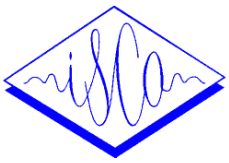

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

WORD CLASS ASSIGNMENT IN A TEXT-TO-SPEECH SYSTEM

Rijk Willemse, Leon Gulikers

Nijmegen University,
P.O. Box 9103, 6500 HD Nijmegen, The Netherlands

Introduction[1]

Word Class Assignment (WCLA) in a Text-To-Speech (TTS) system is necessary for correct grapheme-to-phoneme conversion, and as a precursor for syntactic analysis (necessary for the calculation of intonation contours). Since the words of a language do not form a closed set (especially in languages like German and Dutch that can freely form new compounds), a lexicon will always be incomplete and thus must be supplemented by modules for morphological decomposition, and rule-based WCLA. Moreover, most words can have several word classes; thus, WCLA requires statistical and/or syntactical evaluation of a word class lattice.

We have developed and implemented a modular and language-independent architecture for assigning syntactic classes to words in real-time that consists of three parts. The first part assigns all possible word classes to the input words, based on a 14 k words lexicon, deterministic morphological decomposition, and rule-based WCLA. The second part reduces the number of words in the word class lattice by means of a Viterbi Search. The third part consists of a so-called Wild Card Parser (WCP), a deterministic parser that assigns phrase structure to input sentences.

We have trained the probabilistic part of our system on a training corpus of 590 k tagged words. The performance of the system — which is still under development and, therefore, open to improvement — has been tested on a 10 k word corpus of Dutch texts from weekly newsmagazines. It appeared that the system can handle any sentence (which has been pre-processed in order to expand abbreviations, to translate numbers to sequences of words and to detect names) in real-time on a 386 PC. After WCLA, 95 % of the words have their correct syntactic category. After the operation of the WCP, 92 % of the words have their correct syntactic category. However, all words for which pronunciation depend on word class were given the correct class. The resulting phrase structure is a viable basis for computing neutral prosody.

The word class system used consists of relatively coarse-grained syntactic categories like "noun" (singular and plural), "verb" (singular, plural, infinitive and past participle), "determiner", "adjective", "pronoun".

Lexicon Look-Up

It has been decided to perform only straightforward look-up techniques for words and compounds (c.f. [1]). Additional, form-based morphologically oriented techniques for determination of word classes of words that could not be found in the lexicon are applied in the module for WCLA. Since the words found in the lexicon almost always possess more than one syntactic category, a word class lattice is built that must be reduced as much as possible by the modules that are applied after Lexicon Look-Up, in order to obtain one word class per word.

In the TTS system as we have designed it, the lexicon for words contains the following information:

- the graphemic representation of the word
- the phonemic representation of the word
- the word classes that can be assigned to the word
- the (relative) frequency of the word-word class pair
- the position(s) of stress within the word

Word Class Assignment

The WCLA module of our text to speech (TTS) system takes a possibly incomplete word class lattice as input: i.e. a sequence of words with a set of zero or more syntactic categories — word classes — associated with each word. Most of the words are found in the lexicon, and possess one or more word classes and a probability value. The output of WCLA consists of a complete word class lattice: i.e. a sequence of words with a set of one or more classes associated with each word. Basically, WCLA consists of two parts:

- a set of rules that generate word classes for the input words that have not yet been assigned word classes
- an algorithm that reduces the number of word classes that have been assigned to the input words.

This implies that, generally spoken, during the process of word class assignment the number of word classes will first increase and will later decrease.

WCLA by rule

There are several types of rules for generating word classes. The rules of the WCLA module can be used to assign word classes to words that did not receive word classes during Lexicon Look-Up. The rules are based on form (morphological rules), position in the sentence (context rules), look-up (lexical rules that address lists of words or lists of parts of words), and on a combination of these aspects.

When all rules have been applied to all words of the sentence that could not be found by Lexicon Look-Up, the

[1] The work reported here was carried out as a part of the ESPRIT-project POLYGLOT

10.21437/ICSLP.1992-29

chance exists that the set of word classes corresponding to some words of the input sentence is still empty. For that reason, a default rule is applied when all other rules have been tried. This rule assigns a set of word classes to words that do not possess a word class.

Depending on the status of a rule, two things can happen when it applies:

- when the rule is non-deterministic, all other rules are applied to this word, possibly causing more word classes to be assigned to the current word
- when the rule is deterministic, all other rules are skipped, and the following word of the sentence that was not found in the lexicon is considered

The formalism used to express this linguistic knowledge in the WCLA module is as follows:

$$IF\ conditions\ THEN\ assignment \quad (1)$$

The assignment part of a rule will only be executed when the conditional part of the rule succeeds in its entirety: in this case one word class will be added to the set of word classes corresponding to the current word, provided that it does not occur in this set already. Assigning a word class to a word can be performed in a deterministic way and in a non-deterministic way. An example of a non-deterministic assignment is:

$$CATEGORY(category) \quad (2)$$

Here, the category *category* is assigned to the current word, and the other rules are evaluated. An example of a deterministic word class assignment is:

$$CATEGORY\_U(category) \quad (3)$$

In this case, the category *category* is assigned to the current word, and the other rules that would have been applied, are skipped.

The conditional part of a rule may involve the use of the logical operators AND or OR to combine simple conditions to complex ones. Eight functions have been implemented to formulate conditions. An additional function, $LENGTH$, returns an integer that indicates the length of the word that it takes as an argument. For that reason a condition may also consist of the function $LENGTH$ followed by a relational operator (one of $\leq$, $\geq$, $<$, $>$, $==$, $!=$), and an integer. Furthermore, three additional functions for accessing (parts of) strings have been implemented.

Moreover, an arbitrary number of lists of words or of parts of words can be defined. In the rules, the name of the list can be used to address the contents of such a list. It is possible to compose lists of words, morphemes, or of other relevant parts of words. The formalism for WCLA by rule allows any linguist to easily formulate rules for any language without having to code this knowledge in a programming language. WCLA is set up in a language-independent way: all linguistic knowledge is kept strictly separated from the program.

Reducing the word class lattice

The word class lattice produced by WCLA, can be modelled by means of a transition probability matrix and an emission probability matrix. The words of a sentence correspond to an ordered sequence of word classes. Since, in general, a word can correspond to several word classes: the possibility exists that a sequence of words corresponds to more than one sequence of classes, as is the case in the well-known example "Time flies like an arrow".

For each word of the sentence, one word class can be said to be the most likely to correspond to that word. The likelihood of a word class at word t depends (a.o.) on the previous word class, at word $t-1$, and on the joint probability of the word-word class combination.

A transition probability matrix for word classes is used to determine the likelihood of moving from one word class to another word class. An emission probability matrix is used to determine the joint probability of a word in combination with a certain word class.

Initially, the transition probability matrix and the emission probability matrix are estimated on the basis of a relatively large tagged corpus: i.e. a training text consisting of word-word class pairs. The transition probabilities of the word classes, a_{ij}, are obtained according to Equation 4.

$$a_{ij} = \frac{Number\ of\ transitions\ c_i, c_j}{Number\ of\ transitions\ from\ c_i} \quad (4)$$

The estimation of the emission probability of a word-word class combination is obtained according to Equation 5.

$$b_j(w_t) = \frac{Number\ of\ pairs\ c_j, w_t}{Number\ of\ classes\ c_j} \quad (5)$$

Finding the most likely sequence of word classes in a word class lattice — i.e. finding the most likely sequence of word classes corresponding to a sequence of words — is done by means of Viterbi Search. Since the possibility exists that the sequence of states contains errors — due to insufficient training of the model, and due to long distance syntactic phenomena [2] — Viterbi Search as we implemented it produces a set of highly probable sequences of word classes in the form of a word class lattice.

Syntactic Analysis

The word class lattice produced by WCLA is processed by the module for syntactic analysis: the WCP parses the input word class lattice and selects a set of prosodically relevant constituents from the analysis that is produced for each sentence. Furthermore, the WCP selects word classes from the lattice: when a word class is used in an analysis (when it is not analysed as a Wild Card), all other word classes in the ordered set to which it belongs are discarded.

WCP is a new application of context free grammars: it detects the set of maximum expansions of syntactic constituents — described in a context free grammar — that

covers the maximum part of the input sentences, and it selects a set of word classes from the input word class lattice.

WCP provides one parse for each input sentence. In some cases, a parse contains Wild Cards, i.e. words that cannot be parsed as belonging to a constituent. This enables WCP to deal with unrestricted text, even if it contains non–grammatical constructions. Moreover, WCP provides a way around structural ambiguity. Where backtrack parsers tend to produce more than one analysis for a sentence, WCP provides only one parse: the first set of context free rules of the grammar that succeeds, is applied. By ordering the alternatives of the context free rules, preferences as to the application of the rules can be expressed. In this respect, the parser can be compared to a probabilistic parser after supervised training: the most likely sequence of rules is applied. Extension of the WCP to such a parser will not be difficult, therefore.

A WCP grammar is an ordinary context free grammar which has been tuned to describe sentences in a relatively superficial way, only producing structural information on syntactic categories without going into details as to their semantic structure. The application of the context free rules is subject to three conditions, however:

- the rules apply deterministically
- a special WCP search strategy is used during parsing
- when no analysis can be made according to the context free rules, a minimum number of Wild Cards is allowed in the parse.

The alternatives of the context free rules are evaluated from top to bottom: when an alternative of a rule succeeds, the remaining ones are skipped. Ordering the alternatives appears to be a very powerful means to express linguistic intuitions. It should be realized, of course, that the use of a deterministic parser will, by necessity, result in occasional errors where ambiguity exists as to neighbouring syntactic categories.

The heart of a WCP grammar is the so–called *'s_cat* rule'. The syntactic constituents that the parser looks for, are defined in the grammar by this rule. The Dutch *s_cat* rule is represented in Figure 1.

```
s_cat   :   sentence;
            verb_phrase;
            sub_ordinate_sentence;
            noun_phrase.
```

Figure 1: The Dutch rule for detection of constituents

By means of WCP, a search is performed for the set of maximum expansions of the syntactic category *s_cat* that covers the maximum part of the input. For the Dutch grammar this implies that the parser searches, in the first place, for a maximum expansion of the constituent *sentence*. When no *sentence* — as defined in the grammar — can be found in the word class lattice representing the input sentence, the parser looks for a maximum expansion of the constituent *verb_phrase*. And when that fails, the parser looks for a maximum expansion of the next constituent, until all alternatives of the *s_cat* rule are exhausted. When all alternatives of the *s_cat* rule have been tried and still no constituent is found that covers the complete input sentence, the search strategy used by the WCP system looks for combinations of maximum length of the *s_cat* constituents according to the preferences expressed in the *s_cat* rule, eventually combining *s_cat* constituents with a minimum number of Wild Cards.

The current WCP grammar for Dutch consists of 40 syntactic rules: i.e. a total of 124 alternatives. The number of word classes described in the Dutch WCP grammar equals 21.

Experiment

A corpus derived from various types of texts (newspapers, novels and popular weekly magazines), consisting of 590 k words, i.e. consisting of 382 different word–word class pairs, in 32 k sentences was used to train the transition probability matrix and the emission probability matrix of the WCLA. This corpus will be referred to as the training corpus. Each word of the training corpus was tagged with one word class from a word class system of 21 syntactic categories. Punctuation marks were assumed to act as words also.

The training corpus was used to create a three–dimensional word class transition probability matrix. It was also used for the estimation of the emission probability matrix: all word–word class pairs in the training corpus were counted. (However, for the emission probability matrix, all words that did not occur in the 14 k words lexicon of our system were discarded, since the emission probability matrix was coded in the lexicon.)

A 10 k word corpus of Dutch texts from weekly newsmagazines (the test corpus, hereafter), a part of the Eindhoven Corpus, consisting of 434 sentences, was tagged with our class system of 21 syntactic categories. Punctuation marks were assumed to act as words also. The test corpus was not included in the training corpus. The texts contained in this corpus can be characterized as rather complex: the sentences are longer than in ordinary newspapers and popular weekly magazines, the syntactic constructions and the words are more diverse.

The words of the test corpus were submitted to a provisional algorithm for text pre–processing (TPP), to Lexicon Look–Up, to WCLA by rule, followed by Viterbi Search (producing a set of highly probable sequences of word classes in the form of a word class lattice), and finally to the WCP. The result of application of TPP to the test corpus (9950 words) is that 453 words were pre–processed; they received a tag on the basis of a set of pre–processing rules. The result of the subsequent application of Lexicon Look–Up was that 9001 words were found.

Subsequently, the word class lattice was input to WCLA by rule. After that, Viterbi Search was applied, which produced a reduced word class lattice. Six different values for the thresholdfactor that determine the reduction of the word class lattice by Viterbi Search were used

in order to determine an optimum with respect to the test corpus. Reduction of the word class lattice was done according to the following method. For each word class, a probability is computed by multiplying its transition probability with respect to the previous word classes with its emission probability (i.e. the likelihood of this word class corresponding to the current word). At each word, the word class with the best probability was determined, and at each word a threshold was determined as in Equation 6.

$$threshold = max\ prob * thresholdfactor \quad (6)$$

All word classes corresponding to the word with a probability below the threshold were removed from the word class lattice.

Results

The effects of the different values of the thresholdfactor for reducing the word class lattice are represented in Table 1. WCLA was applied to the output of TPP and Lexicon Look-Up. Of the 9950 words, 453 words were assigned word classes by TPP. Of the remaining 9497 words, 9001 words were found in the lexicon. This implies that 496 words were not found by the subsequent application of TPP and Lexicon Look-Up.

Table 1: Results of the application of WCLA to the output of TPP and Lexicon Look-Up. In the first column the value of the thresholdfactor is represented; in the second column, the number of words for which the correct class did not appear in the lattice; in the third column the mean depth of the word class lattice after WCLA is given; the fourth column shows the number of correct word classes in the first position in the lattice. For reference, the fifth column shows the number of words not found by TPP and Lexicon Look-Up.

thr.	WCLA errors	cl/wrd	correct 1st pos.	Input quality
0.9	421	1.19	8501	496
0.5	445	1.17	8510	496
0.2	455	1.16	8512	496
0.1	456	1.16	8516	496
0.01	458	1.16	8516	496
0.001	458	1.16	8516	496

The number of words for which the correct class did not appear in the lattice produced by WCLA increases, while the thresholdfactor and the number of classes per word decrease. Apparently, by removing word classes from the lattice, more errors are introduced. The main reason for this lies in the fact that the WCLA rules that deal with the word classes of neighbouring words are less powerful when the contextual information decreases.

In Table 2 the results of the application of WCP to the output of WCLA are represented: the word class lattice output by WCLA serves as input for WCP. For reference, the results of WCLA at each value of the thresholdfactor are given again.

Table 2: Results of the application of WCP to the output of WCLA. In the first column the value of the thresholdfactor used during WCLA is represented again; in the second column, the erroneous word classes WCLA assigned are given again the third column shows the number of erroneous word classes found by WCP on the basis of the output of WCLA; in the fourth column the number of Wild Cards introduced by WCP that result in a correct word class in the first position, is indicated.

thr.	WCLA errors	WCP errors	correct Wild Cards
0.9	421	793	802
0.5	445	773	819
0.2	455	771	819
0.1	456	766	820
0.01	458	763	823
0.001	458	763	823

Conclusions

The results presented in Table 2 show the number of erroneous and correct word class assignments with respect to the word classes found in the tagged version of the test corpus. Visual inspection of the errors, however, showed that all words for which the pronunciation depends on their word class, were given the correct class.

It seems that a relatively shallow word class lattice produced by WCLA, has a positive effect on the performance of the WCP, which in turn will enhance the prosodic quality of the TTS system [3]. A thresholdfactor of 0.01 (and less) produces a minimum of WCP errors, while the number of WCLA errors is slightly off minimum. Therefore it seems that a thresholdfactor of 0.01 is sufficient for relatively good word class assignment.

Finally, the results should be interpreted in the light of the relatively poor quality of the provisional module for text pre-processing: it is expected that great gain can be scored by enhancing the quality of TPP. Almost all 496 words that were not found by TPP and Lexicon Look-Up, are names and idiosyncrasies (like uncommon numbers, dates and abbreviations) causing WCLA and therefore WCP to fail.

References

[1] L. Gulikers and R. Willemse, "A Lexicon for a Text-To-Speech System", in: *Proceedings of the ICSLP 1992*, Banff, 1992.

[2] A. Derouault and B. Merialdo, "Natural Language Modeling for Phoneme-to-Text Transcription", in: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, November 1986, pp. 742-748.

[3] R. Willemse and L. Boves, "Context Free Wild Card Parsing in a text-to-speech system", in: *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Toronto, 1991, pp. 757-760.