

A HYBRID APPROACH TO SPOKEN QUERY PROCESSING IN DOCUMENT RETRIEVAL SYSTEM*

Nathalie Colineau

Human Computer Interaction Lab
Thomson-CSF, LCR,
F-91404 Orsay Cedex, France
email: nathalie.colineau@lcr.thomson-csf.com

Ariane Halber

Department of Computer Sciences
ENST (Ecole Nationale Supérieure des Télécommunications)
46 rue Barrault, F-75634 Paris Cedex 13
email: halber@inf.enst.fr

Abstract

In the context of the THISL spoken document retrieval system, we present a hybrid approach to spoken query processing, which enables to increase recognition rates and to extract relevant information for the application. The query processing is distributed between grammar and language model, based on the assumption that a query can be decomposed in two relatively independent parts; the addressing form, which is parsed with a grammar, and the queried content, which is scored with a domain specific language model. Our aim is to retrieve the content sequence, which allows us to consult the database, but also, to keep information about the query formulations in order to develop an interaction between the user and the retrieval engine. This leads us to work closely with the speech recogniser and to carry out together the recognition and the query analysis.

1. Introduction

Speech recognition technology has now reached a stage where it can reasonably provide baseline systems for spoken interfaces. A trivial approach would consist in using a speech recogniser as a black box, which transcribes an acoustic signal in an utterance. This utterance would provide the input of the retrieval engine connected to a documentary database. However, the use of a speech recogniser requires to take into account the recogniser errors and characteristics of spontaneous speech. In research today, it is even far to be solved. A credible alternative would consist in introducing as soon as possible morphological, syntactical and semantic knowledge in the decoding process.

1.1. What recognition output?

Several works [8, 18, 16, 9, 17, 4] show that the best acoustic hypothesis is often insufficient. They emphasise it is necessary to keep the N best hypotheses or the word graph, to perform complementary processing and to improve the recognition rate. Indeed, a grammatical approach may become essential as soon as the application is extended in such way that more complicated grammatical constructions need to be recognised.

1.2. Query understanding

One of the difficulties lies in the query interpretation. Most of the processing are based either on a keyword search [1, 2], or on the use of local grammars [12, 13], or other matching techniques [6]. In all cases, only some informative units, judged relevant for the application, are extracted. This kind of shallow parsing is robust and relatively easy to develop, since complex linguistic phenomena do not have to be taken into account. When the queries are complex, i.e. when the user can ask a spontaneous request and not only give a sequence of keywords, this kind of analysis seems to be insufficient. If we want to introduce understanding, or to be able to lead an interaction, it is necessary to turn towards a complete linguistic analysis [18, 4]. Yet, it is not always possible to develop an exhaustive grammar in documentary base interrogation.

1.3. Spoken query analysis

In the context of a document retrieval application, any grammar of queried topics should be very large and complex, since it should include the language of all the documents. It is thus illusory to try to develop a grammar that would cover syntactically all *content sequences*.

For the THISL project, ABBOT language model has been trained on the British Broadcast News, and not on the user's queries. So, to define the THISL grammar, we have just worked on several examples of typical request given by the BBC. The difficulty is to achieve the recognition of the uttered queries and then analyse them, given that the ABBOT language model was not trained on the queries themselves but on the indexed documentary base. Indeed, the speech recogniser can neither guaranty to recognise the user's query, nor to produce a grammatical output. So, we have decided not to analyse the best ABBOT output, but to work on the word lattice built by the speech recogniser during its decoding stage. Indeed, the word lattice can be seen as a set of hypotheses where it is possible to find a solution, to rebuild completely or partially a query.

* This work has been supported by EC funding, as part of THISL project, ESPRIT-LTR 23495

Our aim is twofold. It consists on the one hand in getting the keyword sequence for the documentary base interrogation, and on the other hand in keeping information about the query itself, in order to develop a small interaction between the user and the retrieval engine. Indeed, linguistic phenomena as anaphora, ellipses or logical connectors can appear in queries. They require a real linguistic analysis in order to be able to understand the query. Our strategy is to develop a hybrid approach of query processing where structured linguistic information (query grammar) collaborate with others kinds of information (the LM and the speech recogniser acoustic scores).

2. System Overview

The THISL spoken document retrieval is based on ABBOT large vocabulary speech recognizer [1], developed by Cambridge University, Sheffield University and SoftSound¹ [11]. It aims at giving journalists an easy access to a database of transcribed broadcast programs. Indexing and retrieval are performed by PRISE (NIST) [10].

2.1. The Speech Recogniser

Spoken queries to the document retrieval system are recognised by ABBOT, using the same LM (Language Model) as the one used to transcribe the broadcast programs and constitute the document-base. We will refer to this LM as the domain LM.

Experiments conducted along this line [1, 2] take *in-domain* spoken queries, i.e. direct query expressing topical questions, rather than containing spontaneous formulation embedding as was collected for our experiment (see Table 1). These previous experiments present a baseline recognition around 30%. This is approximately the figure we get as well for an expert speaker, despite the added difficulty of recognizing somewhat out-of-domain formulation embedding, but a naive speaker performs quite worse, see results in S. 6.

In any case, the precision and recall of content keywords are too low to rely on 1-best acoustic hypothesis only (see Table 9). That is why, primarily, we keep a word lattice as output.

2.2. Word-Graph Analysis

The word lattice produced by the speech recogniser is treated like a weighted finite-state automaton, it is determined and minimised with the FSM (Finite State Machine) tools developed at AT&T² [14, 3].

The analysis fully succeeds if the parser finds a grammatical path from the initial state to the final state in the word-graph, that corresponds to the grammar constraints. If it fails to do so, we lead partial analyses.

The aim is to discover as much high constituents as possible from the initial state to the final state in the word-graph. We determine then a first set of best grammatical paths. After this, the content sequences, which are segmented in these best parses, are extracted and ranked using the domain LM. Details on the grammar are given in Section 3, on the graph parsing in Section 4, and on the graph search in Section 5.

2.3. Document Retrieval

The best content sequence can be piped to the information retrieval engine, which is basically driven by significant keywords. As we have not yet plugged this module, we will report here results on significant keyword recognition, instead of document retrieval success.

A more refined system would first interpret the grammatical embedding of the query, before piping the content sequence to the retrieval engine, in order to better specify the query – using logical connectors between key sequences, adding constraining criteria for the search, clarifying anaphora etc.

3. Spoken Query Grammar

In practice, users address their queries using spontaneous speech, and introduce the content sequence with a formulation embedding, which has much less variability in its vocabulary, syntax and semantics. It typically conveys meta-information on the queried content, like logical connectors, search restrictions, search context, etc. (see Table 1). It is thus perfectly doable to develop a grammar specifically for these formulations, putting aside the analysis of content sequences, which are treated as blind constituents.

3.1. Samples of Queries

The grammar has been written from an analysis of typical queries provided by the BBC (cf. Table 1). These queries are exchanged by telephone between journalists searching information, and a human operator who has charge of making research in the documentary base of the BBC programs.

Have you got anything on... ?
We're working on..., what do you have on ?
I am doing a report on..., can you help me ?
I want a VHS on...
Can you send me tapes/printouts on... ?

Table 1: Typical enquiry formulations

From the most frequent query examples, we observed it was possible to decompose the queries in two relatively independent parts and thus, to consider a specific processing for each of them. Indeed, a query can be split into an addressing form corresponding to the query formulation, and a queried content, also called keyword sequence. When the query formulation is recognised, it is possible to ignore the queried content, and then lo-

¹<ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/data/Abbot-Demo.tar.gz>

² see <http://www.research.att.com/sw/tools/>

cate its position in the query in order to send it as a request to the documentary base.

3.2. Grammar Formalism

The formalism used for the THISL grammar is expressed as a DCG (Definite Clause Grammar), with a context-free backbone and unification feature structures (some rules are listed in Table 2).

```
sentence => yesno(q, bse).
yesno(A, B) => vp(A, B), pp.
vp(A, B) => verbal_core(A, B), np2(A).
verbal_core(A, B) => aux_p(A), verb(_, B).
```

Table 2: DCG rules

3.3. Content Sequences

The indexed data in the documentary base are broadcast news. So, a content sequence can refer to many points of present or past actuality. It is not doable to describe the syntactic structures of the content sequences in the query grammar.

Therefore, the content sequences are not analysed but treated as blind constituents. Indeed, we have defined specific syntactic and lexical rules. These rules are under-specified rules, since they accept any lexeme (no syntactic control), but they are limited (no recursion) and precisely located in the query grammar. These specific rules are located in the grammar rules, where the content sequence is expected. We choose to limit the content sequence rule to five items from the observation of the query corpus given by the BBC.

4. Word-Graph Parsing

4.1. Existing Approaches

The word-graph parsing and search techniques presented here are in part inspired by the work of G. Van Noord and his team [18]. Our work can be related more generally to various proposals to decode word-graphs (or recognition hypothesis stacks) using grammar and language model [17, 5]. The novelty of the approach presented here lies in the way we clearly decompose the analysis between grammar and language model, thus leveling the requirements that fall on the grammar – which is in effect easier to develop, maintain and parse – and avoiding to trigger an exhaustive graph search with the trigram model.

4.2. Parsing Characteristics

The core-parser is a classical DCG parser, it leads a top-down analysis with backtrack and parses the word-graph from left to right [3, 4]. It goes from the top goal – e.g. category *S* – down to rules that lead to that goal, as their right hand side categories become new goals to be completed. The lexical realisation of pre-terminal goals are matched to the current position in the string, and backtrack takes over at the end of a goal.

The way it applies to a graph structure is the following; when a goal is stated to start at a given state, each edge of that state – also called word transition – are tried in turn, in a depth-first way, to see if they match the goal lexical prefix and if they can lead a path in the graph to goal completion. We added to the parser the Chart Parsing technique which aims at keeping structures built during the analysis and reusing them when it is necessary, rather than to rebuild them at each alternative analysis (that is, when the parser backtracks).

When the graph is parsed, it remains to filter, among the found paths, those which correspond best to the uttered query. We will see later that the language model is used to pick out the best analyses by filtering the content sequences.

4.3. Full Analysis

The graph is first searched for fully grammatical path, this can be seen as intersecting lexical predictions of the top category *S* with the word-graph paths of word transitions. The parsing goal is that *S* start at the initial state of the graph and end at the final state of the graph. All full analyses found in the graph are retrieved as potentially correct recognition, the content sequences is covered by a <KW> category, as illustrated in Table 3. In general, concurrent complete analyses can be filtered out by simple pragmatic rules, and do not differ much in their application-oriented semantics.

```
CAN YOU FIND A VIDEO ON <KW> </s>
CAN YOU FIND THAT VIDEO ON <KW> </s>
CAN YOU FIND THE VIDEO ON <KW> </s>
CAN WE FIND THE VIDEO ON <KW> </s>
```

Table 3: Full analysis example

4.4. Partial Analysis

It is sometimes the case that the correct sentence is not in the lattice or that the utterance is not grammatical. So, it is necessary to display a making good strategy for the grammatical analysis, by allowing the partial analysis construction. This is the same parser, which is in charge of leading both complete and partial analyses. We have simply added to the parser some mechanisms allowing to re-launch the analysis anywhere in the graph and to combine these partial analyses together (Figure 1 shows some of the built partial analyses).

Modifications have also been made to the grammar in order to permit partial constructions anywhere in the sequence.

Blocking state The full analysis fails when unable to find a valid grammatical transition allowing to continue the parsing. In this case, the analysis is stopped by *blocking states*. We call *blocking state* a state which is the extreme bound of a grammatical edge, but from which no grammatical edge exists given the current grammatical rule (see Figure 2).

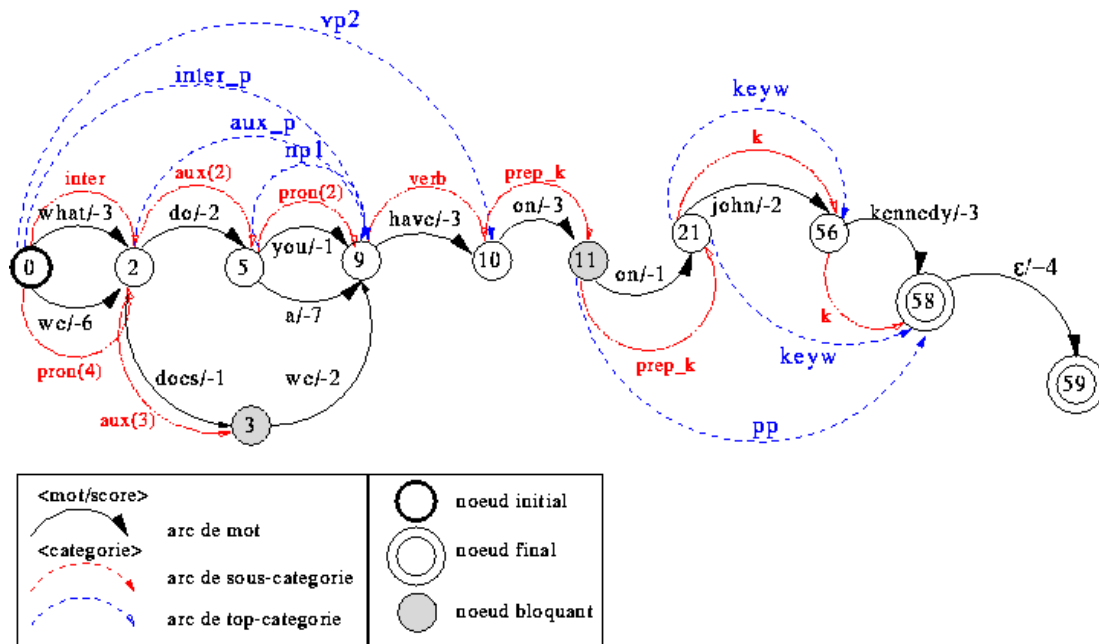


Figure 1: Graph annotated with some grammatical edges, case of partial analysis. The best path is “What do you have on on John Kennedy”, with a grammatical span of 8 words, 2 top category and an acoustic score of -22

First, we determine the set of blocking states. Then, we re-launch the analysis from each blocking state by using the specific partial analysis rules defined in the grammar (in Figure 1, we re-launch the analysis from the nodes 3 and 11). These two stages are repeated until the graph has been fully parsed, that is, until we reach a final state. This parsing do not give the exhaustive set of partial analyses because we only restart them from extreme bounds of grammatical path and not from all the nodes. But we do not loose relevant grammatical solutions, in particular thanks to the redundancy of the word-graph. It is sufficiently connected to guaranty that no correct path is overlooked that way.

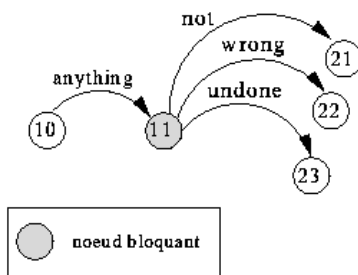


Figure 2: Creation of “unknown” skip

When the parsing is stuck in a blocking state, skip edges, called «*unknown*», are created to its next connected states in the word-graph (cf. Figure 2). Then, the analysis is re-launch from the nodes pointed by the «*unknown*» edges.

5. Word-Graph Search

To combine the partial analyses together and to select the best one, we had to define a set of relevant criteria which determine the score given to each construction.

5.1. Criteria for Scoring

So, each built edge has been associated to a weight which takes into account several factors. We seek:

- to maximise the length of grammatical constituents, in order to favour the most grammatically structured path;
- to minimise the number of built categories to favour a more wide constituent rather than a set of small ones;
- finally we maximise the accumulated acoustic score of the path.

To find the best paths efficiently, we implemented a dynamic programming search, similar to [18]. It consists in keeping for each state e_k the weight associated to the N best known paths from leading to e_k . So, the best paths are found through a score function, which takes the above criteria into account. Table 4 gives the best paths found and the grammatical constituents built into brackets.

| |
|--|
| (WHAT DO YOU HAVE ON)(ON JOHN KENNEDY) [8,2,-22] |
| (WHAT DOES)(WE HAVE ON)(ON JOHN KENNEDY) [8,3,-22] |

Table 4: Partial analyses of Figure 1

5.2. Analysis Ranking

After having retrieved the best grammatical paths we rank them by Language Model log-likelihood (see examples in Table 5 and Table 6). This filtering stage has been led with the domain LM i.e. LM used for recognition. It offers a way to pick out the most probable content sequences. LM probability is computed using a functionality of NOWAY [14], the stack decoder used in ABBOT.

We proceed in two stages. First, we pick out the 100 best among the grammatical paths. Then, the second stage consists in giving to the domain LM these 100 sequences. The LM has to rank these sequences according to their likelihood.

| | |
|-------------------------------------|-------|
| WHAT DO YOU HAVE ON JOHN KENNEDY: | 15.86 |
| WHAT DO YOU HAVE ON JOHN CANDY: | 16.40 |
| WHAT DO I HAVE ON JOHN KENNEDY: | 16.57 |
| WHAT DO YOU HAVE ON JUNE CANDIDATE: | 17.60 |
| WHAT DO YOU HAVE ON DONE CANDIDATE: | 18.28 |
| WHAT DO YOU HAVE ON JUNK CANDIDATE: | 18.38 |

Table 5: Examples of full analysis ranking

| | |
|---|-------|
| (A)(THEY)(HAVE ANYTHING ON JOHN MAJOR): | 20.22 |
| (A)(THEY)(HAVE ANYTHING ON JOHN MAJOR ARE): | 24.17 |
| (A)(THEY)(HAVE ANYTHING ON ON H. R.): | 24.52 |
| (A)(THEY)(HAVE ANYTHING ON TON MAJOR): | 24.63 |
| (A)(THEY)(HAVE ANYTHING ON DON MAJOR ARE): | 27.25 |

Table 6: Examples of partial analysis ranking

6. Results

To illustrate our approach, we made an experiment to compare results given by the speech recognizer alone and results given by doing a collaboration between the grammatical analysis of the word-graph and the domain LM. We experiment on a corpus of 100 queries: 50 query recorded by a native speaker and 50 others by two non native speakers.

6.1. Test Procedure

We present here results obtained by recording a set of 100 plausible queries, pronounced alternatively by two “naive” speakers (a native and a non native), and by an “expert” non native speaker. We work on lattices produced by ABBOT, driven by a standard Business-News LM, “bn97”, which is a trigram model with a 60k word lexicon.

6.2 Recognition Rate

A first observation, on Table 7 and Table 8, is that the post-processing of word lattice improves the recognition rate of the query formulation, with a 31% decrease in WER (Word Error Rate), as compared to just picking the 1-best hypothesis of ABBOT. It appears, that adding a grammatical module allows to better understand the query, since the LM is not suited for such sequences.

| WER on: | ABBOT | ABBOT+NL |
|-------------------|---------|----------|
| Whole query | 45.76 % | 43.35 % |
| Query formulation | 43.34 % | 29.68 % |
| Content sequence | 51.08 % | 71.27 % |

Table 7: Impact of NL processing on word error rate for the non native speakers

On the other hand, we note that the error rate on content sequence increases. It is due to two reasons: first the NL processing is not suited to the content query,

which we have no information about. Second, we will see in Table 9 that the WER are mainly due to insertion of grammatical words which do not match with any documents and thus do not impair the actual document search.

| WER on: | ABBOT | ABBOT+NL |
|-------------------|---------|----------|
| Whole query | 75.52 % | 67.48 % |
| Query formulation | 77.41 % | 52.84 % |
| Content sequence | 72 % | 88.72 % |

Table 8: Impact of NL processing on word error rate for the native speaker

The improvement of the filtering stage leads by the domain LM on the grammatical paths found, should be able to increase the recognition rate on the content sequences.

6.3. Recall and Precision

We want also to assess the contribution of NL processing to the final task of document retrieval. We thus compare keyword recall and precision, obtained on the significant keywords that should drive the document search. We just present results for non native speakers, because recognised hypotheses are too blurred for the native speaker, to allow a post-processing to make any real difference (the keyword recall is way below 30%, even in the word lattice). This speaker had never been confronted to a recognition system and had furthermore a not so standard accent.

We note in Table 9 that figures are less contrastive than before. We observe the keyword recall is quite the same for both, but the keyword precision gets higher with the post-processing module.

| | ABBOT | ABBOT+NL |
|--------------|---------|----------|
| kw recall | 55.68 % | 52.87 % |
| kw precision | 54.44 % | 60.52 % |

Table 9: Impact of NL on document search for the non native speakers

Table 10 and Table 11 present a different view of Table 9. We have split the keyword recall into three cases: when the full keywords have been retrieved, when just a part of the keywords or no keywords have been retrieved.

| | | ABBOT | | |
|------------|-----------|---------|---------|---------|
| | | Full | Partial | No |
| ABBOT + | Kw recall | 71.43 % | 20.00 % | 25.00 % |
| | NL | 0.00 % | 80.00 % | 25.00 % |
| | | 28.57 % | 0.00 % | 50.00 % |
| | | 100 % | 100 % | 100 % |

Table 10: keyword recall for the non native speakers, case of full analysis

The diagonal of Table 10 and Table 11 presents the cases where ABBOT and NL processing have the same performances. What is above of the diagonal presents the improvement of the NL processing in comparison

with ABBOT alone. What is below of the diagonal presents the degradation of the NL processing.

| | | ABBOT | | |
|------------------|---------|-----------|---------|---------|
| | | Kw recall | Full | Partial |
| ABBOT + NL | Full | 62.50 % | 12.50 % | 0.00 % |
| | Partial | 37.50 % | 62.50 % | 33.33 % |
| | No | 0.00 % | 25.00 % | 66.67 % |
| | | 100 % | 100 % | 100 % |

Table 11: keyword recall for the non native speakers, case of partial analysis

We can note that NL processing gets better results for full analyses rather than partial analyses. In Table 10, in the case where ABBOT had retrieved all the significant keywords, we lose sometimes the keyword sequence. But, in the others cases, we show that the NL processing can improve the keyword recall. In Table 11, the results are quite different. In the case where ABBOT had retrieved all the significant keywords, we lose sometimes a part of the keyword sequence but not the whole sequence. In others cases, NL processing improves the keyword recall again.

7. Conclusion

We presented a reasonably light-weight processing to improve spoken interface to document retrieval, by decoding speech recognition lattices with both a small grammar, for query formulations, and a language model, for content sequences.

The results show that our approach is efficient for the query formulation decoding. It is also encouraging to retrieve the content sequences in the word-graph, but we still need to revise further the content sequence processing. The filtering stage is not sufficient enough at that time, to rank the best solution among the first 3 ones.

References

- [1] Abberley D., Renals S., Cook G., Robinson T. 1998. The THISL spoken document retrieval system. In *TREC-6*.
- [2] Barnett J., Anderson S., Broglio J., Singh M., Hudson R., et Kuo S. 1997. Experiments in spoken queries for document retrieval. In *Eurospeech'97*, volume 3: 1323-1326, Rhodes, Greece, September 1997. ESCA.
- [3] Cherchalli S. 1998. Analyse syntaxique de graphe de mots: application à l'interface vocale de grandes bases documentaires. DEA Report, Paris Nord et Paris XIII University.
- [4] Dowding J., Gawron JM., Appelt D., Bear J., Cherny L., Moore R. et Moran D. 1993. GEMINI : A natural language system for spoken language understanding. In proceedings of the 31st annual meeting of the Association for Computational Linguistics, ACL'93, 54-61.
- [5] Goddeau D., Zue V. 1992. Integrating probabilistic LR-parsing into speech understanding systems. In *ICASSP'92*, volume 1: 181-184, IEEE.
- [6] Gorin A. 1994. On automated language acquisition. *Journal of the Acoustic Society of America*.
- [7] Halber A., Colineau N., Cherchalli S. 1998. Grammar and N-gram Collaboration for Information Retrieval Interface. In *SPECOM'98*, ESCA—ELSNET International Workshop on Speech and Computer, St Petersburg, Russia.
- [8] Halber A., 1998. Grammatical factor and spoken sentence recognition. In *Conference on Text Speech and Dialogue*, TSD'98.
- [9] Hanrieder G. & Görz. 1995. Robust parsing of spoken dialogue using contextual knowledge and recognition probabilities. In *ESCA Tutorial and Research Workshop on Spoken Dialogue Systems*, Denmark.
- [10] Harman D. 1992. User-friendly systems instead of user-friendly front ends. *Journal of the American Society for Information Science*, 43:164-174.
- [11] Hochberg M., Cook G., Renals S., Robinson A., Schechtman R. 1995. The 1994 ABBOT hybrid connectionist-HMM large-vocabulary recognition system. In Morgan Kauffman, editor, *Spoken Language Systems*, 170-175. ARPA.
- [12] Lamel L., Bennacef S., Bonneau-Maynard H., Rosset S. et Gauvain J-L. 1995. Recent developments in spoken language systems for information retrieval. In *VIGSO'95*, Denmark.
- [13] Meteer M. et Rohlicek R. 1994. Integrated techniques for phrase extraction from speech. In *Human Language Technology Workshop*, 228-233.
- [14] Mohri M. 1997. Finite state transducers in language and speech processing. In *Computational Linguistics*, 23(2): 269-311.
- [15] Renals S., Hochberg M., 1995. Decoder technology for connectionist large vocabulary speech recognition. Technical Report CS-95-17, Sheffield University, September 1995.
- [16] Roussel D. & Halber A. 1997. Filtering errors and repairing linguistics anomalies for spoken dialogue systems. In *Workshop on Spoken Dialogue Systems*, Madrid, Spain, 11-12 July 1997. ACL/EACL, 74-81.
- [17] Schmid L. 1994. Parsing word graphs using a linguistic grammar and a statistical language model. In *ICASSP'94*, volume 2, 41-44, IEEE.
- [18] Van Noord G., Bouma G., Koeling R., Nederhof M-J. 1997. Robust grammatical analysis for spoken dialogue systems. In *Natural Language Engineering*, 1(1): 1-47.