
Proceedings Paper:
Churcher, G, Atwell, ES and Souter, DC (1996) Dialogues in air traffic control. In: Proceedings Twente Workshop on Language Technology 11 (TWLT 11) Dialogue Management in Natural Language Systems. The 11th Twente Workshop on Language Technology, 19-21 Jun 1996, University of Twente, Enschede, The Netherlands. University of Twente, Enschede, The Netherlands , 101 - 112.

DIALOGUES IN AIR TRAFFIC CONTROL

Gavin E Churcher, Clive Souter & Eric S Atwell
School of Computer Studies,
Leeds Univeristy, Leeds LS2 9JT
UK
Tel: +44 113 2336827
(Email: gavin@scs.leeds.ac.uk)

ABSTRACT

We have taken an off-the-shelf, commercial continuous speech recogniser and conducted evaluations for the domain of Air Traffic Control. The language of this domain proved to be quite unrestricted, contrary to our initial intuitions. Our experiments show that constraints typically used by speech recognisers do not provide accurate enough results and need to be augmented with other knowledge sources and higher levels of linguistics in order to prove useful.

We used three syntaxes based on a corpus of transmissions between the ATC and pilots in order to reflect differing levels of "linguistic" knowledge. Initial experiments demonstrate the benefit of a fully constrained context-free semantic grammar. Further experiments empirically show the benefit to recognition accuracy of using some form of dialogue management system to control the flow of discourse. A corpus-based statistical clustering approach to the segmentation of a dialogue into discourse segments is briefly discussed.

INTRODUCTION

We started a project which intended to use speech recognition technology to automatically transcribe certain, essential parts of transmissions between Air Traffic Control (ATC) and airborne pilots. This information could either be used for ATC training purposes, or for relaying this information back to the pilot in order to reduce the burden of flying. Rather than tackle all important information in the transmission, we concentrated on five areas:

1. Instructions to the pilot to change his/her altitude. Information would be an altitude either in terms of a height in feet or a flight level.

2. Pressure settings for QFE (observed pressure) and QNH (altimeter/sub-scale setting). Pressure settings are measured in millibars.

3. Secondary Surveillance Radar (SSR) settings for squawk values. Squawk values are transponder settings which enable ATC to identify aircraft via radar.

4. Instructions to the pilot to change to another radio frequency.

5. Instructions to the pilot to change his/her heading, a setting measured in magnetic degrees.

Appendix 1 contains some example transmissions by the ATC; important information is highlighted.

The domain was initially thought to be complex, but practical, requiring continuous, speaker independent speech recognition with real-time response. In order to start building a model of ATC utterances, the Radiotelephony Manual [RTF CAP413] was examined. The manual provided protocols and examples for a number of situations such as landing, taking off, changing frequency etc. To have a better idea of the actual language used behind the protocols, a corpus of transmissions was collected.

It was this corpus (see The LBA Corpus below) which led us to believe that the ATC domain used choice phrases for each of the above areas which

could deviate slightly in many different ways. For example, instructing the pilot to change his radio frequency can start with phrases such as: "contact the tower now", "proceed to contact the tower on...", "you are free to call the tower..." etc. These key phrases were also interspersed and surrounded by other 'noise-phrases' representing other information and apparently free English language.

We required a speech recogniser which could transcribe continuous speech for a medium sized sub-language which was highly structured, and yet fairly flexible.

THE SPEECH RECOGNISER

Since, at the start of the project we did not know the true requirements of a speech recognition device, we chose the commercially available Speech Systems Incorporated Phonetic Engine 500 (SSI PE500)[1] speech recognition development kit (SDK) for the IBM Personal Computer. The PE500 aims to provide for continuous, speaker-independent speech recognition, with a 400,000-word vocabulary. The system is provided with two generic speaker models: American male and American female. The speaker model is static and hence cannot be adapted to a British speaker. Since the development of speaker models is an extensive undertaking, it must be carried out by SSI, under contract.

Words not in the vocabulary can be generated by a generalised phonetic transcription algorithm, giving an almost infinite possible lexicon. The number of active words at any one time is controlled by a context-free rewrite grammar of possible utterances. This is precompiled by the developer before use, and does not allow any adjustments to the syntax structure at run time.

We did not wish to use one of the many 'research' speech recognition systems for a number of reasons, despite their greater applicability to the problem. The foremost reason was our desire not to develop a speech recognition system tailored to our task with the large overhead that this would incur. We wanted to see how good commercial, off-the-shelf packages really are, and of course such packages are generally easier to obtain.

[1] The PE500 is available from Speech Systems, Inc. 2945 Center Green Court South, Boulder, CO 80301-2275, USA. Tel: 303.938.1110 FAX: 303.938.1874

The PE500 is aimed at continuous speech recognition for highly structured, low perplexity, command-control applications. Whilst there is no theoretical limit to the number of active words at any one time, there is a continual degradation in performance as the size of the vocabulary and the ambiguity licensed by the syntax increases. This system is not suited for the highly perplex domain of ATC transmission, but was all we had access to at the time.

THE LBA CORPUS

The LBA Corpus was edited to facilitate the analysis of the domain language and has been manually phrase-tagged with around 50 semantic/functional labels. The creation of discourse and semantic functional phrase tags is intended to enhance the existing context-free grammar in order that it might be partitioned to take advantage of the PE500's ability to switch between applicable syntaxes. The utterances have been grouped into dialogues between the ATC and a particular pilot. The controller may be interacting with several pilots in parallel, in which case each pilot-controller 'thread' constitutes a separate dialogue. The corpus should provide evidence of habitual repeated patterns or structures within dialogues, if they exist. For example, consider the interaction between the pilot of aircraft G-AJCT and the ATC, below. The ATC's utterance ("A:") has been tagged in terms of semantic/functional labels. The number in brackets preceding the utterance is the transmission index.

(166) P: leeds approach good morning golf alpha juliet charlie tango is passing 1400 feet on the heading of 240

(167) A: [CALLSIGN charlie tango CALLSIGN] [GREET leeds good morning GREET] [INFO_ID you are identified INFO_ID] [MAN_HEAD continue heading two four zero MAN_HEAD]

THE TEST MATERIAL

We want to show the effect differing levels of 'linguistic knowledge' can have on speech recognition accuracy. How does the system perform with a large, perplex syntax when compared to partial information about key phrases? Is having a syntax much more accurate than simply having a lexicon? Does use of discourse greatly improve recognition? In order to eventually test different

facets of constraints, test material was chosen to reflect a number of properties. These include:

- use of one or more pieces of key-phrase information within a single utterance.
- use of aircraft identifier, otherwise known as callsign, with other key-phrase information, and with non-key information.
- discourse progression with same pilot, consisting of one complete dialogue
- at least 10 utterances.

Given the above criteria, an interaction in the corpus between the ATC and aircraft 908 was chosen, consisting of 19 utterances by the ATC (see Appendix 1).

The PE500 VoiceMatch Toolkit allows integrated collection and testing of speech material and can offer statistics on the accuracy of the decode. Six speakers were used to record the utterances using a proprietary noise-cancelling microphone. Three of the six were female. Recording occurred in a noise-controlled workspace, whilst an extra set of one speaker were recorded under normal office conditions.

The Toolkit allows the developer to use differing parameter settings when decoding speech into transcribed text. These vary by the *slider* setting and the *language weight* setting. The *slider* setting determines the ratio of accuracy to speed used by the decoder, i.e. how much effort the decoder puts into decoding an utterance. The PE500 has seven predetermined settings, three of which were used, approximately generating an increasing level of effort used by the decoder. The chosen slider settings were hence:

- 0, 3, 6

With each slider setting it is possible to vary the *language weight*, or *transcription penalty* value. This is a negative value which penalises excessive transcription of words, i.e. those output by the decoder. The larger the negative value, the greater the penalty and the fewer words output by the decoder. The weight needs to be optimised so that the correct number of words are transcribed. Values ranged between 0 and -150. Five values were chosen:

- 0 (default - no penalty), -40, -80, -120, -150 (maximum penalty)

MEASURES OF ACCURACY

What constitutes an accurate transcription, and how can this accuracy be graded? PE500's VoiceMatch Toolkit decodes an utterance and then attempts to align it with a template of what the utterance should actually be. This results in a number of words matching the template. Words which occur in the decoded text but not in the template are either deleted or substituted. Words which are in the template but not in the decoded text are inserted. Hence there are a number of measures which can be taken into account when calculating the accuracy of the decoded text. The following reflect those which are readily derived from the VoiceMatch Toolkit:

- number of words in input (in template)
- number of words in output (decoded text)
- number of words correct in output, occurring at appropriate place
- number of words needed to be inserted / substituted / deleted to match input

We chose a measure of accuracy based not only on the number of words correct in the output of the system, but also on the number of words actually output, i.e. transcribed. This compensates for over-generation where many more words are transcribed than occur in the speech.

WE%, the percentage of the number of words correct in the decoded text taking into account the deviation of output to input ratio.

number of words correct / (*number of words in input* + | *number of words in output* - *number of words in input* |) (*100)

where | x | is the absolute value of x.

The above measure was calculated for two scenarios: for all words in the template, regardless of whether or not they are in any of the five "key information phrases" (see Introduction) and for words which are only in one of these five phrases. The test material in Appendix 1 indicates which words fall into either category.

SYNTAX 1: BASE SYNTAX

In order to make comparisons between different syntaxes, the first set of decoding was performed using a 'base' syntax. To set the testing base, the

decoder was tested using what is equivalent to a null syntax. This gives the system no knowledge of utterance structure nor permissible utterance sequences. As required by the PE500, the lexicon of the corpus was provided. The base syntax was simulated using an iterative word category which contained all of the words in the corpus. Thus an utterance could consist of one or more of the words in this category. The lexicon consisted of approximately 380 words.

One problem regarding the results was the inability of the system to cope with the number of words decoded from one speaker, using a default language weight of 0. The memory problem caused the system to ignore the test set. To enable further comparisons to be conducted on the results, dummy values were substituted for these results. In this case, WE% = 0.0.

SYNTAX 2: KEY-PHRASE SPOTTING SYNTAX

The second syntax we tested used the same iterative mechanism as that used in the base syntax. In effect, key-phrases were structurally defined, but could have unrestricted words surrounding and between them. In order to restrict the ambiguity of these non-key words they were limited to what occurred immediately before and after each key-phrase. The words were taken directly from the corpus. This syntax performed a kind of key-phrase spotting and allowed 'unrestricted' speech to occur in the same utterance. It is part way between the previous, lexicon-only syntax, and a full structured syntax.

Since key-phrases were to be recognised, the syntax comprised semantic/functional tags, rather than the conventional phrase structure tags. For example, the key-phrase for changing frequency was represented by a semantic tag "ALTER_FREQUENCY" which then was defined using similar tags. The whole syntax consisted of 47 "tags" or non-terminal symbols and 30 defining rules.

SYNTAX 3: FULL CONTEXT-FREE SYNTAX

The third syntax took the key-phrases of the previous, key-phrase spotting, syntax and combined them with structured non-key ('noise-phrases') so that the entire corpus could be parsed by the whole syntax. The syntax consisted of a total of 98 tags, 29 of which related to the structure of key-phrases and 55 of which related to the structure of non-key phrases. The syntax consisted of 97 defining rules.

SUMMARY OF RESULTS, ALL WORDS

Table 1 below is a summary of the recognition accuracy for the various combinations of slider settings and language weights. The combination with the best average was chosen to represent the best and worst performance for that syntax. The values shown are calculated using the WE% measure based on all words in the template. Following the table is a more detailed summary of the results for each syntax.

Base syntax

The best result was from slider setting 3, language weight -80 with an accuracy of 24.91%. The poorest result of 0% accuracy was due to aforementioned transcription problem. The next worse result was of 9.15% for slider setting 0, language weight -40. The base result taking the average for each combination of slider and language weight was 19.32% for slider 0 and weight -80. For all three slider settings, the best weight to use was -80, whilst the worst was 0. No single utterance was 100% correctly transcribed.

Key-phrase syntax

Again the best results were from using a language weight of -80, with a slider setting of 6. The best result was 26.39%. The poorest performance came from using no language weight (i.e. 0) at 7.45% for a slider setting of 0 and weight of 0. The best average result was for slider setting 6 and weight -80 at 21.67%. No single utterance was 100% correctly transcribed.

Syntax	Slider	SSF	Best	Worst	Average
Base	0	-80	24.65	13.52	19.32
Key-phrase	6	-80	26.39	16.56	21.67
Full	6	-40	64.48	47.63	55.26

Table 1: Summary of results for all words

Full syntax

The best results appeared with the use of low transcription penalties (i.e. weight of 0 and -40), at 68.06% for slider setting 6 and language weight 0. In this case, the greater the penalty, the poorer the results. The lowest was 4.09%, occurring with slider setting 0 and weight -150. The best of the averages was 58.30% with the same settings as for the best result. This setting combination also correctly transcribed a total of 15 utterances in their entirety.

SUMMARY OF RESULTS, KEY-PHRASES ONLY

Table 2 represents the same information as the previous one above. The combination with the best average was chosen to represent the best and worst performance for that syntax. The values shown are calculated using the WE% measure based on only the words which occur in the key-phrases in the template. A more detailed summary of the results for each syntax follows.

Base syntax

As can be seen, there is an insignificant improvement between the accuracy of words in key phrases, and all words in the template. The best result was an accuracy of 26.51% for slider setting 0, language weight -120. The best average result was 20.51 for slider setting 3, language weight -80. For all slider settings, best results were obtained from using language weights of -80 and -120. The poorest results can from using a low language weight, i.e. 0 or -40. No single utterance was 100% correctly transcribed.

Key-phrase syntax

Once again, the best results for each slider setting were from using language weight -80. The best results were 29.07% for slider setting 0, and on average, 22.36% for slider setting 6. The poorest results for each slider setting were from using language weight 0, at 10.04 for slider setting 3.

Full syntax

The best result was from slider setting 6 with language weight -40, at 73.17%. The best of the averages was 64.88% for the same settings. The language weight of -40 gives the best results for all slider settings, and once again, the larger the transcription penalty, the poorer the results. The poorest result was 11.8% using slider setting 3 and language weight -150.

COMMENTS ON RESULTS

The first syntax's use of iteration results in over-transcription of short words. This is demonstrated to its extreme by one speaker's decoded text taking more memory than the system can cope with. As the transcription penalty is increased, fewer words are transcribed and accuracy is improved. The best performance was from using large penalties, up to a certain limit. The largest imposed penalty subsequently degraded performance. There was a little improvement for key phrase words. This, however, was not considered significant.

One would expect that the second syntax would improve the accuracy, at least for the structured key phrases. There was an small increase in accuracy from the first syntax, and again a small improvement between all words and words in the key phrases. A problem with the PE500 is the inability to use any form of weighting mechanism in order to prefer key-phrase words over, say non-key phrase words. This could account for the over transcription of non key-phrase words in similar circumstances as the first syntax. A moderate language weight is optimal in this case.

The third syntax did not rely on the iteration mechanism, but instead consisted of defining rules. This syntax is large and ambiguous but greatly improved recognition. Once again, there is a small increase in performance for those words in the key-phrases. Most surprisingly, however, the best results come from using either no transcription penalty or the smallest. This could reflect the PE500's inability to accurately transcribe syntaxes which make extensive use of the iteration mechanism.

Syntax	Slider	SSF	Best	Worst	Average
Base	3	-80	25.29	16.84	20.51
Key-phrase	6	-80	28.09	17.62	22.36
Full	6	-40	73.17	53.89	64.88

Table 2: Summary of results for words occurring in key-phrases only

The first two syntaxes show that there is little difference between one's choice of slider setting, whereas the third syntax shows the opposite with large differences in performance. Use of the iteration mechanism results in over-transcription, hence requiring a higher transcription rate penalty for better results. This is not the case for the third syntax which gives better results for a low transcription penalty values.

USING HIGHER LINGUISTIC LEVELS: TOWARDS A GRAMMAR OF DISCOURSE

We wish to see the effect that higher levels of linguistic information have on the speech recognition performance. In particular, we would like to explore the effect of using a discourse grammar on what is intuitively a well-structured domain. A large, all-encompassing syntax, such as syntax 3, can be broken down into smaller, well-defined subsets provided that there is a definite distinction between dialogue segments in the domain. This smaller syntax is potentially less ambiguous than the original, containing fewer words and less complicated structures. If this is the case, one would expect that the application of this smaller syntax to result in a higher recognition rate.

To obtain some initial results for such use of a syntax, a further set of experiments were conducted using a single subset of syntax 3. This syntax contained enough information to cover the entirety of the test material. Although the combination of key-phrases was reduced, the full expressiveness of the phrases were preserved. For example, although the new syntax would not allow a callsign followed by a change of frequency, it would allow a callsign followed by a change of heading. The choice of callsign is from the original universe of callsigns and the headings still reflect all of the possible changes in heading.

The revised syntax contained 50 tags, one of which defined the start of the utterance, and 48 rules or word categories. The lexicon consisted of 257 words and the number of sentences which could be produced is comparable with the original syntax (compare with the original: 98 tags, 97 rules and 380 words in lexicon).

Tables 3 and 4 below summarise the results for all words in the test material and for key-phrase words only.

For all words, the best performance of 75.53% came from using a slider setting of 6 and language weight of -40. The trend in results is very similar to those for the full syntax where a greater transcription penalty leads to poorer results. The best average was 66.33% with a slider setting of 6 and no transcription penalty. This is 8.03% higher than the respective original syntax. This combination of slider and penalty gives a total of 26 sentences transcribed without any errors, 11 more than the original syntax.

The best result of 78.92% came from a combination of a slider setting of 6 and no language weight. The best average of 71.28% was obtained from the same settings. This is an increase of 6.4% on the original syntax.

It is not surprising to see the same trends in this syntax as in the original. A low or non-existent language weight gives the best results. An increase of around 8% may not be much but does highlight the increase in performance by using smaller subsets. The subset used in this case was comparable to the original since it was still a large and potentially ambiguous syntax. We hope that the use of smaller subsets, applied through a discourse grammar would lead to greater improvements in performance.

Slider	SSF	Best	Worst	Average	No. Utts Correct
6	0	74.18	59.66	**66.33**	**26**
6	-40	**75.53**	52.75	60.83	25

Table 3: Summary of results for all words using subset syntax

Slider	SSF	Best	Worst	Average
6	0	**78.92**	63.31	**71.28**
6	-40	77.3	67.07	70.89

Table 4: Summary of results for words occurring in key-phrases only, using subset syntax

THE SEGMENTATION OF DIALOGUE

Discourse can be broken into discourse segments which reflect a set of utterances with some properties in common. A discourse segment can be the utterances discussing a certain topic. It can also be the discourse between a set of speakers, in other words, a dialogue. In the ATC application it is helpful to divide the total set of utterances by the ATC and respective pilots into dialogues. For example, a discourse can be the all the utterances by the ATC and pilots between the ATC starting his/her shift and finishing. A dialogue will then be all the utterances concerning the ATC and a particular pilot. Individual dialogues can be further divided into segments indicating the flow of the discourse.

For this approach to work, we need a method for dividing the dialogue into maximally distinct discourse segments. Unfortunately, discourse grammar is a loosely-formalised area with few formal guiding principles, so we turn to automatic "Machine Learning" techniques for segmentation. Corpus-based statistical clustering techniques have been applied to other segmentation/labelling problems in NLP, e.g. clustering words into word-classes [Atwell & Drakes 87, Hughes 94, Hughes & Atwell 94], and clustering texts into related languages [Churcher 94, Souter et al. 94].

The automatic segmentation of a dialogue should provide the basis for the generation of a discourse grammar. A discourse grammar would allow a speech recognition system to apply syntaxes which have immediate relevance to the utterances being spoken at the time. Furthermore, additional language models can be applied to the discourse structure as it evolves.

DIALOGUE SEGMENTS

The ATC Approach corpus is already divided into utterances between a pilot and the ATC. Each set can be thought of as a discourse segment.

One feature of the ATC dialogues is that they can be interleaved with one another, posing the problem of dialogue tracking. This has partially been tackled in [Grosz 86] and other modelling strategies.

As an example, a dialogue can be split up into functional units: a segment can be thought of as a GREETING exchange, some INFORMATION exchange and a SIGNING OFF exchange, where a protocol for ending the dialogue exists. With other discourse segments, each of these units may consist of more utterances or fewer, or introduce other, finer units.

METHOD OF SEGMENTATION USED

In order to assist the generation of a discourse grammar, it is useful to look at the semantic labels used throughout the corpus. Here is an example dialogue extracted from the corpus. Only the semantic tags are shown for clarity:

```
1 (34)  [+CALL]  [GREET]
2 (36)  [+CALL]  [AFFIRM]  [INFO_CURRENT]
[+INF_QNH]
3 (38)  [+CALL]  [AFFIRM]
4 (39)  [+CALL]  [REQ_CONFIRM]
5 (41)  [+CALL]  [THANKS]  [INFO_POS]
[INFO_END]  [+ALT_FR]  [INFO_LOC]
6 (43)  [BYE]
```

The simplest method of automatically dividing the discourse is to divide it into roughly equal parts based on the number of sub-segments desired. For example, two 'clusters' would divide the discourse into utterances 34-38, 39-43. Three 'clusters' would divide it into 34-36, 38-39, 41-43.

Taking each set of clusters for all discourse segments, the similarity between different sub-segments can be calculated using some measure. We decided to initially try our approach using the key information phrase labels only, ignoring the noise information.

COMMENTS ON CLUSTERING APPROACH CHOSEN

The above segmentation technique is very simple and thus suffers from a number of disadvantages. As can be seen from the example, choosing three or less clusters will result in the incorrect placing of utterance 36 into the first sub-segment.

(a) +CALL GREET (b) +CALL ALT_FR (c) +CALL ALT_HD

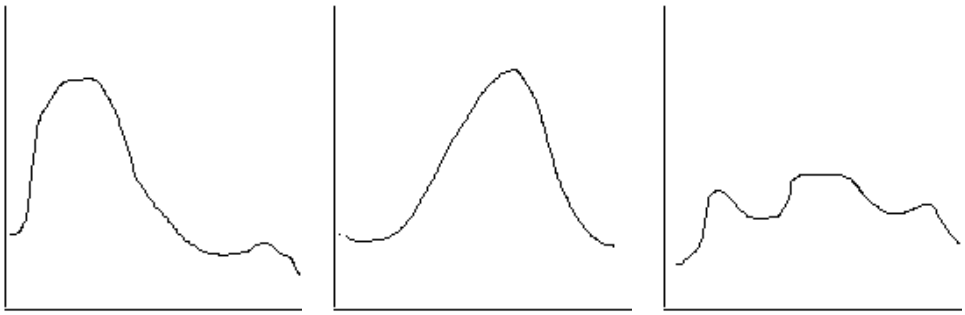

Figure 1: (Y: frequency of rule; X: utterance position in segment)

Dividing the segment by hand into functional units resulted with utterance 36 being placed into sub-segment 2, i.e. the INFORMATION exchange unit. The strict division of dialogue into 'roughly' equal parts results in utterances being placed into wrong sub-segments.

One way to view the discourse segment is as a continuum of semantic tags, both because of the above problem and due to the more or less uniform distribution of some common sequences of tags. A technique which can be adapted for this purpose is explained in [Hughes 94]. Hughes uses a normalised frequency distribution of word / word-type position within a sentence. For example, consider the frequency distributions in figure 1 for three tag sequences.

The example tag sequences show the following:

(a) a definite peak towards start of discourse segment
(b) a definite peak towards end of discourse segment
(c) no definite peak - a more or less uniform distribution throughout discourse segment

Frequency distributions and hence derived probability distributions can be used by the discourse level instead of using distinct segments to distinguish between differing sections of discourse. This approach combats the problem of utterances which are divided into the incorrect segment.

MEASURE OF SIMILARITY BETWEEN SUB-SEGMENTS

A bigram frequency model was generated for each cluster set. This simple model of sequences of tags in clusters allowed a correlation coefficient to be calculated and clusters within the same set compared.

First, the corpus of dialogues was divided according to the number of clusters chosen, then given to an n-gram model generation program. The statistical package, SPSS was used to generate the correlation coefficient between different pairs of clusters. This data was then used by a clustering package to generate dendograms indicating the similarities between the clusters. The clustering algorithm used was Ward's which uses a statistically based dissimilarity measure [Ward 63, Wishart 69] favoured by Hughes [Hughes 94] for clustering words.

CLUSTERING RESULTS

Four sets of clusters were generated, using clusters of number 3, 4, 5 and 6. The dendograms of three and five clusters in figures 2 and 3 below show the grouping of different clusters, the closer to the right a join between two clusters, then the greater the similarity between them.

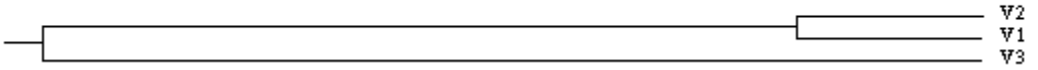

Figure 2: Dendogram using three clusters

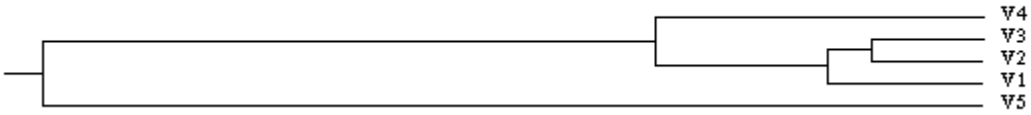

Figure 3: Dendogram using five clusters

CONCLUSIONS FROM INITIAL CLUSTERING METHOD

The correlation values showed that many of the clusters were very similar. The greater number of clusters chosen, the greater the variance between them. At five clusters, the correlation coefficient between the first and the last cluster drops to 0.7542, the lowest value present.

Another approach which should be considered is that of an intention or plan level, one level higher than the discourse level. Just as syntax is considered as parts of discourse segments, discourse can be considered as parts of a plan. For example, the frequency distribution of tags in one discourse segment where a pilot intends to land at the airport may be quite different to that of one where the pilot is taking off and leaving the ATC area. This difference in the plan or intention of the pilot should be taken into consideration when segmenting the discourse.

Dividing dialogues into sets which have the same intention / plan generates a problem of its own. A much greater number of segments are required, and hence a larger corpus, in order to provide adequate numbers of instances.

There has to be evidence that each discourse sub-segment is distinct enough from its neighbours in order to create a discourse grammar which is more effective than simply using a single syntax, [Churcher et al. 95]. Initial correlation coefficients show that there is little difference between successive sub-segments. However, this may be the result of using a very simple and error-prone clustering method. Further work using a dynamic clustering method or frequency distributions should be considered before concluding that a discourse grammar is unfeasible in this instance.

USE OF CONTEXTUAL INFORMATION

The use of a natural language component to constrain the output of the system could increase the system's recognition performance. In this domain, there is also a wide range of contextual knowledge which could be incorporated into the system, either by means of a database containing information applicable to the local area around the ATC, or by controlling the speech recognition unit itself. The contextual knowledge which could be applicable includes the following:

1. Current callsigns being used in airspace.
2. Current transponder settings (squawks) being used by aircraft.
3. Current pressure settings of the local area, etc.
4. Regional geographical landmarks.
5. Transponder code ranges used at LBA.
6. Radio frequencies used at or around LBA.
7. Runway identifiers used at LBA.

The first three items contain information which exists for differing periods of time. For example, the callsigns currently being used exist only for the duration that the pilot is in LBA airspace. The remainder of the information is local to LBA, itself.

As an example of how this information may be used, consider the transponder or 'squawk' codes which range in value from 0400 to 0420, in octal and that only one aircraft in LBA airspace can have a

particular code. This information can assist the choice of the correct code.

CONCLUDING REMARKS

The above results show the advantages of using a full, context-free syntax in the domain of Air Traffic Control transmissions using the formalism provided by the PE500. The use of key-phrase spotting with the mechanism of iteration produced inaccurate transcriptions with results little better than not having a syntax at all. Some form of weighting mechanism for the key-phrases may be of value in increasing the performance.

The PE500 is designed for low vocabulary, low perplexity, command-control speech recognition. It is not designed to perform well on large and ambiguous syntaxes and this is reflected by the results. Its performance is poor when compared to the research systems used in the recent ARPA Wall Street Journal competition [Collingham 94, ARPA 94] but it must be noted that the system was not "trained" nor optimised for the domain or speakers, except that a syntax was provided. Hence, this set of experiments have been a comparative study of the use of differing levels of linguistic information using a commercially available speech recogniser.

The use of a discourse grammar to divide the large syntax into smaller syntaxes may improve performance. The smaller syntaxes may perform better due to lower perplexity and ambiguity and could be applied as the discourse progresses. Such use of higher level "linguistic knowledge" together with contextual information should, in theory, improve the performance of the continuous speech recogniser. The representation of such a discourse grammar is not clear. Automatic clustering of a corpus may assist the identification and representation of distinct dialogue segments, if they exist for a particular domain language.

BIBLIOGRAPHY

[ARPA 94] Proceedings of the ARPA Spoken Language Systems Technology Workshop, March 1994.

[Atwell & Drakos 87] E Atwell & Nikos Drakos. "Pattern Recognition Applied to the Acquisition of a Grammatical Classification System from Unrestricted English Text" in Bente Maegaard (ed), "Proceedings of the Third Conference of European Chapter of the Association for Computational Linguistics", pp56-63, New Jersey, Association for Computational Computational Linguistics. 1987.

[Churcher 94] GE Churcher. "A comparison of the bigraph and trigraph approaches to language identification", Undergraduate Project, School of Computer Studies, Leeds University, Leeds. 1994.

[Churcher et al. 95] GE Churcher, ES Atwell, DC Souter. “Developing a Corpus-Based Grammar Model Within a Continuous Commercial Speech Recognition Package”. Research Report Series, Report 95.20, School of Computer Studies, Leeds University, Leeds. 1995.

[Collingham 94] R Collingham. “An Automatic Speech Recognition System for use by Deaf Students in Lectures”, Unpublished PhD Thesis, Laboratory for Natural Language Engineering, Dept. Computer Science, University of Durham. September 1994.

[Grosz 86] BJ Grosz. “Attention, Intentions, and the Structure of Discourse”. Computational Linguistics, Vol 12, No. 3, 1986

[Hughes 94] J Hughes. “Automatically Acquiring a Classification of Words”. Ph.D.

Thesis, School of Computer Studies, Leeds University, Leeds. 1994.

[Hughes & Atwell 94] John Hughes & E Atwell. "The automated evaluation of inferred word classifications" (with John Hughes) in Tony Cohn (ed), "Proceedings of European Conference on Artificial Intelligence (ECAI)", pp535-539, Chichester, John Wiley. 1994.

[PE500 SDK] PE500™ System Development Kit, Syntax Development Guide. 1994. Available from Speech Systems, Inc. For contact details see footnote 1.

[RTF CAP413] Radiotelephony Manual (CAP 413), Civil Aviation Authority, London, 1992.

[Souter et al. 94] DC Souter, GE Churcher, J Hayes, J Hughes & S Johnson, "Natural Language Identification using Corpus-Based Models", in Hermes, "Journal of Linguistics", 13-1994, pp183. 1994.

[Ward 63] JH Ward. “Hierarchical Grouping to Optimize an Objective Function”. Springer-Verlag, Berlin. 1963.

[Wishart 69] D Wishart. “An Algorithm for Hierarchical Classifications”. 22, pp 165. 1969.

APPENDIX 1

TEST 908 SENTENCE LIST (KEY SUB-PHRASES ARE UNDERLINED)

1. nine zero eight standby for further descent expect vector approach runway three two information charlie current q n h one one zero five and q f e nine nine one millibars
2. nine zero eight report your heading
3. nine zero eight roger continue that heading descend to altitude four thousand feet leeds q n h one zero one five
4. flight knightair nine zero eight turn left heading zero eight five
5. two eight nine zero eight leeds
6. runway one four is available vectors to a visual approach if you wish give you about two seven track miles to touchdown
7. expect a visual approach runway one four q f e nine nine zero millibars proceed descent altitude three thousand five hundred feet
8. q f e nine nine zero millibars for runway one four
9. two eight nine zero eight turn right heading one zero zero
10. nine zero eight roger maintain
11. two eight nine zero eight descend to height two thousand three hundred feet q f e nine nine zero millibars
12. on that heading you'll be closing for a visual final that's about five miles you've got approximately one one track miles to touch down
13. nine zero eight descend height one thousand five hundred feet q f e nine nine zero
14. nine zero eight your position five north west of the field report as you get the field in sight
15. zero eight nine zero eight turn right heading one four zero
16. zero eight nine zero eight descend to height one thousand two hundred feet
17. on the centre line three and a half miles to touchdown
18. thanks happy to continue visual
19. contact the tower one two zero decimal three