THE LINGUISTIC PROCESSOR IN A MULTI-LINGUAL TEXT-TO-SPEECH AND SPEECH-TO-TEXT CONVERSION SYSTEM

L. Boves (*,%) & M. Refice (#,%)

ABSTRACT

This paper summarizes the work being undertaken in ESPRIT project #860, 'Linguistic Analysis of European Languages'. The major objectives of this project are to provide statistics on the frequency of use of words, n-graphs, n-phones, and ambiguities (both in spelling and in pronunciation) as well as to develop a language model that can help in resolving ambiguities and that can perform a number of additional tasks in office automation and in speech technology. The language model will be developed and all attendant information will be provided in a unified way for a large number of European languages. In this presentation attention will be focussed on the language model.

INTRODUCTION

This report describes the work in the EEC-ESPRIT project 'Linguistic Analysis of European Languages', that started in the late fall of 1984. At that time the consortium that was to carry out the work comprised six partners, viz. Ing. C. Olivetti Technological Research (Torino, Italy) as prime contractor, and Acorn Computers Ltd. (Cambridge, U. K.), Centro Studi Applicazioni in Tecnologie Avanzate (CSATA, Bari, Italy), Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingenieur (LIMSI, Orsay, France), Ruhr Universität Bochum (F.R.G.), and Katholieke Universiteit Nijmegen (The Netherlands) as partners. Six languages were to be studied, viz. Italian, English, Spanish, French, German, and Dutch, the study of the Spanish language being assigned to CSATA. In the spring of 1986 the consortium was extended by two more partners, viz. Universidad Nacional de Educacion a Distancia (UNED, Madrid, Spain) and University of Patras (Greece). The extension of the consortium had two main effects: The Greek language was included, and the role of CSATA changed, because the study of Spanish was made the responsibility of UNED. CSATA is now charged with the development and testing of large parts of the consortium-wide software.

Initially, the project was funded for three years; accordingly, its goals were somewhat limited. After the extension of the consortium the total duration of the project was increased to four years, which allowed the formulation of slightly more ambitious goals.

THE OBJECTIVES OF THE PROJECT

The main objective of the project is to provide a language independent software environment for dealing with the linguistic phase of a number of applications in the realm of office automation such as high quality, natural sounding text-to-speech conversion for unlimited vocabularies, automatic speech recognition for large vocabularies, omni-font optical character reading including

(*) Institute of Phonetics, Nijmegen University, P.O. Box 9103, 6500 HD Nijmegen, The Netherlands
(#) CSATA, 173 Via Amendola, 70126 Bari, Italy
(%) This paper is presented on behalf of the complete consortium. Space limitations prohibit to mention the names of all contributors.
automatic reading of handwriting. A number of sub-goals were also defined, including the development of compatible systems for grapheme-to-phoneme and phoneme-to-grapheme conversion, the computation of statistics on ambiguities (both homophones and homographs), the creation of syntactically tagged text corpora, and the generation of statistical information on words, n-graphs, n-phones, etc. Space limitations prevent us from providing detailed information on most of these sub-goals.

The global description of the objectives of the project deserves some comments and explanations. First of all, the software system to be developed should be maximally language independent. Thus, in the description of the individual languages the highest conceivable level of standardisation should be pursued and the software system to be developed should resemble a shell that can be filled with the (standardised) description of knowledge about a certain language, rather than a self-contained and 'finished' product. Consequently, considerable effort has been invested in matters of standardisation and in the definition of the overall architecture of the target system.

Secondly, the restriction to the linguistic phase of the intended applications means a restriction to processing on the symbolic level. Work in the present project will not cross the boundary between discrete linguistic symbols and continuous processes in some physical domain. In terms of speech applications this means that the synthesiser in a text-to-speech and the acoustic pre-processor in an ASR system will not be considered. One further limitation is that, for the time being, only isolated words will be considered in ASR. Thus, the notoriously difficult segmentation problem will not be handled.

Thirdly, from the very start of the work it was realised that even a seemingly unambiguous task like grapheme-to-phoneme conversion in speech synthesis cannot be accomplished without using at least some syntactic knowledge. Thus, a decision on the type of linguistic model to be used had to be made at an early stage of the project. It was decided to aim at a probabilistic positional grammar (a Markov-type grammar) based on transition probabilities of pairs and triplets of syntactic categories, to be implemented in such a way that syntactic knowledge that cannot be expressed in the form of transitional probabilities can be added in a natural way if it appears to be necessary.

The use of Markov-type models immediately incurs the necessity of defining training texts. It was decided to start out with a training corpus of approximately 100,000 words, to be chosen in such a way that the corpora for all languages were as similar as possible and of the type of language likely to be encountered in an office environment. The closest approximation to this ideal was the choice of texts from the official EEC publications that are available in all languages of the community (except Spanish at the start of the project). In a later stage of the project the corpus was extended by the addition of another 300,000 words of EEC publications plus 100,000 words of newspaper text. The processing of the newspaper text has two aims: It allows some feeling to be obtained for the degree to which the Markov model depends on the origin and type of training text and it allows a closer comparison of the results for the Spanish language with those of the remaining languages, because for Spanish the initial corpus is based on 100,000 words taken from the journal El Pais, due to the lack of Spanish versions of the EEC material.

STANDARDISATION

In order to achieve international cooperation in a complicated domain like linguistic processing several standards had to be agreed upon. One of the first was the choice of an operating system and programming language. VAX-VMS was chosen as the operating system. In choosing a common programming language AI-languages like Lisp and Prolog were not considered as attractive options, because it was estimated that the computational requirements incurred by these languages would be inconvenient if not completely prohibitive. Therefore, the ESPRIT recommendations for the use of programming languages left the consortium with the choice between Pascal and 'C'. A detailed comparison of these languages led to an eventual preference for 'C'. Given the shell-like character of most of the software it is guaranteed that the linguists working in the project will not be bothered with limitations of a procedural language like 'C' for expressing linguistic facts. The working environment created in 'C' offers convenient means for expressing linguistic knowledge, without much of the computational burden of the traditional AI languages.
A second aspect where standardisation was essential is the use of computer codes for a phonetic alphabet. The common alphabet should at least be compatible with the coding systems used by the partners in their pre-existing systems for grapheme-to-phoneme conversion. Thus, an inventory had to be made of all phonemic elements used in the partner's systems. Most of the partner's systems appeared to generate an output that is not strictly phonemic: apart from phonemes proper a number of allophones are generated, especially where the allophones result from exceptionless phonological (assimilation) rules. Next, a notational formalism had to be designed. It was decided that the basic phoneme codes would consist of single ASCII characters, chosen so as to maximise the similarity with the IPA symbols. Where phonemes are in an open-closed opposition, the lower case characters are reserved for the closed and the upper case characters for the open member of the pairs. Length has to be marked by a '·' following the segment, in conformance with the IPA convention. In the standardised Computer Phonetic Alphabet (CPA) diphthongs are represented by the code for the phoneme most closely resembling the initial part of the diphthong, followed by a '/' as a marker for diphthongisation. The same convention is used to distinguish affricates from their cognate fricatives. Nasal vowels are coded by a '...' following the vowel symbol. In total the CPA comprises some 80 codes, including symbols for pauses and word stress.

Standardisation was also essential for the coding of syntactic classes. For the purposes envisaged a simple coding scheme consisting of no more than the major word classes like 'verb', 'noun', etc. is demonstrably inadequate. Thus, a far more detailed coding scheme is necessary, sufficiently general and powerful to be applicable to all languages under study. Eventually, a coding scheme was developed in which each code comprises seven fields. The first field contains the main word class. The second field, which is used for all regular classes, is used for the major subclass. Some of these subclasses for the main class 'Noun' include 'common', 'proper name', 'acronym', and 'number'. The remaining subclasses are used to code aspects like tense/mood, person/number, case, gender, and specific inflectional forms. In specifying codes, fields that are considered irrelevant at that moment or for the purpose at hand can be assigned a dot '.', meaning that the subclass in question can be specified, but is not; alternatively, the field can be assigned a dash '-', meaning that all possible codes in that position check. In addition, a powerful means for defining so-called cover symbols is provided. As its name implies, a cover symbol stands for a set of individual classes.

Additional standards that had to be designed concerned matters like program documentation, information exchange, formatting and storage of raw texts, processed texts, lexica, statistics etc. These standards cannot be dealt with in any detail here.

**THE TRAINING SYSTEM**

It is well known that any Markov model has to be initialised or trained. Essentially, this training consists of building a number of data structures. The first structure is a lexicon of all possible words that may occur, with their attending probability of occurrence. The second structure is formed by one or more matrices describing the probability of occurrence of pairs, triplets, quadruplets, etc. of items from the dictionary. The Markov model we are working on is based on a lexicon containing a large number of word forms, together with a comprehensive account of their legal word classes, and two- and three-dimensional matrices of transition probabilities between word classes. Clearly, the probabilities specified in the matrices depend on the choice of syntactic categories along the dimensions. One of the research topics for the moment is to determine the optimal level of detail in the codes for each dimension.

In order to initialise the lexicon and the transition matrices one has to have access to (preferably large) amounts of texts in which the words are tagged with word classes. This introduces the need for tools that automate the tagging process as far as possible. Research is under way to build a bootstrapping or self-learning system that, in interaction with a linguist, uses existing matrices and lexica in order to propose tags for new texts and that subsequently uses the newly tagged texts to update the lexica and matrices. The major concern in the development of a self-learning tagging system is how to guarantee that no formally legal but yet incorrect tags will pass unnoticed. Spin-offs of the training process are statistics on the frequency of use of words and word
classes, statistics on the proportion of words in a text covered by the n most frequent words in a language, etc. These statistics have been computed for the first training corpora and will be subsequently updated when larger corpora have been processed.

THE MATRIX EDITOR EMMA

In order to assist linguists in their task to define the optimal sets of word classes a tool has been built for the manipulation of transition matrices. This tool, called an Editor for Matrices from Markov Analyses, or EMMA, enables the user to build, inspect, change, and update two- and three-dimensional transition matrices. In order to be sufficiently fast in an interactive environment a data structure had to be designed that allows an efficient storage and retrieval of data in extremely large matrices. It is estimated that in the eventual systems between 150 and 200 word classes will be needed along any of the dimensions of the two- and three-dimensional transition matrices. In the case of the threedimensional matrix less than 1% of the entries in the 200 x 200 x 200 (worst case) matrix are expected to be different from 0. Such a matrix is far too large to be stored in main memory using the straightforward storage techniques. Yet, in order to be useful in on-line parsing, the entries of the matrix must allow rapid access, either by cell if all three indices are given, or by row/column if any two out of three indices are known. Thanks to the sparseness of the matrix a storage scheme based on triply linked lists could be employed that combines reasonably compact storage with reasonably fast access. A set-theoretic formalism has been defined for specifying so called Cover Symbols, i.e., sets of word classes. The formalism allows concise hierarchical definitions of Cover Symbols to be used. Each Cover Symbol definition is automatically checked for consistency. Matrices manipulated with EMMA need not be square or cubic; any dimension may contain an arbitrary number of different symbols. EMMA can build new matrices from tagged texts or from calculations on existing matrices, for instance by using newly defined (sets of) cover symbols.

THE OVERALL ARCHITECTURE OF THE LINGUISTIC PROCESSOR

Although Markovian grammars have proved to be extremely powerful in a variety of applications, it is not expected that a Markov model will be able to deal with all details a comprehensive linguistic processor has to handle. In particular, there will be cases where a Markovian parser may leave someone with an ambiguous output in which yet only one alternative is completely legal. This alternative would be picked by a system utilising the relevant syntactic and/or semantic knowledge that cannot (easily) be expressed in the form of transition probabilities. One possible alternative, is to express the knowledge contained in the transition matrices in the form of rules that allow one to be more specific, for example by including information on lexical items that must be present instead of mere word classes (Ref. 1). However, not all relevant linguistic knowledge can be expressed in the formalism of a probabilistic positional grammar. In order to be able to exploit this syntactic (and in a later stage perhaps also semantic) knowledge an architecture is under development that will combine a Markovian parser with modules containing (heuristic) syntactical rules into one efficient processor. This processor will be organised around a dynamic blackboard as its central data and control structure.

REFERENCES