University of Leeds
SCHOOL OF COMPUTER STUDIES
RESEARCH REPORT SERIES
Report 96.04

Using a Commercial Speech Recogniser Within the Domain of Air Traffic Control

by

Gavin E Churcher, Clive Souter & Eric S Atwell
Division of Artificial Intelligence

February 1996

We are grateful to the University of Leeds and Visionair Ltd for funding this research

Abstract

We have taken an off-the-shelf, commercial continuous speech recogniser and conducted tests using three syntaxes for the domain of Air Traffic Control. The syntaxes are based on a corpus of transmissions between the ATC and pilots and reflect three differing levels of "linguistic" knowledge. The first represented the system where, in effect, there would be no syntax but a lexicon of all words in the corpus. The second took a partial look at syntactic information by using a key phrase spotting mechanism. The third represented the entire syntax of the corpus. Initial experiments show that key phrase spotting is insignificantly more accurate than no syntax at all, whilst use of a complete syntax can improve performance, to a point. The benefits of a discourse grammar are 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 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). 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 strict context-free 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.

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 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 structureless lexicon? Does use of discourse greatly improve recognition? In order to eventually test different facets of constraints, the 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.

---

[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 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 outputted by the decoder. The larger the negative value, the greater the penalty and the fewer words outputted 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 (see Appendix 4 for examples). 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

Four accuracy measures can hence be derived:

WP% - percentage of words correct from the number words occurring in the template.

I.e.: *number of words correct in output* / *number of words in input* (*100)

WT%, percentage of words correct out of the number of words in the decoded text. This is useful since an overgenerative grammar would produce a large number of words, many of which could be correct but have many 'deleted' words occurring between each correct word.

I.e.: *number of words correct in output* / *number of words in output* (*100)

WER, the proportion of words which have to be inserted, substituted or deleted in comparison to the number of words matching the template.

I.e.: *number of words inserted, substituted & deleted* / *number of words in input*

WE%, the percentage of the number of words correct in the decoded text taking into account the deviation of output to input ratio. This would be a combination of both WP% and WT%:

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 measure, WE% was chosen as an indicator of the accuracy of the decoded text since it took into account the issue of overgeneration of words in relation to the number of words in the template.

The above measures were 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.

Syntaxes used

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.

The following example illustrates the structure of the "null" syntax used by the PE500 in this instance. It makes use of a construct which iterates a word category one or more times and uses this to allow a sentence to comprise of one or more words without any *'syntactic'* information.

S --> {* +WORD *}	Sentence *'S'* rewrites as one or more instances of *'+WORD'*
+WORD == a abeam approach ...	Word category contains all the words in the lexicon

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.

Results for Base syntax, all words in template

The following tables represent summaries for each combination of slider setting with language weight (SSF) for the accuracy measure, WE%. The best and worst and average accuracies are indicated for all seven speakers for each slider setting and language weight. Values in bold indicate the best or worst slider/language weight setting.

Slider	SSF	Best	Worst	Average
0	0	9.75	**0**	7.29
0	-40	18.6	9.15	13.88
0	-80	24.65	13.52	**19.32**
0	-120	23.65	12.21	17.12

0	-150	18.13	11.45	13.80
3	0	9.5	**0**	7.01
3	-40	18.92	9.72	13.74
3	-80	**24.91**	15.48	19.17
3	-120	22.74	14.24	17.92
3	-150	18.62	10.53	14.19
6	0	9.45	**0**	6.94
6	-40	18.87	9.26	13.50
6	-80	23.86	15.58	19.29
6	-120	22.37	15	17.65
6	-150	18.79	10.47	14.60

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.

Results for Base syntax, key-phrase words in template

The following table summarises the results for the base syntax, taking only key-phrases into consideration.

Slider	SSF	Best	Worst	Average
0	0	15.18	**0**	9.99
0	-40	21.13	13.13	17.51
0	-80	25.86	14.67	20.38
0	-120	**26.51**	13.17	19.72
0	-150	23.6	14.11	17.53
3	0	14.2	**0**	9.28
3	-40	22.16	12.63	17.36
3	-80	25.29	16.84	**20.51**
3	-120	24.1	14.2	20.01
3	-150	23.31	12.35	17.47
6	0	14.15	**0**	9.26
6	-40	22.16	12.12	17.13
6	-80	23.86	16.67	20.15
6	-120	23.35	14.88	19.96
6	-150	22.98	12.35	18.08

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.

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.

As an example, consider this extract from the syntax which indicates the unrestricted, unstructured, non-key words surrounding the structured key-phrase words:

S --> {* (+B_AH) *} +ALT_HEAD {* (+A_AH) *}	Sentence *'S'* rewrites as
+B_AH == copied good morning roger thanks ...	zero or more instances of any word in the *+B_AH* category
+ALT_HEAD --> turn {left | right} ...	followed by a structured instruction to change heading
+A_AH == altitude and approach closing delta ...	followed by zero or more instances of any word in *+A_AH* category

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 and 30 defining rules.

Results for key-phrase spotting syntax, all words in template

The table below summarises the results for the key-phase spotting syntax.

Slider	SSF	Best	Worst	Average
0	0	11.41	**7.45**	9.44
0	-40	19.35	12.24	15.89
0	-80	25.17	16.29	20.43
0	-120	23.84	14.62	17.82
0	-150	20.06	10.26	12.88
3	0	11.76	8.07	9.39
3	-40	19.94	12.17	15.84
3	-80	25.52	16.61	21.19
3	-120	23.2	14.72	18.13
3	-150	19.46	11.92	14.21
6	0	11.44	8.07	9.43
6	-40	19.89	13.25	15.89
6	-80	**26.39**	16.56	**21.67**
6	-120	23.68	15.11	18.50
6	-150	19.05	12.17	14.85

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.

Results for key-phrase spotting syntax, key-phrase words in template

The following table summarises the results for the key-phrase spotting syntax, taking only key-phrases into consideration.

Slider	SSF	Best	Worst	Average
0	0	16.19	10.19	12.60
0	-40	24.34	12.97	19.03
0	-80	**29.07**	16.16	21.73
0	-120	26.47	15.24	19.73
0	-150	25.61	12.5	16.21
3	0	17.25	**10.04**	12.54
3	-40	23.32	12.5	19.24
3	-80	27.84	17.1	21.96
3	-120	25.75	16.36	20.15
3	-150	25.47	13.66	17.85
6	0	16.56	10.27	12.53
6	-40	21.68	14.52	19.08
6	-80	28.09	17.62	**22.36**
6	-120	25.9	16.36	20.54
6	-150	24.22	13.66	18.44

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.

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. The key-phrase tags used can be seen in Appendix 2.

Results for full syntax, all words in template

The following table represents the summary of the results for the fully structured syntax. As an additional column, the number of utterances transcribed 100% correctly is indicated.

Slider	SSF	Best	Worst	Average	No. Utts Correct
0	0	27.75	19.05	23.19	10.00
0	-40	39.66	17.92	24.02	8.00
0	-80	26.55	8.61	18.29	5.00
0	-120	18.72	9.45	12.89	3.00
0	-150	14.09	**4.09**	8.57	2.00
3	0	55.4	26.42	41.98	12.00
3	-40	55.97	33.63	43.20	13.00
3	-80	50.46	29.41	35.68	10.00
3	-120	32.98	12.92	24.58	5.00
3	-150	21.6	5.65	15.72	3.00
6	0	**68.06**	47.7	**58.30**	15.00
6	-40	64.48	47.63	55.26	**16.00**
6	-80	51.71	35.63	44.19	11.00
6	-120	40	18.27	32.08	7.00
6	-150	28.87	8.81	23.25	6.00

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.

Results for full syntax, key-phrase words in template

The table below indicates the results as for the above table, but only taking key-phrase words into consideration.

Slider	SSF	Best	Worst	Average
0	0	50.9	24.07	33.29
0	-40	61.59	29.01	38.66
0	-80	50.31	19.75	35.04
0	-120	41.88	18.63	27.11
0	-150	27.5	13.04	18.78
3	0	65.5	39.08	51.05
3	-40	68.48	45.73	56.53
3	-80	65.03	43.21	51.45
3	-120	51.23	23.46	41.21
3	-150	41.88	**11.8**	29.31
6	0	69.28	50.56	62.13
6	-40	**73.17**	53.89	**64.88**
6	-80	66.67	49.07	59.24
6	-120	63.98	34.57	49.44
6	-150	51.55	21.12	41.14

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.

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).

Below are the tables for all words in the test material and for key-phrase words only.

Results for subset syntax, all words

Slider	SSF	Best	Worst	Average	No. Utts Correct
0	0	52.92	23.66	36.60	16
0	-40	56.39	26.53	34.44	16
0	-80	35.71	11.93	24.83	11
0	-120	22.91	11.38	17.11	7
0	-150	18.76	4.89	11.00	4
3	0	68.12	49.51	56.07	21
3	-40	57.99	41.83	49.51	21
3	-80	55	36.21	43.11	15
3	-120	39.94	23.39	31.90	10
3	-150	30.03	**2.69**	20.64	8
6	0	74.18	59.66	**66.33**	**26**
6	-40	**75.53**	52.75	60.83	25
6	-80	55	43.45	49.50	18
6	-120	44.84	30.22	36.51	11
6	-150	38.15	9.58	26.25	9

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.

Results for subset syntax, key words only

Slider	SSF	Best	Worst	Average
0	0	65.48	33.54	48.37
0	-40	72.56	37.65	52.15
0	-80	65.03	24.84	45.71
0	-120	49.38	18.63	36.33
0	-150	40.62	9.94	24.95
3	0	72.12	56.9	64.46
3	-40	72.56	54.6	63.45
3	-80	69.94	49.69	60.68
3	-120	58.75	40.37	52.06
3	-150	51.85	**4.97**	39.14
6	0	**78.92**	63.31	**71.28**
6	-40	77.3	67.07	70.89
6	-80	71.6	60.25	65.50
6	-120	61.73	48.15	55.89
6	-150	57.14	22.36	46.16

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.

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 first syntax which simulated a grammar with only a lexicon and no model of syntactic structure peaked at 24.91% accuracy for all words and 26.51% for key-phrase words. The second syntax using a key-phrase spotting technique peaked at 26.39% for all words and 29.07% for key-phrase words. The final syntax which used a semantic/functional context free grammar peaked at 68.06% for all words and 73.17% for key-phrase words. It is interesting to also note that the use of a noise controlled environment made little difference to the transcription accuracy. This can be ascribed to the use of the proprietary noise-cancelling microphone.

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.

Bibliography

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

[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.

[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.

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

Appendix 2

Key Phrase Semantic Labels for Fully Structured Grammar

Below are a list of semantic tags used to represent the structure of key information phrases as used in syntax 3. The list is akin to immediate dominance rules where no order is inferred in the daughters. For an example of a parsed sentence, see Appendix 3.

ALT_CLIMB+	Instruction to climb
FL+	Relevant altitude expressed as a flight level
HEIGHT+	Relevant altitude expressed as a height in feet
DIGIT+	Digit zero to nine
ALT_DESC+	Instruction to descend
HEIGHT+	Relevant altitude expressed as a height in feet
DIGIT+	Digit zero to nine
ALT_HEAD+	Instruction to change heading
DIGIT+	Digit zero to nine
ALT_FREQ+	Instruction to change radio frequency
LOCAL_FREQ+	Structure for common, local frequencies
LEEDS_TOWER+	Leeds frequency
E_MIDLANDS+	East Midlands frequency
WARTON_RADAR+	Warton Radar frequency
LEEMING+	Leeming frequency
LINTON+	Linton frequency
MAN_CONTROL+	Manchester frequency
ALT_SQUAWK+	Instruction to change SSR setting
LOCAL_SSR+	Structure for set of local SSRs
OCTAL+	Digit zero to seven
CALLSIGN+	Structure for callsign
COMMERCIAL+	Structure for commercial flights
COMPANY+	Structure for company aircraft
HELICOPTER+	Structure for helicopters
NON_DESC+	Structure for non-descript callsigns
ALPHA+	International alphabet (i.e. alpha, beta ...)
DIGIT+	Digit zero to nine
INFO_QFE+	Information on the current QFE
QFE+	Structure for QFE
LOCAL_RW+	Structure for local runways
DIGIT+	Digit zero to nine
INFO_QNH+	Information on the current QNH
QNH+	Structure for QNH
LOCAL_AREA+	Structure for local area indication
DIGIT+	Digit zero to nine

Appendix 3

Examples of parsed test sentences

The sentences below are taken from Appendix 1 and reflect how the key-phrase tags are used in Appendix 2. Tag +ATC is equivalent to the sentence 'S' rewrite tag.

```
(+ATC (CALLSIGN+
                    (COMMERCIAL+ flight
                                                (COMPANY+ knightair) (DIGIT+ nine) (DIGIT+ zero) (DIGIT+ eight)))
        (ALT_HEAD+ turn left heading (DIGIT+ zero) (DIGIT+ eight) (DIGIT+ five)))
```

flight knightair nine zero eight turn left heading zero eight five

```
(+ATC (CALLSIGN+
                    (COMMERCIAL+ (DIGIT+ nine) (DIGIT+ zero) (DIGIT+ eight)))
        (ALT_DESC+ descend height
                    (HEIGHT+ (DIGIT+ one) thousand (DIGIT+ five) hundred feet))
        (INFO_QFE+ q f e
                    (QFE+ (DIGIT+ nine) (DIGIT+ nine) (DIGIT+ zero))))
```

nine zero eight descend height one thousand five hundred feet q f e nine nine zero

```
(+ATC (ALT_FREQ+ contact
                                (LOCAL_FREQ+
                                                (LEEDS_TOWER+ the tower one two zero decimal three))))
```

contact the tower one two zero decimal three

Appendix 4

Example transcriptions for best and worst recogniser settings

The examples show what was actually transcribed by the system with the prompted sentence.
#_ indicates a silence word substitute

Base syntax, Best: Slider setting 3, SSF -80

PROMPT			nine	zero	eight			report	your	heading		
TRANS	#_	#_	nine	zero	might	up	m	little	high	hand	#_	#_

PROMPT			nine	zero			eight	roger		continue	that	
TRANS	#_	#_	nine	zero	a	go	edge	and	#_	continue	ahead	

PROMPT	heading	descend	to	altitude	four	thousand	feet		leeds
TRANS	d	and	descent	slowly	traffic	thousand	leeds	#_	leeds

PROMPT	q	n	h	one	zero	one	five		
TRANS	to	your	h	position	own	four	and	#_	#_

Base syntax, Worst: Slider setting 0, SSF -40

PROMPT						two	eight	nine	zero	eight	leeds		
TRANS	#_	#_	c	ready	mind	is	your	or	edge	eight	the	descent	

PROMPT			two	eight		nine	zero	eight				turn	
TRANS	#_	#_	to	your	a	mind	zero	eight	s	got	ever	i	and

PROMPT	right	heading			one		zero	zero		
TRANS	eight	heading	when	is	your	is	e	and	#_	#_

Key-Phrase spotting syntax, Best: Slider setting 6, SSF -80

PROMPT			nine	zero	eight	report	your	heading		
TRANS	#_	#_	nine	zero	eight	abeam	little	hand	#_	#_

PROMPT			nine	zero	eight	roger		continue	that	heading	
TRANS	#_	#_	nine	zero	eight	position	#_	continue	ahead	d	and

PROMPT	descend	to	altitude	four	thousand	feet		leeds	q	n	
TRANS	and	descent	slowly	traffic	thousand	leeds	#_	leeds	to	your	

PROMPT	h	one	zero	one	five		
TRANS	h	position	own	four	and	#_	#_

Key-Phrase spotting syntax, Worst: Slider setting 0, SSF 0

PROMPT															
TRANS	#_	#_	c	a	a	give	m	line	is	your	or	or	m	h	

PROMPT	two	eight	nine	zero				eight	leeds						
TRANS	two	e	the	descent	#_	#_									

PROMPT				two		eight		nine	zero						
TRANS	#_	#_	#_	two	your	eight	m	line	zero	a	eight	s	#_	e	

PROMPT							eight							turn	right
TRANS	two	of	your	i	and	eight	ahead	and	e	when	is	your	in		

PROMPT	heading	one	zero	zero											
TRANS	the	is	e	and	#_	#_	#_								

Fully structured syntax, Best: Slider setting 6, SSF 0

PROMPT						nine	zero	eight		report	your				
TRANS	#_	#_	#_	#_	nine	zero	eight	#_	go	ahead	#_	#_			

PROMPT	heading														
TRANS															

PROMPT				nine	zero	eight		roger				continue			
TRANS	#_	#_	#_	nine	zero	eight	go	ahead	and	#_	#_	continue	on		

PROMPT	that	heading		descend	to	altitude	four	thousand	feet						
TRANS	that	heading	and	descend		altitude	four	thousand	feet	#_	#_				

PROMPT	leeds		q	n	h		one	zero	one	five					
TRANS	leeds	#_	q	n	h	#_	one	seven	one	four	#_	#_	#_		

Fully structured syntax, Worst: Slider setting 0, SSF -150

PROMPT			two	eight	nine	zero	eight	leeds							
TRANS	#_	#_			nine	zero	eight	leeds	#_	#_					

PROMPT			two	eight	nine	zero	eight		turn	right	heading	one			
TRANS	#_	#_	two	eight	nine	zero	eight	#_	turn	right	heading	one			

PROMPT	zero	zero													
TRANS	seven	zero	#_	#_											

Subset syntax, Best: Slider setting 6, SSF 0

PROMPT						nine	zero	eight		report	your	heading			
TRANS	#_	#_	#_	#_	nine	zero	eight	#_	go	ahead	maintain	#_			

PROMPT				nine	zero	eight		roger				continue		
TRANS	#_	#_	#_	nine	zero	eight	go	ahead	and	#_	#_	continue	on	

PROMPT	that	heading		descend	to	altitude	four	thousand	feet		
TRANS	that	heading	and	descend		altitude	four	thousand	feet	#_	#_

PROMPT	leeds		q	n	h		one	zero	one	five		
TRANS	leeds	#_	q	n	h	#_	one	seven	one	four	#_	#_

Subset syntax, Worst: Slider setting 3, SSF -150

PROMPT			two	eight	nine	zero	eight	leeds				
TRANS	#_	#_			nine	zero	eight	leeds	#_	#_		

PROMPT			two	eight	nine	zero	eight		turn	right	heading	one
TRANS	#_	#_	two	eight	nine	zero	eight	#_	turn	right	heading	one

PROMPT	zero	zero		
TRANS	seven	zero	#_	#_