had".*

Particle set				
/SH/	/SH IY/	/SH IY HH/	/SH IY HH AE/	/SH IY HH AE D/
/IY/	/IY HH/	/IY HH AE/	/IY HH AE D/	
/HH/	/HH AE/	/HH AE D/		
/AE/	/AE D/			
/D/				

The discovered particles can be connected to compose the complete pronunciation for the word sequence in many ways. While the complete set of such compositions can be very large, they can be compactly represented as a graph, as illustrated in Figure 1. The nodes in this graph contain the particles. An edge links two nodes if the last phoneme in the particle at the source node immediately precedes the first phoneme in the particle at the destination node. The entire graph can be formed by the simple recursion of Table 4. Note that in the final graph nodes represent particles and edges indicate which particles can validly follow one another.

Searching the Graph

Any path from the start node to the end node of the graph represents a valid particlization of the word sequence. The graph thus represents the complete set of all possible particlizations of the word sequence. We derive a restricted subset of these paths as valid particlizations using a simple graph-search algorithm.

We assign a score to each node and edge in the graph. Node scores are intended to encourage the preference of longer particles over shorter ones. We enforce particularly low scores for particles representing singleton phonemes, in order to strongly discourage their use in any particlization. The score for a node

Table 4. *Algorithm for composing particle graph.*

Given:	
Particle set	$\mathcal{P} = \{R\}$ composed of particles of the form $R = /p_0\ p_1\ \cdots\ p_k/$, where p_0, p_1 etc. are phonemes.
Phoneme sequence	$\mathbf{P} = P_0\ P_1\ \cdots\ P_N$ derived from the word sequence
CreateGraph(*startnode***, j, P,** *finalnode***):**	
For each	$R = /p_0\ p_1\ \cdots\ p_k/ \in \mathcal{P}$ s.t. $p_0 = P_j, p_1 = P_{j+1}, \cdots, p_k = P_{j+k}$:
i.	Link *startnode* $\rightarrow R$
ii.	If $j + k == N$:
	Link $R \rightarrow$ *finalnode*
	Else:
	CreateGraph(R, $j + k + 1$, **P**, *finalnode*)
Algorithm:	
	CreateGraph(<s>, 0, **P**, </s>)

n representing any particle P is given by

$$Score(\mathbf{n}) = \begin{array}{ll} \alpha & \text{if } length(Particle(\mathbf{n})) == 1 \\ \beta / length(Particle(\mathbf{n})) & \text{otherwise} \end{array} \quad (1)$$

where $length(Particle(N))$ represents the length in phonemes of the particle represented by node N. Node scores are thus derived solely from the particles they represent and do not depend on the actual underlying word sequence. In our implementations α and β were chosen to be -50 and -10 respectively.

Edge scores, however, do depend on the underlying word sequence. Although particles are allowed to span word boundaries, we distinguish between within word structures and cross-word structures. This is enforced by associating a different edge cost for edges between particles that occur on either side of a word boundary than for edges that represent particle transitions with a word. The score for any edge **e** in the graph is thus given by

$$Score(\mathbf{e}) = \begin{array}{ll} \gamma & \text{if } word(particle(source(\mathbf{e}))) == word(particle(destination(\mathbf{e}))) \\ \delta & \text{otherwise} \end{array} \quad (2)$$

where $word(particle(source(\mathbf{e})))$ is the word within which the trailing phoneme of the particle at the source node for **e** occurs, and $word(particle(destination(\mathbf{e}))$ is the word within which the leading phoneme of the particle at the destination node for **e** occurs. We have found it advantageous to prefer cross-word transitions to within-word transitions and therefore choose $\beta = -10$ and $\gamma = 0$.

Having thus specified node and edge scores, we identify the N *best* paths through the graph using an A-star algorithm [7]. Table 5 shows an example of the 3-best particlizations obtained for the word sequence "SHE HAD".

Table 5. *Example particlizations "SHE HAD".*

/SH IY HH AE D/	
/SH IY/	*/HH AE D/*
/SH IY HH/	*/AE D/*

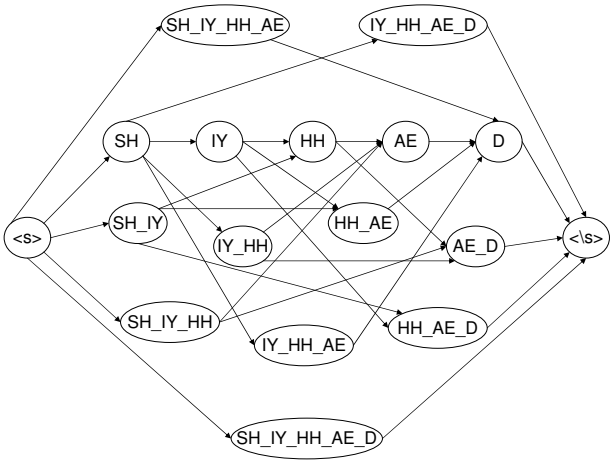

Fig. 1. Search path displaying all possible particlizations of the utterance "she had" (/SH IY/ /HH AE D/) with particles of length up to 5 phonemes.

4.2 Deriving Particle Sets

We are now set to define the procedure used to obtain particle sets. Since our final goal is document retrieval, we obtain them by analysis of a training set of documents. We begin by creating an initial particle set that comprises all phoneme sequences up to five phonemes long. We then use this particle set to obtain the 3-best particlizations of all the word sequences in the documents in the training set. The complete set of particles used in the 3-best particlizations of the document set are chosen for our final particle set. In practice, one may also limit the size of the particle set by choosing only the most frequently occurring particles. To ensure completeness we also add to them all singleton-phoneme particles that are not already in the set in order to ensure that all queries and documents not already in the training set can be particlized.

The above procedure generally delivers a particle set that is representative of the training document set. If the training data are sufficiently large and diverse, the resultant particle set may be expected to generalize across domains; if, however, the training set comprises documents from a restricted set of domains, the obtained particle set is domain specific. It is valid to obtain particles directly from the actual document set to be indexed. However, if this set is small, addition of new documents may require extension of the particle set to accommodate them, or may result in sub-optimal particlization of the new documents.

Finally, the algorithm of Section 4.1 does not explicitly consider the inherent frequency of occurrence of particles in the training data (or their expected frequency in the documents to be indexed). In general, particle-occurrence statistics could be derived from a statistical model such as an N-gram model detailing co-occurrence probabilities of particles, and impose these as edge scores in the graph. Particle set determination could itself then be characterized as an iterative

maximum-likelihood learning process that alternately obtains N-best particlizations of the documents and co-occurrence probabilities from these particlizations; however we have not attempted this in this paper.

5 Document Retrieval using Particles

Figure 2 depicts the overall procedure for document retrieval using particles. All documents are converted to particle-based representations prior to indexing. To do so, the 3-best particlizations of each sentence in the documents are obtained. This effectively triples the size of the document.

Queries are also particlized. Once again, we obtain the 3-best particlization of the query and use all three forms as alternate queries (effectively imposing an "OR" relation between them).

When queries are *spoken* we employ an ASR system to convert them to text strings. More explicitly, we extract the K-best word sequence hypotheses from the recognizer. In our implementation, K was also set to 3. Each of the K-best outputs of the recognizer is particlized, resulting a total of $3K$ alternate particlizations of the query, that are jointly used as queries to the index.

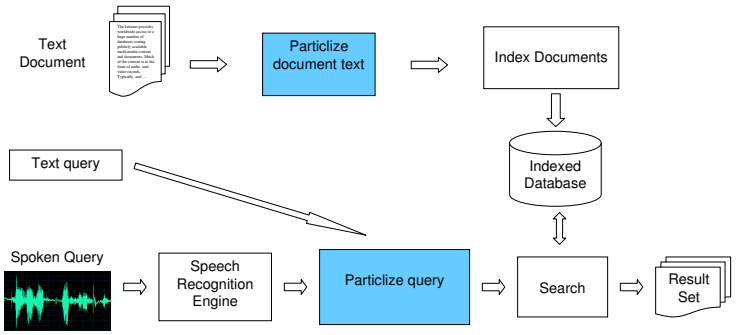

Fig. 2. Particle-based retrieval

6 Experiments

In this section, we compare document retrieval between a word-based system and a particle-based one. For each of these, we present results on textual and spoken query. The document retrieval engine used was our SpokenQuery (SQ) [2] system that can work from both text and spoken queries. Spoken queries are converted to text (N-best lists) using a popular high-end commercial recognizer.

We created indices from textual documents obtained from a commercial database that provides information about points of interest (POI), such as business name, address, category (e.g. "restaurant"), sub-category, if applicable (e.g., "french"). To evaluate performance as a function of index size we created 5 different indices containing 1600, 5500, 10000, 22000 and 72000 documents.

In the word-based system evaluation, the query is presented unmodified to the SQ system. In the particle-based evaluation, the query is transformed to a particle-based list by the algorithm presented in Section 4, and presented to SQ.

We used the *limited* or *bounded recall rate* as metric of quality of retrieval. The *recall rate* is commonly used to measure sensitivity of information retrieval systems. It is defined as the number of true positives normalized by the sum of true positives and false negatives. But this definition unfairly penalizes cases where the number of true positives is higher than the number of documents retrieved. The bounded recall normalizes the number of correct documents found by the minimum between the number of correct documents and the number of documents retrieved.

The test set consists of an audio database collected internally. This database, named *barePOI*, consists of about 30 speakers uttering a total of around 2800 queries. The queries, read by the speakers, consist of POI in the Boston area. We used the transcriptions only in the text queries experiments in Section 6.1 and the audio in the spoken queries experiment in Section 6.2.

6.1 SpokenQuery Performance using Word- and Particle-Based Text Queries

The text queries were generated from the transcriptions from the barePOI database. To simulate misspellings, we simulated errors in the queries. The queries could be word-based or particle-based. We use the more general label "term", which refers to "word" in the case of word-based retrieval and to "particle" in the case of particle-based retrieval. We randomly changed terms in the queries in a controlled manner, so that the overall rate of change would go from 0%, the ideal case, to 40%. Figure 3 presents the results for both word-based and particle-based experiments. The solid lines represent results using particle-based queries, whereas dashed lines represent word-based results. Lines with the same color represent the same numerical term error rate.

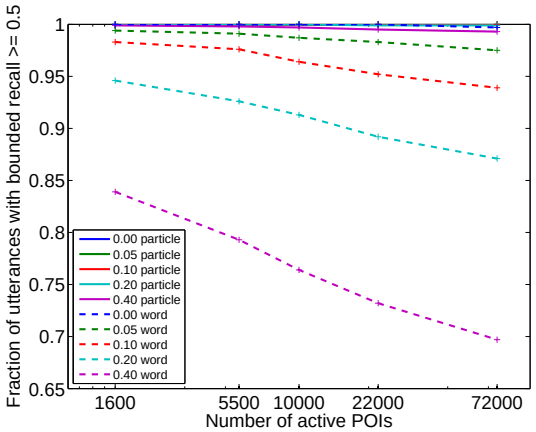

Fig. 3. Bounded recall for BarePOI test set with word- and particle-based retrieval from text queries, at several term error rates.

Note that we do not claim that the particle and word error rates are equivalent, or that there is a simple mapping from one to the other. Consider, for example, the case where a word has been replaced in the query. When we map

this query into a sequence of particles, one word error, a substituted word, will map to a sequence of particles that may have an error count ranging from zero up to the number of phones in the word. Therefore, the number of particle errors is not predictable from the number of word errors.

Figure 3 confirms that particle-based retrieval works spectacularly well as compared to word based retrieval. A system using particle-based text retrieval would benefit since there is no need for text normalization, provided that a reasonable pronunciation guesser is available to convert a text string to a particle sequence.

6.2 SpokenQuery Performance using Word- and Particle-Based Spoken Queries

The spoken queries were the audio portion of the barePOI database. We artificially added car engine noise at different levels of signal to noise ratio (SNR) to simulate real conditions.

The Word Error Rate (WER) for each of the test conditions is presented in Figure 4. Note that, as expected, the WER increases when the number of POI increases, since the higher vocabulary size and larger language model increase confusability. As expected, the WER also increases when the noise level increases.

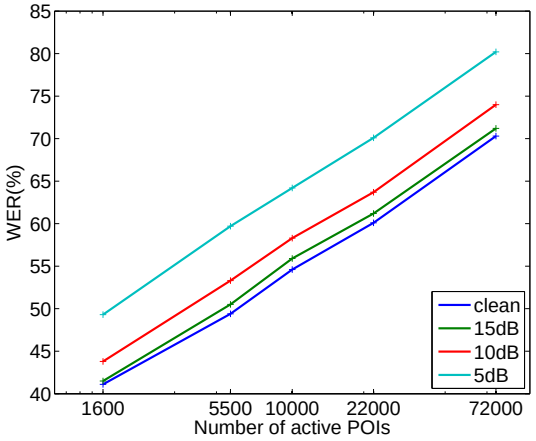

Fig. 4. Word error rate at several noise conditions.

Figure 5 presents the bounded recall for word-based and particle-based retrieval using a commercial speech recognizer's recognition results. The different colors represent speech at different SNR levels. We note the smooth degradation as the POI size increases. Since word error rate tends to increase with increasing SNR, it is clear that particle-based SpokenQuery shows much better robustness to error rate over the range of active POI used in the experiment (1600 to 72000). Particle-based SpokenQuery shows much better performance than word-based SpokenQuery in all conditions.

7 Conclusion

In this paper we have proposed an alternative to meaningful-word-based representations of text in documents and queries using phonetically described particles

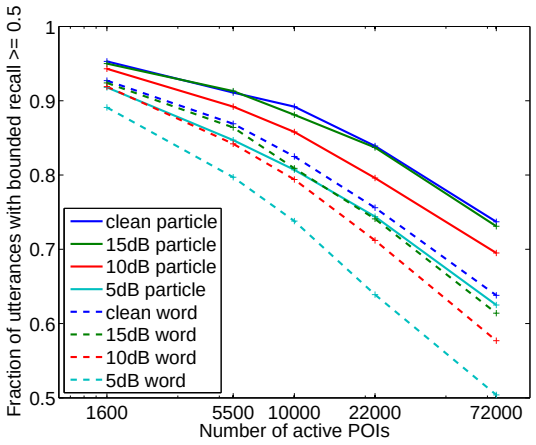

Fig. 5. Bounded recall for BarePOI test set with word- and particle-based retrieval from a commercial recognizer's output.

that carry no semantic weight. Performance in this new domain is shown to be superior to that obtained with word-based representations when the text is corrupted. We have shown that improvement in performance is obtained both when the documents and queries are purely text-based, and when queries are actually spoken and converted to text by a speech recognition system.

The results in this paper, while showing great promise, are yet preliminary. Our particle sets were domain specific. We have not attempted larger scale tests and are not aware of how the scheme works in more diverse domains or for larger document indices. We also believe that performance can be improved by optimizing particle sets to explicitly discriminate between documents or document categories. On the speech recognition end, it is not yet clear whether it is necessary to first obtain word-based hypotheses from the recognizer or better or comparable performance could be obtained if the recognizer recognized particles. Our future work will address all of these and many other related issues.

References

1. Thong, J.M.V., Moreno, P.J., Logan, B., Fidler, B., Maffey, K., Moores, M.: Speechbot: an experimental speech-based search engine for multimedia content on the web. IEEE Trans. Multimedia **4** (2002) 88–96
2. Wolf, P.P., Raj, B.: The MERL SpokenQuery information retrieval system: A system for retrieving pertinent documents from a spoken query. In: Proc. ICME. (2002)
3. Ogilvie, P., Callan, J.: Experiments using the lemur toolkit. In: Proc. TREC. (2001)
4. Whittaker, E.W.D.: Statistical language modelling for automatic speech recognition of Russian and English. PhD thesis, Cambridge University (September 2000)
5. Daelemans, W., Bosch, A.V.D.: Language-independent data-oriented grapheme-tophoneme conversion. In: Progress in Speech Processing, Springer-Verlag (1996)
6. Mikheev, A.: Document centered approach to text normalization. In: Proc SIGIR, ACM (2000) 136–143
7. Daniel Jurafsky, J.H.M.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall (2000)