ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Ian Ruthven (Ed)

Miro'95

Proceedings of the Final Workshop on Multimedia Information Retrieval
(Miro '95)
Glasgow, Scotland
18-20 September 1995

Paper:

A General Approach to Speech Recognition

Christoph Gerber

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

A General Approach to Speech Recognition

Christoph Gerber

Department of Computing Science, University of Glasgow,
Glasgow G12 8QQ, UK

Abstract

The aim is to build a speech based query interface to a text retrieval system. The interface will be independent of the underlying Information Retrieval engine. First steps in this project will be presented in this paper. A study of the properties of typical Information Retrieval queries has shown that the speech interface must effectively address requirements like speaker independence, fluently spoken speech, and uncontrolled vocabulary. These requirements can only be approximated with the currently available speech recognition techniques. Furthermore, such requirements demand a multidisciplinary approach from areas like signal processing, pattern recognition, and linguistics. In order to coordinate these approaches and to be open for further developments in these areas, a speech recognition architecture has been introduced and this will also be presented.

1 Introduction

Speech is the most natural and convenient way for humans to exchange information. In contrast to typing, speech communication requires no special skills.

The aim of speech recognition (SR) is to create machines that can receive spoken information and respond appropriately to that information. Such machines approximate human communication. Written language is derived from speech and is like speech itself human culture. Today, most text documents are created and stored electronically therefore hand writing has been replaced by typing but both requires the knowledge of a written language. From the view of Computing Scientists, writing can be regarded as a special technique for storing information. If an ideal speech-to-text translator would exist then a written language would lose its attractiveness to some extent. Anyway, such an ideal system is not available yet. So the more philosophical question of the meaning of written language need not to be answered now. Electronic text documents will have their attractiveness furthermore. Modern Information Retrieval (IR) tools like NRT [1] are used to find certain information in large text document bases. To access the information, queries are formulated by users. Up to now, queries have been formulated by typing a message, related to the users' information need. Retrieval strategies, described in [2,3] are used to extract query related information from the document base. The question is why typing when speaking is so convenient. It is clear that this results from the limits of the current SR technology.

Current speech recognition systems are not ideal compared to a human's capability in understanding speech. A lot of problems are not solved yet. The main problem of speech recognition lies in the variability of speech. A message can be pronounced quite differently by varying the speaking rate, the individual sounds, the speaker, the gender of the speaker. This variability enables human to understand a variety of different pronunciations of a given language whereas it results in uncertainty which must be handled by the speech recognition system. However at the end of eighties two efficient approaches, (1) the statistical approach by using Hidden Markov Models (HMM), and (2) the connectionist approach by using Neural Networks were established. These techniques make modest and constrained speech understanding applications feasible. At present such constrained applications work satisfactorily in a limited context. If the limited context is natural then the constraints of the SR system are transparent to the user. In that way that the user not necessarily recognises the weakness of the SR system. Where do the constraints come from? The constraints for an SR system come in general from the characteristics of the input language which must be processed.

A Spoken Query Recogniser (SQR) which constitutes a speech interface to a text-based IR system, is a constrained application which works in a natural limited context. The constraints for the SQR come from the properties of the queries:

- Queries are normally short only a few words. This is a benefit in favour of space and time complexity of applied SR algorithms.
- The core of queries are content bearing words. These words contain often more than one syllable. Therefore they are in general better to recognise, e.g., as showed in [4].
- Stop words like prepositions are the hardest to recognise. They are not considered in IR. Lee et al. [5] reported that stop words were responsible for about 50 % of the recognition errors on the SPHINX SR system.
- Finally, a query spoken by the user to a system is generally better formulated in contrast to colloquial speech.

Thus, the impact of uncertainty in speech recognition can be minimised by exploiting the above restrictions. On the other hand, there are also disadvantages from the characteristics of the queries:

- Queries constitute not necessarily a grammatically correct expression. Typically, they are sequences of loosely coupled words. So, syntactical information above the word level cannot be used to reduce uncertainty.
- The index vocabulary of practicable IR applications is uncontrolled and can reach the size of the human mental lexicon. So, the vocabulary of the speech recognition system must be uncontrolled too.
- Typical IR applications are not restricted to an individual single user. Queries are formulated by many different users. That means the SR system must be speaker independent.
- Queries are spoken fluently. That means the SR system must be able to process continuous speech.

Now, our long-term goal is to develop an SQR which exploits not only the above mentioned constraints but also which addresses an unlimited vocabulary as well as speaker independence and continuously spoken speech. An unlimited vocabulary means that the vocabulary used by the IR application should be a subset of the lexicon of the SQR. It is known, that this goal can only be approximated with the current available techniques but to be open for further developments in SR and to be independent from a special technique, a framework will be used that gives the project a great flexibility and has some other interesting properties.

The paper is organised as follows: In the second section the basic strategy for the currently used acoustic front-end is described. In the third section, the realisation of the acoustic front-end is pointed out. The fourth section demonstrates an application of the acoustic front-end in regard to Information Retrieval. The fifth section describes the framework.

2 Strategy

The acoustic front-end the lowest level of the SQR is realised by using Hidden Markov Models. This decision has been made for the following reasons: (1) The number of successful realised SR prototypes with HMMs. (2) HMMs are more simple to analyse mathematically than neural nets. (3) The availability of a toolkit called HTK [6] which is a collection of basic algorithms necessary for HMM training and recognition.

An HMM can be regarded as a knowledge base for the variability of a speech sound. This variability can be modeled by estimating the HMM parameters from a corpus of speech data. The HMM parameters are a set of transitions modelling the time flow, and a set of states with an attached probability distribution, modelling the spectral features of a speech sound. The HMM maps a segment of a speech waveform to a corresponding linguistic unit, e.g., to a phoneme, syllable, word or a sentence. It represents then an instance of a linguistic unit. The question is which linguistic unit should be used? This point is still highly discussed in speech research. Even in the very up-to-date study [7], recommending the feasibility of the big SR project verbmobil, is no evidence given that the proper speech unit has been found: 'Despite the great amount of work that has been done on speech recognition, especially over the past decade, we believe that it is still far from clear what the size of the segment should be that a system should aim to recognize'.

The HMM has to be trained from existing speech. For each instance of a linguistic unit a huge amount of corresponding speech must be provided. This training problem restricts the choice to a unit smaller than a word because it is impossible to provide enough training material for each word. In addition, we want not to be fixed to a certain vocabulary size a priori. Words should be generated just by concatenating smaller units. Ideally would be the phoneme. There are merely about 50 of them, words can be built just by concatenating phonemes according to a phonemic described lexicon. So a lot of training examples could be provided but their disadvantage is that they are highly sensitive to the so called coarticulation effect. That is adjacent phonemes influence each other, so actually to each phoneme exists a set of different variations dependent from the context in which the phonemes occur. However this context dependency can be modeled by the individual phoneme HMM to some extent.

The decision has been made to use phonemes as the basic linguistic unit which will be modeled by the HMM. The output of the HMM front-end will be a sequence of phonemes. As stated above, a query can be regarded as a sequence of loosely coupled words, which are not necessarily syntactically related to each other. Therefore the goal is to extract words from the phoneme sequence. It is obvious that the recognised phoneme sequence will be noisy according to the limitations of the model dealing with the uncertainty of speech. The question is how many noise can be admitted. To address this problem the recognition task is regarded from a view of the Coding Theory. In general if a message without redundancy is transmitted then possible errors in the message cannot be corrected. However, if a redundant message is transmitted, then possible errors can be corrected or at least detected. Languages are very redundant. According to Shannon [8] has written English roughly a redundancy of 50 %. This means that 50 % of a message is enough to decode the content of a message. It is plausible that this is also true to some extent for a phonemic described language. But the following question arises: What is the redundancy

Figure 1: Redundancy of the phonemic code transcribing the English language

of a language? That is human's knowledge about language which can be subdivided in knowledge about syntax¹, semantics and pragmatics. Thus errors could be reconstructed by using human's knowledge about language.

A simple experiment was done which might give a sense of the redundancy which results from the syntax of a language. In a 100,000 word phonemic described lexicon, all words with 6 phonemes were picked out. 6 phonemes corresponds to the average length of words. Then in every word one phoneme was substituted by another phoneme and this over all possibilities. The consequence was that words became equal. Figure 1 shows how many words became equal for each phoneme pair. The maximum was the replacement of phoneme 'ih' by 'ah' which produced 500 equal words. The average was 11 equal words. In most cases of replacements, however, no equal words were produced which you can see in the diagram. The good result that relative less words became equal is because of the enormous gap of possible words with 6 phonemes (that are 6^{48}) and the actually used words in English that are about 20,000.

The question is not whether we should use human's knowledge rather than how can we apply it. The acoustic front-end which maps the physical representation of speech to linguistic units, is the interface to the machine's cognition system. So further steps to spot finally words must be based upon the linguistic units from the acoustic front-end. It can be regarded as the core of the SQR. This acoustic-front end will be described in the next section.

3 Realisation of the Acoustic Front-End

Based on the decision to use phoneme HMMs, a suitable speech database had to be found for training the HMMs. The TIMIT [9] database has been selected because it is especially designed for speaker independent, continuous speech phoneme recognition. It contains speech from 630 speakers, representing 8 major dialect regions of American English. Totally there are 6300 distinct sentences included. 6300 sentences may imply that it is a huge amount of speech data for training but measured in time, there are only 6 hours speech. This is rather a modest figure imagining what we have to listen to learn our language. As phonemic alphabet serves the set of 48 phonemes used by Lee and Hon [10].

¹Syntax means in this context knowledge about composing of higher level linguistic units from lower level linguistic units.

Figure 2: Applied Hidden Markov Model structures

Before the HMMs can be trained, it must be decided which HMM structure will be used and what kind of spectral features of the speech signal will be modeled by the HMM. Not the raw speech waveform is modeled rather a set of spectral features which results from an analysis of the speech signal. The so called Mel-frequency Cepstral Coefficients [11], their first and second order derivatives and an speech signal energy coefficient are used. It has been shown in [12,13] that this choice of parameters works well. These parameters are used to describe 30 ms of the speech signal. So far, no special recommendation for the ideal structure of a phoneme HMM can be found in the relevant SR literature. A suitable structure must be found by experiment. Five alternative structures have been used for experimentation. The recognition result will help to decide which of them will be used further. The tested HMM structures are sketched in Figure 2. The time and space complexity increases from model 1 to model 5. The left-to-right character of the HMMs seems to be a reasonable choice considering the time flow of phonemes. Intuitively one might expect that the more complex models represent the phoneme better because there are more skip transitions which might reflect the different duration of the individual phonemes better. Table 1 shows the evaluation of the five models by using the recognition rate as a measure evaluated on the TIMIT database. The given recognition rate of the individual models is relative to model 1. Surprisingly, the simplest model seems to be the best. Even if the individual figures could not be taken so seriously there is at least no clear discrepancy which would favour a particular model. As mentioned, the time and space complexity of model 1 is the lowest, hence model 1 will be used for further work.

Model number	1	2	3	4	5
Rel. recognition rate (%)	100	99.49	99.90	99.6	97.4

Table 1: Relative recognition rate of different phoneme HMM structures

The TIMIT speech data has been used for training as well as for the evaluation of the phoneme recognition rate. In general, the more training speech the better will be the approximation of the HMM to the corresponding phoneme. A better approximation can also be achieved by applying more sophisticated HMM structures but that requires again more training data. So the evaluation of the different HMM structures is also a function of the amount of training data and thus the data in Table 1 is only true for a particular training configuration.

80 % of the speech of the TIMIT data has been used for training, and the remaining 20 % for recognition. Speakers occurring in the training set do not occur in the test set. Both training and recognition have been done by using HTK. The training task maximises the probability of generating a phoneme by applying the Baum-Welsh algorithm to the corresponding HMM. The recognition task finds the most likely HMM sequence that is the most likely spoken phoneme sequence by applying the Viterbi algorithm [14]. Not only the probability of the individual HMMs is considered in recognition but also the probability how the individual HMMs are connected. This probability reflects a priori knowledge of the language. If each HMM can follow equally likely each other HMM then the used a priori knowledge is zero. Another alternative is the use of the co-occurrence probability of two phonemes, which is called the bigram language model. Both types have been used for evaluation. The bigram probability has been estimated over the whole TIMIT corpus. Table 2 shows the recognition result in detail. The test set contained a total of 50754 phonemes.

	No language model	Bigram language model
Correct	59.85 % (58.77 %)	62.04 % (64.07 %)
Substitutions	32.40 %	30.74%
Deletions	7.74 %	6.85%
Insertions	12.33 %	11.39 %

Table 2: Phoneme recognition rate of the acoustic front-end

This result is comparable with other evaluations using the same database. The recognition rate reported by Lee and Hon [10] is stated in brackets. However, they used a reduced set of 39 phonemes. Robinson and Fallside [15] reported a recognition rate of 66.7 % with bigram model as well as a reduced set of 39 phonemes but they used the connectionist approach. The HMMs may have not reached their maximum recognition accuracy yet. However the comparison shows that the basic decisions must not have been wrong. The substitution error rate is high but with respect to the redundancy stated in Figure 1, some of these errors might be corrected by using higher level knowledge. Even the incorporating of the phoneme co-occurrence statistic improved the result clearly.

The acoustic front-end is the set of 48 trained phoneme HMMs. A simple approach could be just to use the bigram language model and to map the recognised phoneme sequence to words. Such an approach is stated in the next section.

4 The String Matching Experiment

A simple approach to finally receive words would be just to apply a string matching algorithm which matches the recognised phoneme sequence against a phonemic transcribed dictionary. The acoustic front-end of section 3 with the bigram language model is the basis for this experiment. It provides only language knowledge on the phoneme level but it is on the other side the most flexible one because there is practically no vocabulary limit. It is obvious that the recognised phoneme sequences are always noisy because of the limited recognition performance. Therefore the string matching algorithm must be error tolerant to some extent. There are plenty of string match algorithms which can manage that. Figure 3 shows the flow chart of the string matching experiment. It works as follows. Words are requested to be spoken in isolation. The surrounding silence is cut off by an endpoint detection algorithm. The acoustic front-end provides the corresponding phoneme sequence. The length of the recognised phoneme sequence is determined, that is the number of phonemes. Then all phonemic transcriptions of the words in the dictionary which have the length m of the recognised phoneme sequence $\pm u$ phonemes tolerance are matched against the recognised phoneme sequence by the string matching algorithm. The output is a list of the N -best matched words. A dynamic programming (DP) algorithm described in [16] has been applied for string matching. The algorithm is based on the Levenshtein metric and allows string comparisons with k differences. A string x has a distance k to a string y if string x can be converted into string y by a sequence of k substitutions (S), insertions (I) or deletions (D) of a character. An other class of string matching algorithms are algorithms that allow k mismatches which means that only substitution errors would be considered. These algorithms would be not accurate enough for phoneme recognition because of the high insertion and deletion errors, see Table 2. The string comparison is carried out between the recognised phoneme sequence and the reference phoneme sequences from the dictionary. The reference phoneme sequence with the smallest distance k to the recognised phoneme sequence is the match. The following error weighting function has been used for the DP algorithm,

$$w(S, I, D) = 21 \times S + 17 \times I + 15 \times D$$

where the weights reflect the individual errors of Table 2. The advantage for having such a weighted string matching result

Figure 3: Flow chart of the string matching experiment

is that it allows the ranking of the N-best sequences. Such a ranking scheme has been applied so that the N-best sequences are considered as output result.

The evaluation has been done only by a few speakers, however, in a real natural office environment. Speakers' voice was not included in the training data (speaker independent evaluation). Such an evaluation is quite time expensive. Therefore only a few examples have been tested but they show the trend. The applied dictionary had a size of 6300 words including word inflections. The dictionary contains small words like prepositions as well as long and content-bearing words. The tolerance u of the phoneme sequence has been set to 2. The number of test examples is too less to calculate and represent reliable figures. However, the evaluation has shown that this direct approach works after all for long words which have more syllables than two or two stressed syllables. It does hardly work for short words. For example, the word 'university' was in 90 % of the cases under the 15 best matched words, 'information' 70 %, 'administration' 65 %, 'meanwhile' 50 %, 'taxi' 80 %, 'car' 10 %, and 'the' 0 %. One reason that it does not work for short words is because of the ratio between the number of errors and the word length. This ratio is in general smaller for long words than for short words. The consequence is that errors in small words result in much more possible alternative words. An experiment has been done which emphasises this fact.

In a dictionary of 110,000 phonemic transcribed words all possible substitution errors of the kind 'substituting one phoneme by another' have been applied. The consequence was that words became equal. For example substituting the phoneme

Figure 4: Histogram of normalised number of new generated words due to substitution errors with respect to the word length

'hh' by 'm' of the word 'hh ou s e' (house) causes a new valid word 'm ou s e' mouse. All these new valid words were counted and the number of the new produced words has been recorded with respect to the length of the words measured in phonemes. However, it must be considered that words of different length have different occurrence frequencies. Thus, to obtain more realistic results the number of new produced words is normalised by the occurrence frequency of the words. Figure 4 shows the result. It can clearly be seen that based upon substitution errors for long words much more less new words were produced than for short words. Thus, long words have a much better chance to be recognised than shorter ones! It must be admitted that recognition with string matching carried out by this experiment is in general poor. Nevertheless, it was expected that hardly anything will be recognised therefore the result caused a good impression. It must be considered that a poor language model was used, the dictionary size was not so small, and the evaluation has been done in a real environment with all the disadvantages like computer noise. It could be argued that this experiment works only with single words but the extension to speaking word sequences merely with the constraint of speaking with a pause of 200 ms between the words is only a technical matter. Such pauses can be detected and then the sequence of words can be splitted and treated with the means of the stated isolated word recogniser.

Assuming that the recognition rate would be more robust, this approach would be appropriate as speech interface to an text-based Information Retrieval system. Because the list of the N-best recognised words fits in the concept of modern Information Retrieval systems. At first, the stemming process of IR systems reduces often the N-best list considerably. Table 3 shows the 15 best recognised words and their stemmed forms for the spoken word 'university.' The porter algorithm [17] has been used for stemming. The stemmed form 'univers' appears three times which can be regarded as a reduction of the list. At second, the N-best list supplies weights for the recognised words. These weights can directly be passed to the IR system as term weights. The transfer of these weights is plausible. At third, the assessment of the retrieval result by the user provides relevance feedback information which helps implicitly to locate the correct words.

Rank	Recognised word	Weight	Stemmed word
1	university	44	univers
2	unanimity	54	unanim
3	curiosity	58	curios
4	universality	58	univers
5	university-wide	61	univers wide
6	complicity	64	complic
7	analyticity	65	analyt
8	unrealistic	68	unrealist
9	eccentricity	68	eccentr
10	electricity	71	electr
11	explicitly	71	explicitli
12	consistently	72	consist
13	possibility	74	possibl
14	uni-directional	74	uni direc
15	instability	75	instabl

Table 3: Impact of stemming on the 15-best word list for the spoken word 'university'

Furthermore, tools like word-sense-disambiguators and semantic-cooccurrence filters can additionally be applied to reduce the uncertainty of the N-best word list. So, the IR system could help to reduce the uncertainty of the recogniser by its inherent properties in a way that it is transparent to the user. However, the main drawback of this kind of speech interfaces is the case when the spoken query words are not represented in the N-best list. It happens when the spoken words are not in the dictionary or when the recognition performance is in general too poor. The recognition performance of this direct approach alone is too weak. Although, the fact that it works to some extent for long words suggests that further research should be invested.

If this approach would work satisfactorily for all words then the incorporating of further knowledge sources would not be necessary! Although human's enormous language knowledge which is intensively used in speech understanding implies

that for an automatic SR system a more complex approach is needed. An improved version of the direct string matching approach could be applied as an individual speech recognition module in a combined speech recognition system where several modules contribute their strength and compensate among each other their weaknesses. So the general approach is to encode words by incorporating human's knowledge about language in a modular way. It requires a practicable organisation that allows using of multidisciplinary resources of which we are not necessarily aware at moment. The overall result could be that the recognition accuracy of the phonemes is despite of incorporating several knowledge sources to low and that the decision of using phonemes as the linguistic unit was wrong. In the near future it could prove to be that the neural network approach is much better. That is why we are looking for a flexible way which (1) does not fix us to a special technology, and which (2) allows us that individual employed techniques can be exchanged easily against better ones without affecting other employed techniques. A way which give us these properties is stated in the next section.

5 The Relation & Representation Space

In speech recognition, several approaches exist for solving the task. Some of them come from the pattern recognition side like statistical, syntactical, and neural network approaches. The techniques provided by these approaches can be characterised as algorithms solving problems with mostly a polynomial time complexity. Their space complexity is finite thus they can be processed by machines. Given an algorithm f applied to an input x , an output y will be produced as a 'relation' of f and x . Other support comes from the linguistic side like phonology, morphology, syntax, semantics, and pragmatics. A lot of this linguistic knowledge cannot always be expressed in algorithms which can be solved in a practicable time and space complexity. But those which can be processed by a machine can be characterised as follows: Given a knowledge k applied to an input x , an output y will be produced as a relation of knowledge k and x . That knowledge is practically human's knowledge about language. Pattern recognition and linguistic solutions are involved in each another. They are just different point of views. An algorithm represents a prior knowledge, and some knowledge can be implied to an algorithm but not necessarily for all human's knowledge exists algorithms. Both strategies have in common that the input and output must be any 'unit' with a certain 'interpretation'.

So several techniques exist to translate human speech in text. At the moment it cannot be decided or guessed which particular technique is the best to do the recognition task. It cannot even be claimed that these are the ultimate techniques. There will be to expect completely new techniques. In particular the application of linguistic knowledge is still a challenge. However what seems to be clear is that not one particular approach will solve the recognition problem, at least not for high requirements like continuous speech, speaker independent, and uncontrolled vocabulary. Therefore in the future, several approaches from the pattern recognition side as well as from the linguistic side have to be combined to solve the problem in common. The main challenge will then be to combine the different techniques in a well defined manner. The techniques will now be called 'relations' to emphasise their character as operator and position between input and output units. The relations need not to be mutually exclusive merely of the sequential schedule of up to date used 'von Neumann' computing architectures. If it makes sense, relations should work in parallel as well as sequential. The relations have in common that they generate, based upon a given input, a given output. However it is obvious that they use not necessarily all the same input as well as generate not the same output. That means not the same units. For instance, the input of one relation could be phonemes and it generates words. The input of another could be sentences and it produces syllables.

So the input and the output of a set of relations is polymorphic. In general the relations work together and all are interested in doing the best to solve the recognition task. Therefore they have to communicate with each other. Communication can be done reasonably if they communicate only via their collected input and produced output. Thus a communication between relation f and relation g can be regarded as that relation f uses the output of relation g as input or vice versa. The communication protocol between the relations is defined by the input and output units themselves. Units must be unique for a clear communication. To represents the output of algorithms, units must be able to express a partial knowledge state. Conclusion, for a speech recognition system an architecture is required which makes it possible:

1. To accommodate units which (1) must be unique and which (2) must be able to allow ideally all (relations) SR algorithms to express their contributions.
2. To integrate a set of different relations operating with different input and output units.
3. To add additional new relations (perhaps later) which influence not the processing of existing relations.

To fulfill requirement 1, a suitable description for a unit must be found. The following point of view should help. An ideal speech-to-text system would translate a spoken sentence to a sequence of letters. This sequence can be interpreted as a 'sentence'. Additionally it must be identified by an arbitrary identifier to distinguish it from other sentences or to identify its content. An arbitrary identifier could be used but its sequence of letters is already a suitable identifier. The same sentence can be spoken and translated at a later 'time'. Distinguishing this sentence from formerly spoken can be done just by using the start or end

time of translation. The sentence is composed of expressions which are interpreted as words. A word is identified by its sequence of letters too and has also been translated to a certain time. A word is composed of phonemes. A phoneme has also been translated to a certain time and is also identified by a symbol. So symbols are concatenated to a sequence of symbols to a certain time with a certain interpretation. This fact holds not only for linguistic units also other physical units will have its interpretation and symbolic identification as well as a certain time of activation. Conclusion, it seems sensible that units must consist at least of the three-tuple (interpretation, time, symbol). To fulfill requirement 2, a space of units must be provided on which the relations can operate and interact. To fulfill requirement 3, the space of units and the relations must be independent and also the relations themselves must be independent of each another.

This can be achieved by creating a 'relation space' (RLS) and an 'representation space' (RPS) with a relation independent structure. The relations need not know anything about each another. This excludes not that relations assume a particular state which was generated by other relations. New relations can demand new interpretations thus the number of interpretations must not a priori be determined. In harmony with that, the following formal definition is given for the representation space:

Representation space: $\{ \text{Interpretations} \times \text{Times} \times \text{Symbols} \times \text{Confidences} \}$

Confidences = $\{ \text{confidence} \mid \text{confidence} \in \mathbb{R} \}$

Symbols = $\{ \text{symbol} \mid \text{symbol} \in \Sigma^* \}$

Times = $\{ \text{time} \mid \text{time} \in \mathbb{N}, \text{time} \geq 0 \}$

Interpretations = $\{ \text{interpretation} \mid \text{interpretation} \in \Sigma^* \}$

$\Sigma = \{ A, \dots, Z, a, \dots, z \}, \Sigma^* \rightarrow \mathbb{N}$

Σ^* denotes all strings which can be generated over the alphabet according to the algebra of strings given, e.g., in [18]. The RPS can be regarded as embedded in the euclidean vector space of $\mathbb{R}^4$. The coordinates are interpretation, time, symbol, and confidence. A unit is represented as a point 'P(interpretation, time, symbol, confidence)' in $\mathbb{R}^4$. So, the metric of the vector space can be used by the relations. For instance, the euclidean distance between two points can be interpreted by the relations. An 'interpretation' states the meaning of a symbol. For example, an interpretation called 'phoneme' categorises the symbols which have the meaning of phonemes. Obviously, an interpretation will be used to categorise linguistic units but there is no restriction to that. An interpretation can also be regarded as the set name of the symbols whereas the 'symbols' themselves are the elements of the set. For instance, the symbol 'ih' can be used to identify the phoneme 'ih', or a symbol 'house' can be used to identify the word 'house'. Interpretation and symbol can be named by any string contained in the language Σ^* . A 'time' states the end time of a symbol relative to the beginning of a 'valid sequence' (definition of valid sequence follows). So the difference between the time of symbol and its successor symbol is the duration of the symbol. Now a sequence can be defined: A 'valid sequence' is a time ordered sequence of symbols on a certain interpretation with no time gap which starts at time zero and ends when no further symbol which is an element of Σ^* follows. Thus each interpretation has its valid sequence. For example, on the interpretation 'syllable' each syllable has to be a real syllable according to some universal language rules but two successive syllables need not to be a valid word. A 'confidence' states the quality of a symbol conditioned on a given interpretation and time. It allows to consider potential alternative symbols. If a symbol x has a higher confidence value than a symbol y then symbol x is regarded as the better hypotheses. A confidence is uniquely defined as the likelihood,

$$\text{confidence} = -\log P(\text{symbol} \mid \text{time}, \text{interpretation})$$

where this definition is obligatory for all relations.

So far, the representation space provides a communication protocol for a broad class of algorithms. The next step is to fill the RPS by introducing relations. One relation from which will be started is the acoustic front-end described in section 3. Another relation could be the string matching module described in section 4. All relations together make up the relation space. The main question will be the sensible arrangement of the relations. The ordering of the relations will play a major role. A similar framework has been realised in the mid 70's by Reddy et al. [19]. They used the so called 'blackboard architecture'. This approach was mainly focused on combining the domain of linguistic knowledge sources by using a common data structure called a blackboard. They used an agenda-based top-level control module to trigger their functions. The blackboard approach fell into disfavor in comparison to the faster, and simpler control strategies of the more loosely coupled models [20]. The lack of computing power at that time was another drawback. In this framework, just the order of the relations is used which can either be sequential or parallel as a sort of inherent fixed control structure. The relation space will then just look like an electrical resistor network where the resistors are the counterparts of the relations and the representation space provides the conductivity. Relations present a kind of static prior knowledge. Several connected relations constitute again a prior knowledge, or a relation itself can be subdivided in an atomic set of relations. Conclusion, it seems to be plausible to use a fixed structure of relations.

Figure 5: Network of relations corresponding to a linear system

The result is a static network of relations. This network can be treated as a ‘Linear System’ with all its properties. The four-tuple $\langle \text{interpretation, time, symbol, confidence} \rangle$ has then the character of a discrete signal where the confidence is the superimposed parameter as a function of the three independent variables: $\text{confidence} = f(\text{interpretation, time, symbol})$. Thus, linear algebra can be applied for data fusion. Figure 5 shows an example of a relation network.

It is interesting to note that these kind of structures are not unknown in the cognitive science community. In cognitive science, a structure like the RPS is called a ‘physical symbol system’. Newell and Simon [21] claim with their hypotheses about physical symbol systems that such systems have the necessary and sufficient means for general intelligent actions. The representation space and the relation network reflects also to some extent Chomsky’s standpoint stated in [22]. He assumes that the mind can be regarded as a modular structure of partial systems (relations). This mental structure bears the grammatical competence (relation network) which is a set of rules (relations). The task of the mental structure (representation space) is to relate and generate certain mental representations like sounds and meanings.

Even if Chomsky’s, Newell’s and Simon’s view may not be shared by everyone and will always be discussed, we must keep in mind that for speech recognition the only example is the human, and there is no doubt that this is still a mysterious example. These views make this example accessible in a way that it can be imitated by a machine. For instance, Chomsky’s modular structure (relation space) addresses the proven way to manage complex software problems: ‘divide and conquer’. The physical symbol system (representation space) constitutes the basis for a universal machine from which practicable computers can be derived.

6 Conclusion and Future Work

In this paper the problem domain of a speech interface for text retrieval has been stated. The final goal is a Spoken Query Recogniser for a text-based IR system. Despite of some advantages resulting from the properties of IR, there is not much difference to a continuous speech recognition system with uncontrolled vocabulary. The main advantage is that queries are short and that stop words need not to be recognised correctly. The basic decision has been made to use an HMM based acoustic-front end with phonemes as subword units. The so far realised acoustic front-end and the current recognition performance is the starting point for further steps. The goal of the SQR will be to encode words out of the noisy phoneme sequence. The string matching experiment has been applied as short-term approach. It has pointed out its weaknesses and thus the demand for more complex encoding steps. However, the fact that based on a poor language model long words are recognised is quite promising and will therefore further be investigated. Even if this approach could be improved considerably, the recognition performance will be too weak alone. In general plenty of additional knowledge and algorithms must be applied to the noisy phoneme sequence. A framework has been introduced which allows the incorporating of various knowledge sources and algorithms from different domains.

The current challenge is the definition of as many relations as necessary that finally a sequence of loosely coupled words with a set of possible alternative words will be the output of the SQR. This sequence will be the top layer of the representation space which will then be passed to the IR system. For each word candidate a set of possible words is allowed to use more

credit from the properties of the IR system in case if not all uncertainty of the speech utterance could have been removed. So some uncertainty will be transferred to the text query. The relevance feedback power of the IR system will then be used to resolve the remaining uncertainty of the query.

7 Acknowledgement

The author would like to thank the Glasgow IR group and especially Mark Sanderson for the many and interesting discussions on the topic of the paper.

8 References

1. Sanderson, M., van Rijsbergen, C.J. NRT (News Retrieval Tool), Electronic. Publishing: Origination, Dissemination and Design, 4.4, 1991, pp. 205-217
2. van Rijsbergen, C. Information Retrieval. Butterworths. London, 2. edition, 1979
3. Salton, G., McGill M. J. Introduction to Modern Information Retrieval. McGraw-Hill Inc., New York, 1983
4. Wilcox, L.D., Bush, M.A. HMM-Based Wordspotting for Voice Editing and Indexing. Proceedings of the European Conference on Speech Communication and Technology, vol. 1, Genoa, September 1991, pp 25-28
5. Lee, K.-F., Hon, H.-W., Reddy, D. R. An overview of the SPHINX speech recognition system. IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 38, January 1990, pp. 35-45
6. Young, S.J., Woodland, P.C., Byrne, W.J. HTK: Hidden Markov Model Toolkit V1.5. Cambridge University Engineering Department Speech Group and Entropic Research Laboratories Inc., September 1993
7. Kay, M., Gawron M. J., Norvig P. Verbmobil: A Translation System for Face-to-Face Dialog. CSLI, Lecture notes, No. 33, Leland Stanford Junior University, USA, 1994
8. Shannon, C. E. Prediction and Entropy of Printed English. Bell System Technical Journal, 30, January 1951, pp. 50-64
9. Garofolo S. J., Lamel F. L., Fisher W. M., Fiscus J. G., Pallett, S. D., Dahlgren, L. N. DARPA TIMIT: Acoustic-Phonetic Continuous Speech Corpus CD-ROM NISTIR 4930. National Institute of Standards and Technology (NIST) Gaithersburg USA, February 1993
10. Lee, K.-F., Hon, H.-W. Speaker-Independent Phone Recognition Using Hidden Markov Models. IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 37, November 1989, pp. 1641-1648
11. Davis, S. B., Mermelstein, P. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 28, August 1980, pp. 357-366
12. Rabiner, L. R., Wilpon, J. G., Soong, F.K. High Performance connected digit recognition using hidden Markov models. IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 37, August 1989, pp. 1214-1225
13. Rabiner L. R., Pan K. C., Soong, F. K. On the performance of isolated word speech recognizers using vector quantization and temporal energy contours. AT&T Technical Journal, 63, September 1984, pp. 1245-1260
14. Rabiner, L. R. Tutorial on Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE. vol. 77, February 1989, pp. 257-286
15. Robinson, T., Fallside, F. Phoneme Recognition from the TIMIT database using Recurrent Error Propagation Networks. Technical report CUED/F_INFENG/TR.42, Cambridge University Engineering Department, Cambridge, England, March 1990
16. Sankoff, D., Kruskal, J.B. Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley Publishing Company, USA, pp. 18-21, 1983
17. Porter, M. F. An algorithm for suffix stripping. Program, vol. 14, 1980, pp. 130-137
18. Hopcraft, J., Almoign, J. Introduction to Automata Theory, Languages and Computation. Addison-Wesley, USA 1979

19. Lesser, R. V., Fennell R. D., Erman, L.D., Reddy R.D. Organization of the Hearsay II Speech Understanding System. IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 23, February 1975, pp. 11-24
20. Waibel, A., Lee, K.F. Readings in Speech Recognition. CA:Morgan Kaufman Publishers, Inc., San Mateo, 1990
21. Newell, A., Simon H. A. Computer Science as Empirical Inquiry: Symbols and Search. Communications of the Association for Computing Machinery, vol. 19, 1976, pp 113-126
22. Chomsky, N. Rules and Representations. Columbia University Press, USA 1980