# Error Analysis of Automatic Speech Recognition Using Principal Direction Divisive Partitioning*

David McKoskey and Daniel Boley

Department of Computer Science and Engineering
University of Minnesota, Minneapolis, 55455, USA
mckoskey@cs.umn.edu

**Abstract.** This paper describes an experiment performed using the Principal Direction Divisive Partitioning algorithm (Boley, 1998) in order to extract linguistic word error regularities from several sets of medical dictation data. For each of six physicians, two hundred finished medical dictations aligned with their corresponding automatic speech recognition output were clustered and the results analyzed for linguistic regularities between and within clusters. Sparsity measures indicated a good fit between the algorithm and the input data. Linguistic analysis of the output clusters showed evidence of systematic word recognition error for short words, function words, words with destressed vowels, and phonological confusion errors due to telephony (recording) bandwidth interference. No qualitatively significant distinctions between clusters could be made by examining word errors alone, but the results confirmed several informally held hypotheses and suggested several avenues of further investigation, such as the examination of word error contexts.

## 1 Introduction

Industrial grade speech recognition has made numerous advances in recent years, especially in corpus based implementations. Modern recognition software such as the application used for this study, often employs a sophisticated combination of techniques for matching speech utterances with their most likely or most desirable text representation (e.g. Hidden Markov modes, rule-based post-recognition processors, partial parsers, etc.). Under ideal conditions, these models enjoy a combined recognition accuracy that approaches 100%. However, word errors due to the misrecognition of an utterance are still not very well understood. Many simple factors influence word recognition accuracy, such as model parameters (e.g. language model scaling factors, word insertion penalties, etc.), speech fluency or disfluency, and items missing from the recognition model's vocabulary. Other factors are more complex, such as the influence of vocal prosody, or vowel devoicing.

Tuning these recognition tools requires extensive analysis, experimentation, and testing. One useful technique for analyzing word errors is linguistic analysis,

* This work was partially supported by NSF grant IIS-9811229.

R. López de Mántaras, E. Plaza (Eds.): ECML 2000, LNAI 1810, pp. 263–270, 2000.

in which one inspects the available data in search of word error exemplars that adequately represent the more general case. Filled pauses ("um" or "ah"), for example, have been successfully modeled using this technique, and have been shown to "follow a systematic distribution and well defined functions" [4]. As a result recognition accuracy for medical dictation is enhanced by representing the frequency of filled pauses in the recognition model's training data [5]. Unfortunately, other word errors, such as words mistakenly recognized as filled pauses (e.g. "um" may be mistakenly recognized as "thumb" or "arm") [4] are much more difficult to analyze because of their sparsity. In cases where word errors are sparse, error detection by inspection, while still the most accurate of any technique, becomes much more arduous and/or costly.

As part of the Web ACE Project [1], the Principle Direction Divisive Partitioning (PDDP) algorithm was originally designed to classify large collections of documents gleaned from the World Wide Web by clustering them on word frequency. Each document is encoded as a column vector of word counts for all words in the document set, and the document vectors combined into a single matrix. The clustering process recursively splits the matrix and organizes the resulting clusters into a binary tree.

The clustering process consists of four steps:

1. Assign the input matrix as the initial cluster and root of the output PDDP tree. For the initial iteration, the root node is also the only leaf node.
2. Calculate the *scatter value* for all leaf nodes in the PDDP tree and select the node with the largest scatter value.
3. For each document $\mathbf{d}$ in the selcted cluster $\mathtt{C}$ containing $\mathbf{k}$ documents, assign $\mathbf{d}$ to the left or right child of $\mathtt{C}$ according to the sign of the linear discrimination function $g_{\mathtt{C}}(\mathbf{d}) = \mathbf{u}_{\mathtt{C}}^T(\mathbf{d} - \mathbf{w}_{\mathtt{C}}) = \sum_{i=1}^{n} u_i(d_i - w_i)$ where $\mathbf{w}_{\mathtt{C}}$ is the centroid of the current cluster and $\mathbf{u}_{\mathtt{C}}$ is the direction of maximal variance, or principle direction of $\mathtt{C}$. If $g_{\mathtt{d}} \leq 0$, then place $\mathbf{d}$ into the new left child node of $\mathtt{C}$, otherwise, place $\mathbf{d}$ into the new right child of $\mathtt{C}$.
4. Repeat from step 2

The vector $\mathbf{w}_{\mathtt{C}} \stackrel{\text{def}}{=} \frac{1}{k}\sum_j \mathbf{d}_j$ is the mean or centroid of node $\mathtt{C}$. The *scatter value* used for this study is simply the sum of all squared distances from each document $\mathbf{d}$ to the cluster centroid $\mathbf{w}$, though any other suitable criterion may be used as well. The principle direction $\mathbf{u}_{\mathtt{C}}$ corresponds to the largest eigenvalue of the sample covariance matrix for the cluster $\mathtt{C}$. This calculation is the costliest portion of the algorithm, but can be performed quickly with a Lanczos-based singular value solver. The splitting process repeats until either the maximum scatter value of any leaf node is less than the scatter of all current leaf node centroids (a stop test), or until a desired total number of leaf nodes has been reached [2].

Two strengths of the PDDP algorithm include its competitiveness with respect to cluster quality and run time. Previous analysis indicates that PDDP run time scales linearly with respect to the density of the input data matrix, not its size [2]. Studies comparing entropy measures between PDDP and other

clustering methods (such as Hypergraph or LSI) indicate that PDDP exhibits competitive performance on cluster entropy ("cluster quality") measures [2]. For these reasons, the PDDP algorithm was seleted to cluster several sets of medical dictation data, clustering on the frequency of word errors in each dictation document. We had no solid hypotheses about what sort(s) of results the clustering would reveal, but hoped the cluster trees would:

(a) reveal any linguistic regularities in the word errors of each cluster, and
(b) indicate any relationships between specific word errors and the physician or physicians that most often make(s) them

Results from the mining process would be used to further refine the acoustic and/or language models required by the recognition software (used for this study and elsewhere), and to provide new parsing rules for error correction during post-recognition processing.

## 2 Data Characteristics and Processing

Modern medical practice typically includes document dictation for the sake of expediency. For example, a doctor dictates his or her patient chart notes into a recording device, and the audio is replayed for a medical transcriptionist (a proficient typist with extensive medical training). The transcriptionist types the dictation, formats the text as chart notes, and submits them to the dictating doctor for inspection. Once the notes are inspected, proofread, and approved, they are inserted into the patient's medical record. Below is an excerpt from a sample finished transcription:

⟨ date ⟩⟨ name ⟩
Subjective: patient is a 51-year-old woman here for evaluation of complaints of sore throat and left ear popping.
Objective: The patient is alert and cooperative and in no acute distress. External ears and nose are normal.
Assessment: Upper respiratory tract infection.
Plan: treat symptomatically with plenty of fluids, a vaporizer and analgesics as needed.

Linguistic Technologies Inc. (LTI), a medical transcription company based in St. Peter, MN USA, performs recognition on medical dictation audio using an automated speech recognition application. This application is comprised of a Hidden Markov Model decoder, acoustic model, language model, and language dictionary. A rule-based post processor is also used after recognition is complete, to perform several simple parsing tasks, such as formatting numbers (e.g. "one hundred forty over eighty" becomes "140/80"). The output text is then corrected and formatted by a medical transcriptionist for final approval by the dictating

physician. The recognition output, if sufficiently accurate, significantly reduces the medical transcriptionist's workload.

For each of six physicians (henceforth talkers), two hundred finished medical dictations and their corresponding recognition output files were selected. Each set of files was sanitized to remove demographic and time stamp data. Recognition output was conditioned in order to normalize the text (downcase all words, convert numbers and punctuation to text, use standard representations for contractions and abbreviations, etc.). Normalization also included substituting tokens (called TT-words) for common words or phrases (e.g. "TT_nad" is substituted for "no acute distress"), words that require capitalization (e.g. proper names) or words that predictably required specific punctuation marks (e.g. "TT_yearold" was substituted for "year-old"). Finished dictations were treated using PLAB, a proprietary algorithm developed at LTI for inferring transcription of actual speech from formal transcription [5]. This process also included text normalization and rendered the finished dictation into a form that conformed accurately to what was actually said in the original dictation audio. For example, the above finished dictation, after sanitizing and PLAB processing, would look like this:

```
   <s> dictating on paragraph TT_scolon patient is a
fifty one TT_yearold woman here for evaluation of ah
complaints of a sore throat and left ear popping period
the TT_patient alert cooperative and in TT_nad period
external ears and nose are normal period TT_acolon upper
respiratory tract infection period paragraph plan colon
will treat this symptomatically with plenty of fluids
ah ah vaporizer and analgesics as needed period </s>
```

The PLAB output and normalized/sanitized recognition output were then aligned word by word. Alignment errors were then divided into three categories:

1. Insertions: words the recognizer inserted that were not in the final dictation (e.g. the software recognized a cough, throat-clearing, or other such utterance as a word).
2. Deletions: words the recognizer deleted by mistake, the reverse scenario of an insertion error.
3. Substitutions: words the recognizer confused (e.g. "he" and "she" are easily confused).

Here is a sample excerpt from an alignment file, illustrating the three types of errors:

```
TT_ocolon          TT_ocolon
the                --                 DELETION
TT_patient         TT_patient
is                 ---                DELETION
```

```
alert              alert
and                ---              DELETION
cooperative        cooperative
and                and
in                 in
---                no               INSERTION
---                acute            INSERTION
---                distress         INSERTION
TT_nad             since            SUBSTITUTION
external           external
ears               ears
...                ...
```

A matrix containing counts of each word error by document was created for each error category (insertion, deletion, and substitution). Each matrix was then clustered using the PDDP algorithm, which separated the documents in each matrix into clusters by word error. Euclidian Norm scaling was used [1], and the algorithm was halted after fifty clusters were obtained. Histograms were created for each cluster, indicating the number of documents for each talker in that cluster. The cluster's ten most common word errors were also reported, as indicated by the cluster centroid's ten highest values. If the cluster was split, then the ten highest and ten lowest principle direction word errors were also reported, indicating the word errors with the greatest contribution to the split.

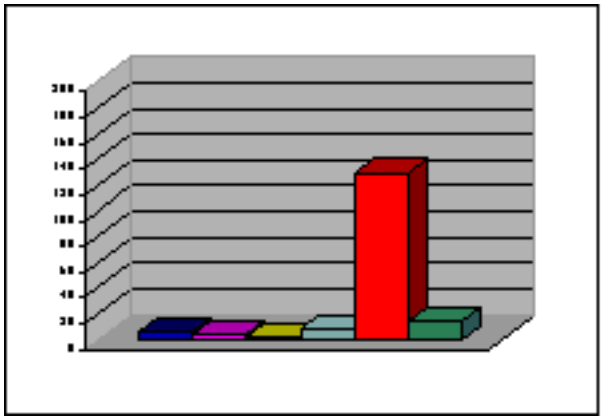
## 3 Results

Sparsity measures for each matrix were taken by simply dividing the number of entries greater than zero by the total size of the input matrix. These measures indicated that all input matrices were between 0.15% and 0.72% fill. This is very sparse, which showed that the data and algorithm were a good match. Some clusters showed a high frequency of a single talker's documents, but significantly fewer documents from other talkers, as illustrated in Fig. 1). These clusters showed a relationship between the strongly represented talker and the word errors of that document's centroid. (In Figures 1 and 2, each colored column represents a different talker. The y-axis on the left edge of the graph contains a scale of 0 to 200, the maximum number of documents for any talker.)

Other cluster histograms showed a more equal representation among talkers, indicating that word errors reported by the centroid (and in the centroids of other, similar clusters) were of a more global character, as indicated in Fig 2.

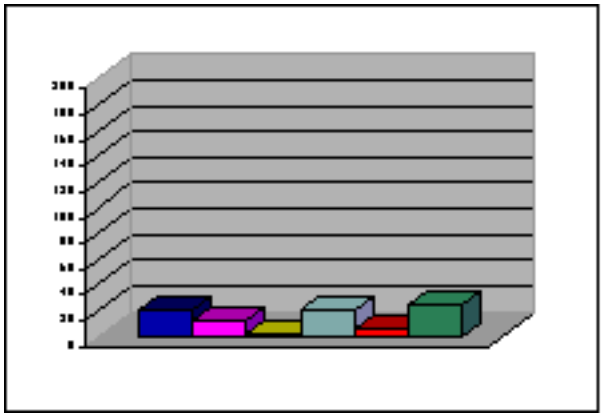
### 3.1 General Characteristics

Most of the words reported at each cluster and at each split were short words and function words. "Short words" are words that contain only one or two syllables, such as "he" or "she" ("longer words" will refer to words of three or more



| | |
|---|---|
| and | 0.27611 |
| nine | 0.19283 |
| ah | 0.17519 |
| is | 0.17195 |
| ninety | 0.16003 |
| the | 0.11742 |
| to | 0.11731 |
| are | 0.09040 |
| oh | 0.08868 |
| has | 0.08609 |
| twenty | 0.07591 |

**Fig. 1.** This cluster and centroid words indicate a strong relationship between a particular talker and particular word errors



| | |
|---|---|
| the | 0.37699 |
| to | 0.16542 |
| of | 0.11077 |
| ah | 0.09663 |
| has | 0.08082 |
| hundred | 0.06940 |
| in | 0.06397 |
| one | 0.06218 |
| is | 0.04893 |
| and | 0.04880 |
| her | 0.04545 |

**Fig. 2.** This cluster suggests that the centroid values are likely global (more ubiquitous) errors

syllables). Function words have little semantic content, but have grammatical function instead, such as determiners ("a", "an", "the"), conjunctions ("and", "or", "but"), copulas ("is", "was", "were") and quantifiers (e.g. numbers). There was a great amount of overlap between these two categories, as most function words are short and many short words are function words.

### 3.2 Vowel Destressing and Cliticization

Notably, most short word and function word errors contained a destressed vowel. Vowel destressing often co-occurs with cliticization, in which the short word is

"attached" to one of its longer neighbor words. For example, the word "and" in the above excerpt is destressed and cliticized in the phrase "ears and nose": the "a" is destressed and deleted, and the "d" is deleted. The result is an utterance that sounds like "ears anose" or "earsanose" unless spoken very carefully. The recognition software treated most words of this type as noise or filled pauses and discarded them. Vowel destressing and cliticization for short words and function words was common throughout most of the medical dictation examined. While still an unconfirmed hypothesis, we suspect that many destressed words were located near the ends of phrases, a point at which a talker's speech is likely to accelerate.

### 3.3 Vowel Syncope

Other clusters showed evidence of vowel syncope, in which unstressed vowel sounds in quickly spoken words are deleted. For example, a talker might signal to the medical transcriptionist the end of one paragraph and the beginning of another simply by saying "paragraph". Even in relatively unhurried speech, though, this word was often said quickly, and in the process, the second and "a" in "paragraph" was deleted. The result was an utterance that sounded like "pair-graph". Said even more quickly, the third "a" was also deleted: "pair-grph". As a result, recognition software misidentified the word containing the syncopated vowel(s), making a substitution error (e.g."oh" for "zero"), or treated the utterance as noise or a filled pause and discarded it, making a deletion or insertion error. Syncope was also ubiquitous throughout the medical dictations examined, though not as common as short word errors.

### 3.4 Telephony Interference

Cluster centroids and splits also showed some evidence of telephony bandwidth interference. Words (especially short words) that contained voiceless fricative consonants ("f", "th", "s", "sh", etc.), were easily confused, especially in cases where the fricative carries the greatest amount of word information (e.g. "he" versus "she"). These words were also easily mistaken as noise or filled pauses, though short words more frequently than longer words (words of three or more syllables).

## 4 Discussion / Future Work

Several conclusions can be drawn from the above results. Firstly, word errors involving short, destressed words and function words are ubiquitous throughout the medical dictations examined with the PDDP algorithm. Most often, these words were confused with other function words, brief periods of silence, background noise, or filled pauses. We hypothesized prior to the study that this was the case, but until now, had no way to easily visualize it. One task for subsequent studies would be to cluster the PDDP tree using a centroid stopping

test (described earlier), and re-agglomerate several of the leaf clusters, without regard to which side of the PDDP tree the leaves are situated. This way, clusters that were accidentally fragmented on one dimension during a split along another dimension could be reassembled.

Secondly, number words may or may not cause recognition accuracy problems, because it is known that the first twenty or so words of any dictation contain the patient name, current date, and the name of the dictating physician. These excerpts are rarely, if ever, recognized accurately. Instead, post-recognition processing (simple parsing) seems to more easily rectify problems organizing and correcting word errors involving number words. Future work will more carefully exclude the initial portion of the dictation alignment, so that clustering results will concern only number words found in the body of the dictation text.

Finally, we also noticed that several talkers were split off into their own clusters, such as the cluster shown in Fig. 1. Most often, one or two high frequency word errors were responsible for separating out a specific talker, but more generally, we were unable to discern any qualitatively significant word features that distinguished words in these clusters from word errors elsewhere in the tree. For example, a high frequency of deletion errors involving the word "and" separated out one talker, but all by itself, the word "and" isn't significantly different from the word "an", especially in telephone speech. The distinguishing factor(s), then, must reside not only in the distinguishing words themselves, but in the context in which those words were situated. One important next step for this study will be to examine context effects surrounding word errors, including word collocation and syntactic part of speech.

## References

1. Boley, D.: Principal direction Divisive Partitioning. Data Mining and Knowledge Discovery. **2:4** (1998) 325-344 264, 267
2. Boley, D., Borst, V.: Unsupervised Clustering: A Fast Scalable Method For Large Datasets. U of MN CSE Report TR-99-029 1998 264, 265
3. Pakhomov, S.: Modeling Filled Pauses in Medical Dictations. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. (1999)
4. Pakhomov, S., Savova, G. 1999 "Filled Pause Distribution and Modeling in Quasi-Spontaneous Speech. In Proceedings of the 14th International Congress of Phonetic Sciences (ICPS). (1999) 264
5. Sullivan - Pakhomov, S., Schonwetter, M.: US Patent Application For A Method and System for Generating Semi-Literal Transcripts for Speech Rercognition Systems. (2000) 264, 266