Character Stream Parsing of Mixed-lingual Text

Harald Romsdorfer and Beat Pfister

Speech Processing Group
Computer Engineering and Networks Laboratory
ETH Zurich
{romsdorfer,pfister}@tik.ee.ethz.ch

Abstract

In multilingual countries text-to-speech synthesis systems often have to deal with sentences containing inclusions of multiple other languages in form of phrases, words or even parts of words. Such sentences can only be correctly processed using a system that incorporates a mixed-lingual morphological and syntactic analyzer. A prerequisite for such an analyzer is the correct identification of word and sentence boundaries. Traditional text analysis applies to both problems simple heuristic methods within a text preprocessing step. These methods, however, are not reliable enough for analyzing mixed-lingual sentences.

This paper presents a new approach towards word and sentence boundary identification for mixed-lingual sentences that bases upon parsing of character streams. Additionally this approach can also be used for word identification in languages without a designated word boundary symbol like Chinese or Japanese. To date, this mixed-lingual text analysis supports any mixture of English, French, German, Italian and Spanish.

1. Introduction

Mixed-lingual sentences can only be correctly processed by a polyglot text-to-speech (TTS) synthesis system that incorporates a morphological and syntactic analysis of the input text, as e.g. shown in [1, 2, 3, 4]. Such a mixed-lingual morphological and syntactic analyzer yields the syntactic structure of the sentence and the morphological structure of the words including their lexically annotated transcription and language. Thus, identification of the base language of a sentence and of the languages of foreign inclusions is solved inherently by morphological and syntactic analysis.

A prerequisite for such an analyzer is the correct identification of syntactic words. Syntactic words are the terminal elements of syntax analysis. In contrast to orthographic words, that are delimited by blank characters and therefore can easily be identified in text preprocessing, syntactic words are more difficult to identify and do not always correspond to orthographic words due to different graphemic phenomena, like

- *word contractions*, e.g. English ”he’s”, ”Mary’s”, German ”das ist’s” (that’s it) or Italian ”po’d’acqua” (some water),
- word forms spanning multiple orthographic words (so called *multi-word lexemes*), e.g. English ”in fine” (adverb) or French ”est-ce que” (interrogative particle),
- *ambiguous punctuation symbols*, e.g. a period at the end of an abbreviation may also be a full stop to indicate the end of the sentence at the same time, and
- *languages without a designated word separation symbol* like Chinese or Japanese. E.g. [5] gives a good overview of the problems text analysis for Chinese is confronted with.

In this paper we first describe an approach to identify syntactic words as it is implemented in the polyglot TTS synthesis system *polySVOX* of ETH Zurich. We demonstrate that by means of this approach word contractions, multi-word lexemes and sentence ends can be correctly identified even within mixed-lingual contexts. Additionally, we show how this approach can be used to disambiguate words in Chinese texts.

2. Identification of syntactic words

In order to correctly identify syntactic words within a graphemic input text, morphological and syntactic knowledge is necessary. Therefore, it is not reasonable to do this identification in some text preprocessing step. We better integrate identification of syntactic words into morphological and syntactic text analysis. This analysis is realized as a bottom-up chart parser for penalty-extended definite-clause grammars (DCG). An input scanner normalizes the graphemic input text character by character in a stream-like fashion. For this normalized character stream, a contiguous sequence of matching lexemes is looked up in a morpheme lexicon. The chart parser itself operates on three different levels: a word, sentence and paragraph level. Each level is provided with a separate set of grammar rules. Analysis for each level is triggered by the preceding level. Word analysis, finally, is triggered by the input scanner.

Figure 1 illustrates this approach with a morphological and syntactic analysis of the English sentence: ”It’s in St. Mary’s St.”. The correct pronunciation of this sentence [ɪts ɪn sənt meəriz stri:t] requires to identify the first ”St.” as abbreviation of ”Saint” and the second one as abbreviation of ”Street”. This can be achieved by syntactic means, that have to provide the correct analysis of ”It’s” as a personal pronoun followed by a contracted verb form and of ”Mary’s” as possessive form of a noun.

In the following we shortly describe the main processing steps of our text analysis:

Text normalization generates out of the graphemic input text or input stream a well-defined character stream. As we use character tokens instead of word tokens, also punctuation characters, the blank character, carriage return, the newline character and other special characters can be included as separate tokens. Text normalization primarily takes care that all capital letters are converted to lowercase letters, all sequences of contiguous space characters are reduced to one space character and all illegal input

characters are deleted from the character stream. Additionally, a paragraph boundary symbol "<PB>" is inserted at the end of the stream.

Lexicon lookup looks for all possible decompositions of the character stream into the lexemes of the morpheme lexicon. For each matching lexeme, a corresponding edge into the chart. These edges are shown in the "lexicon lookup" section in Figure 1. In the morpheme lexicon the keyword ':WORD_END' indicates a possible word boundary after the respective lexemes, as can be seen in Table 1.

Word analysis is started only at unambiguous word boundaries in order to prevent incorrect analysis results. A chart vertex is an unambiguous word boundary if the associated lexemes of all edges ending in this vertex are tagged by the keyword ':WORD_END', and no edge is crossing this vertex. The character token sequence starting form the previous unambiguous word boundary up to the current one is then parsed for all contiguous sequences of words that are morphologically correct as defined by a word grammar, cf. Table 2. The resulting syntactic word lattices are inserted into the chart. These constituents are shown in the "word analysis" section in Figure 1.

```
PCT_E (f,s)  "."      ""    :WORD_END
PCT_E (f,s)  ". "     ""    :WORD_END
PRGTRM ()    "<PB>"   ""  0 :WORD_END
TRM_E (?)    " "      ""  0 :WORD_END
TRM_E (?)    ""       ""  20
TRM_E (abbr) ""       ""  1
PERSS_E (sg,p3,n,s)    "it"       "'It"
PERSS_E (pl,p1,n,o)    "'s"       "z"
PREPS_E ()             "in"       "'In"
NS_E (ncl1,sgen1,n)    "street+"  "str'i:t+"
NS_E (abbr,nosgen,n)   "st"       "str'i:t+"
NTS_E (ntcl2)          "st"       "s'@nt+"
NPRS_E (ncl1,sgen1,f)  "mary+"    "m'e_@ri+"
NE_E (ncl1,sg)         ""         ""
NE_E (abbr,sg)         ""         ""
NE_E (abbr,sg)         "."        ""
NTE_E (ntcl2)          "."        ""
NTE_E (ntcl2)          ""         ""
NGE_E (sgen1,sg)       "'s"       "z"
AUXBS_E (sg,p3,ind,pres,yes)  "'s"  "z"
AUXHS_E (sg,p3,ind,pres,yes)  "'s"  "z"
```

Table 1: Some entries of the English morpheme lexicon: A lexical entry consists of a constituent name and a set of grammatical features, graphemic and SAMPA-like phonemic representation in double quotes followed by an optional penalty value with a default value of 1. The language of an entry is encoded as suffix of the constituent name, e.g. '_E' indicates an English constituent. The optional keyword ':WORD_END' indicates a possible word boundary.

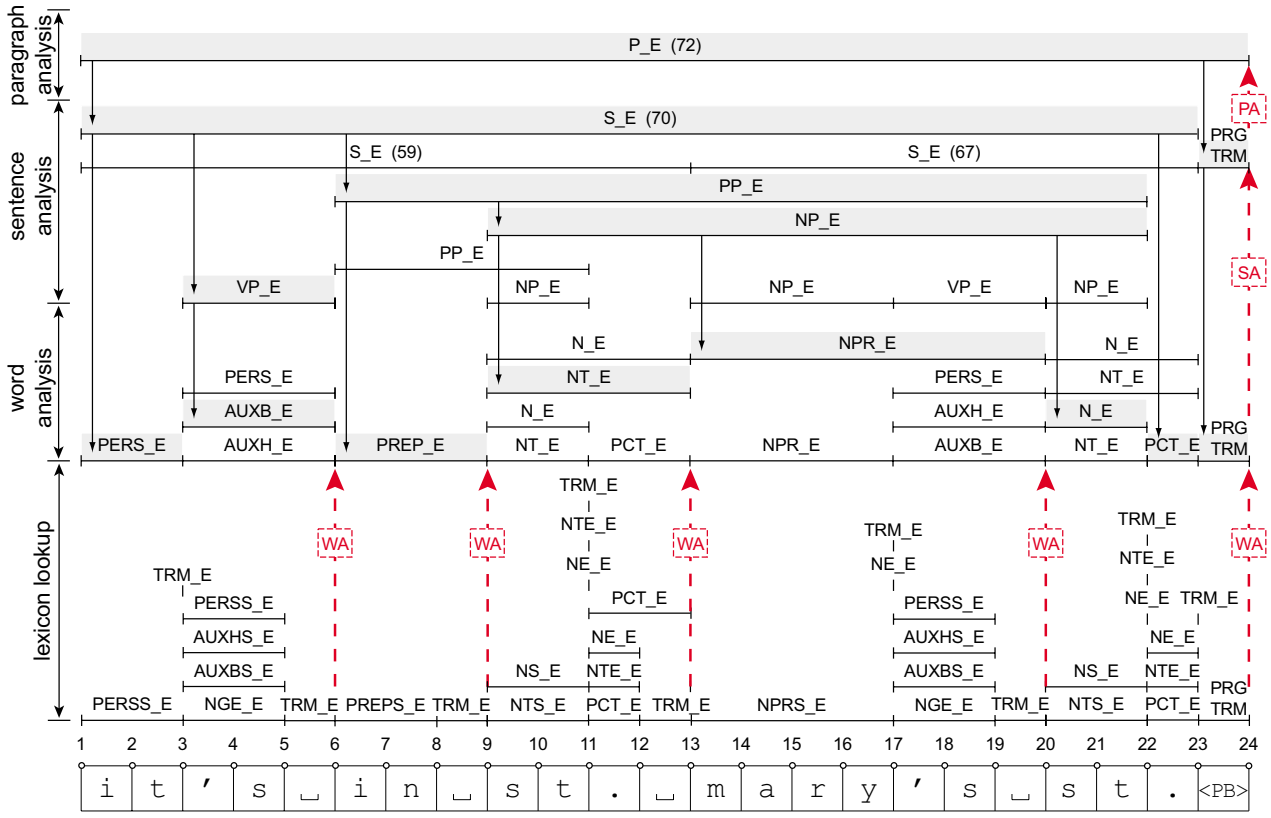

Figure 1: Representation of the simplified chart resulting from morphological and syntactic analysis of the sentence "It's in St. Mary's St.": At the bottom the normalized input character sequence is shown. Edges are drawn without constituent feature values. For a set of edges with the same associated constituent but different feature values that span the same vertices only one edge is shown. The "lexicon lookup" section contains edges associated to the lexemes found during lexicon lookup. The "word analysis", "sentence analysis" and "paragraph analysis" sections contain edges associated to constituents resulting from the respective analysis levels. The minimal penalty values of sentence and paragraph constituents are denoted in parenthesis at their associated edges. Arrows with dashed lines indicate trigger events. The constituents of the final syntactic parse tree are shown with grey background.

Sentence analysis is designed similar to word analysis. Terminal elements are the word constituents of word analysis. Sentence analysis is started only at an unambiguous sentence boundary. This is at the next chart vertex where the associated word constituents of all edges ending in this vertex are tagged by the keyword ':SENT_END' and no edge is crossing this vertex. This keyword is set by word grammar rules, as shown in Table 2. Sentence analysis is needed to disambiguate morphologically ambiguous words. The results of sentence analysis are all possible syntactically correct sequences of sentences, as defined by a sentence grammar. These results are again inserted into the chart as shown in section "sentence analysis".

Paragraph analysis is started at an unambiguous paragraph boundary. This is at the next chart vertex where the associated sentence constituents of all edges ending in this vertex are tagged by the keyword ':PARA_END' and no edge is crossing this vertex. This keyword is set by sentence grammar rules, cf. Table 3. The sentence constituents serve as terminal elements for syntactic analysis of the paragraph. Out of the set of possible sentence sequences, paragraph analysis returns the sentence sequence with minimal total penalty.

2.1. Analysis of contracted word forms

The approach presented here allows to correctly analyze ambiguous contracted word forms. The key idea is to include in morphological analysis beside of blank characters also empty characters as word delimiters. These delimiters are listed as 'TRM_E' in the morpheme lexicon in Table 1 and are used in the word grammar rules in Table 2 to terminate each word constituent. Thus, joint orthographic words can be split into a sequence of syntactic words. In order to prevent incorrect word splits, the empty word delimiter has got a higher penalty, cf. Table 1. Additionally, specific word categories like abbreviations can use separate empty word delimiters with a lower penalty value, as e.g. 'TRM_E(abbr)' in Table 1. These empty word delimiters are not tagged with ':WORD_END', so word analysis is triggered only at the unambiguous ends of orthographic words.

We illustrate the use of empty word delimiters for the analysis of contracted word forms. In the sentence in Figure 1 one example is the token sequence "'s", that can be a contracted form of a verb, a contracted personal pronoun or the suffix of a noun in possessive form. As illustrated, four different lexemes of the lexicon in Table 1 match "'s" and are inserted into the chart. In case of "it's␣" word analysis returns only three morphologically correct sequences of syntactic words: a personal pronoun 'PERS_E' followed by either the contracted form of the personal pronoun "us" (PERS_E) or of the auxiliaries "be" (AUXB_E) or "have" (AUXH_E). In case of "mary's␣" the second word grammar rule of Table 2 additionally allows a morphological analysis of the complete orthographic word as possessive form of a proper noun 'NPR_E'.

Another example is the token sequence "st.". This may be an abbreviation of the noun "street" or the noun title "Saint". The period may be part of the abbreviations or a full stop indicating the end of the sentence. Lexicon lookup inserts two lexemes for the stem "st" (NS_E and NTS_E) and four for the according endings (NE_E and NTE_E) into the chart. These endings allow to form abbreviations with or without period. Additionally, lexemes for the punctuation symbol 'PCT_E' are inserted. Word analysis produces four different readings for this token sequence: a noun 'N_E' or a noun title 'NT_E' or a sequence of a noun or a noun title followed by a punctuation symbol 'PCT_E'.

Sentence and paragraph analysis produce finally the correct reading for each contracted word form as long as they can be disambiguated by syntactic means. Using the sentence grammar rules listed in Table 3 the sentence of Figure 1 can be correctly analyzed as an English sentence S_E. The first "'s" is an auxiliary "be" (AUXB_E), and the second "'s" is the possessive form of a proper noun. The first "st." is analyzed as abbreviation of "Saint", while the second one is the abbreviation of "street" followed by a full stop.

As can be verified in Figure 1 this input sequence could also be analyzed as a sequence of two English sentences. Doing so, the first "st." would be incorrectly analyzed as abbreviation of "street", and the second "'s", also incorrectly, as an auxiliary "be".

```
PCT_E (?F,?T) ==> PCT_E (?F,?T) * :SENT_END
PRGTRM ()     ==> PRGTRM () * :SENT_END
N_E (?N,?G,?SG) ==> NS_E (?NCL,?SG,?G)
                    NE_E (?NCL,?N)
                    NGE_OPT_E (?SG,?N)
                    TRM_E (?NCL) *
NPR_E (?N,?G,?SG) ==> NPRS_E (?NCL,?SG,?G)
                      NE_E (?NCL,?N)
                      NGE_OPT_E (?SG,?N)
                      TRM_E (?NCL) *
NGE_OPT_E (?SG,?N) ==> *  0 :INV
NGE_OPT_E (?SG,?N) ==> NGE_E (?SG,?N) *  0 :INV
NT_E () ==> NTS_E (?NTCL)  NTE_E (?NTCL)
            TRM_E (?NTCL) *
AUXB_E (?N,?P,?M,?T,?POS) ==>
                        AUXBS_E (?N,?P,?M,?T,?POS)
                        TRM_E (std) *
PERS_E (?NR,?P,?G,?C) ==> PERSS_E (?NR,?P,?G,?C)
                          TRM_E (std) *  0
```

Table 2: Rules from the English word grammar. A grammar rule is optionally followed by a penalty value. The keyword ':INV' after a grammar rule makes the corresponding branch of the resulting syntax tree invisible. The keyword ':SENT_END' specifies a word constituent to be a possible sentence end.

```
PRGTRM () ==> PRGTRM () * :PARA_END
S_E (?T) ==> PERS_E (?N,?P,?,s)
             VP_E (ind,?T,?N,?P,?,fin)
             PP_E ()
             PCT_E (f,s) *
VP_E (inf,?T,?N,?P,?,?) ==>
                    AUXB_E (?N,?P,inf,?T,pos) *
PP_E () ==> PREP_E (?) NP_E (?,?) *
NP_E (?N,?G) ==> NPRP_E (?,?) N_REP_E (?N,?G) *
N_REP_E (?N,?G) ==> N_E (?N,?G,?) * :INV
N_REP_E (?N,?G) ==> N_E (?,?,?)
                    N_REP_E (?N,?G) * :INV
NPRP_E (?N,?G) ==> NT_E (?)
                   NPR_REP_E (?N,?G) * :INV
NPR_REP_E (?N,?G) ==> NPR_E (?N,?G,?) * :INV
NPR_REP_E (?N,?G) ==> NPR_E (?,?,?)
                      NPR_REP_E (?N,?G) * :INV
```

Table 3: Rules from the English sentence grammar. The keyword ':PARA_END' specifies a sentence constituent to be a possible paragraph end.

Paragraph grammar rules, as shown in Table 4, that define a paragraph as a sequence of sentences, prevent this incorrect analysis result. As the penalty values of grammar rule production and of the rule subconstituents are added up to form the penalty value of the rule head, the penalty value of a paragraph consisting of the two short sentences is higher (7 + 59 + 67) than the penalty value of a paragraph consisting only of the longer sentence (2 + 70).

2.2. Analysis of multi-word lexemes

The approach presented here is also well-suited for multi-word lexemes. E.g. consider the preposition ”in front of”: As blank characters are processed like other characters, lexicon lookup treats multi-word lexemes like any other lexeme. Additionally, word analysis is started only at the end of such a multi-word lexeme, because the associated chart edge spans the whole multi-word lexeme including the blank characters. Thus, word analysis is not triggered after ”in” and ”front”.

To describe ”in front of” as a multi-word lexeme is very convenient for syntax analysis, whereas it is not relevant for pronunciation. For other word forms, like the adverb ”in fine”, pronounced as [ɪn ˈfa͡ɪni], multi-word analysis is a necessity to disambiguate it from the preposition ”in” [ɪn] followed by the adjective ”fine” [fa͡ɪn]. E.g. consider the sentence ”He’s in fine condition in fine.”: Using multi-word lexemes, the final ”in fine” can be correctly analyzed as an adverb.

3. Sentence end identification

Similar to the identification of syntactic words, sentence end identification also requires morphological and syntactic knowledge. In our approach we analyze punctuation symbols as a special form of syntactic words. Thus, the end of a sentence is determined within morphological and syntactic analysis. The following points summarize the general ideas in sentence end identification:

- In case of *unambiguous sentence-final punctuation symbols*, sentence analysis can be started immediately. This is done at chart vertices where all word category edges that end in this vertex are tagged with the keyword `':SENT_END'`.
- For *ambiguous punctuation symbols*, all alternative word categories are inserted into the chart and sentence analysis is not started until the next unambiguous sentence end has been reached.

Figure 2 illustrates both situations: In case of `"street.␣"`, as presented on the left side, word analysis returns an English noun ‘`N_E`’ with an empty noun ending ‘`NE_E`’ that is terminated by an empty word delimiter ‘`TRM_E`’. This noun is followed by an unambiguous sentence end ‘`PCT_E`’ that spans the period and the blank character, cp. Table 1.

In contrast to this, the right side of Figure 2 shows word analysis results in case of an ambiguous sentence end. The period in the input sequence `"st.␣"` may be a full stop indicating the sentence end as well as the termination of the abbreviation

```
P_E () ==> S_REP_E () *
S_REP_E () ==> S_E (?) * :INV
S_REP_E () ==> S_E (?)
               S_REP_E () * 5 :INV
```

Table 4: Rules from the English paragraph grammar.

of ”street” or ”Saint”. Word analysis therefore produces four different word sequences for this input: a noun ‘`N_E`’ or a noun title ‘`NT_E`’ or a sequence of a noun or a noun title followed by a punctuation symbol ‘`PCT_E`’.

These alternative word sequences can be disambiguated by subsequent syntax analysis. Figure 1 illustrates such a disambiguation: As sentence end decision in chart vertex 13 is ambiguous (two word category edges without `':SENT_END'` end in this vertex), sentence analysis is not started until the final paragraph boundary symbol `"<PB>"` has been reached. Sentence analysis produces two different sentence sequences containing two different readings of the first period, i.e. a full stop or part of an abbreviation. Subsequent paragraph analysis finally disambiguates the category of this punctuation symbol by selecting the sentence sequence with minimal total penalty, as described in Section 2.1.

4. Analysis of mixed-lingual sentences

Mixed-lingual sentences can contain contracted word forms, abbreviations or multi-word lexemes of multiple languages simultaneously. These word forms may even be homographs or mixed-lingual word forms themselves. For a mixed-lingual analyzer it is therefore necessary to apply the rules for identification of word contractions, abbreviations, multi-word lexemes and sentence ends of all these languages simultaneously.

The approach for identification of syntactic words as presented in Section 2 can be extended for analyzing mixed-lingual sentences. We construct such a mixed-lingual analyzer following the procedure described in [1]: First we have to design the corresponding set of monolingual analyzers that support the approach described in Section 2. Each monolingual analyzer includes its own lexicon and its own word, sentence and paragraph grammars. As for all grammars the same DCG formalism is used, it is possible to apply the same chart parser for all of these monolingual analyzers.

Then we have to design for each language pair a so-called inclusion grammar. These bilingual inclusion grammars define the elements of one language that are allowed as foreign in-

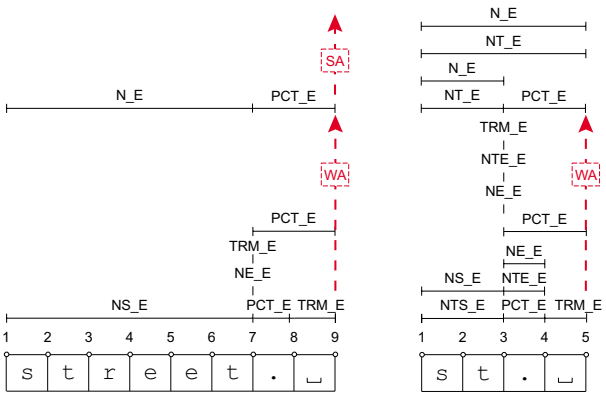

Figure 2: For the input text ”Street. ” word analysis returns a noun ‘`N_E`’ followed by an unambiguous sentence end ‘`PCT_E`’. Thus, sentence analysis is started at chart vertex 9. In case of the input text ”St. ” the period is ambiguous: it is either a punctuation symbol ‘`PCT_E`’ or part of a noun ‘`N_E`’ or a noun title ‘`NT_E`’. Therefore sentence analysis is not triggered at vertex 5.

clusions in the other language. In order to get a mixed-lingual analyzer we have to load all monolingual lexica and grammars together with their bilingual inclusion grammars. This mixed-lingual analyzer is now able to process sentences like

> ”Er hat’s mit *Red Hat’s Journaling File System* probiert.”
> (He tried it with Red Hat’s journaling file system.)
>
> ”Comment avez-vous osé vous attaquer à l’*Adagio* d’*Hammerklavier*?”
> (How did you dare to tackle the Adagio of the Hammerklavier?)

The resulting chart of mixed-lingual syntax analysis of the first sentence is illustrated in Figure 3: the two homographs ”hat’s” are correctly analyzed as a German verb ”hat” (has) plus contracted pronoun ”es” (it) and as possessive form of the English noun ”hat”. Also the English noun phrase ’NP_E’ is correctly identified and mapped onto a German noun phrase using an inclusion grammar rule.

In the second sentence the mixed-lingual contracted forms ”l’Adagio ” and ”d’Hammerklavier” are correctly analyzed as Italian and German inclusions with contracted French determiners.

5. Languages without word separation

Chinese or Japanese texts normally lack word separation characters. As our text analysis processes the input character-wise and does not rely on a designated word separation symbol, it is also well suited for processing such texts.

This can be demonstrated by means of an English example: If all blank characters are removed from the sentence of Figure 1 the resulting input sequence is `"it’sinst.mary’sst."`. Figure 4 illustrates a simplified chart from morphological and syntactic analysis of this sequence.

It is easy to verify that the syntactic parse tree of Figure 4 is exaclty the same as the one of Figure 1.

Another problem processing texts of these languages is that the same character sequence may be split differently into words depending on syntactic and semantic contexts, cp. [5]. As an example, consider the Chinese character sequence 研究生 , that forms a complete noun in the sentence

研究生	亦般	年闹	大
yan2-jiu1-sheng1	yi4-ban1	nian2-ling2	da4
’Master student’	’generally’	’age’	’old’

whereas it is separated into a verb and a noun prefix in sentence:

他	宅	**研究**	生 命起源
ta1	zhai	**yan2-jiu1**	**sheng1**-ming4-qi3-yuan2
’He’	’doing’	’research’	’the origin of life’

As long as such character sequences are lexically ambiguous, the text analysis presented here can correctly disambiguate them using appropriate morphological and syntactic grammar rules.

Furthermore, texts of these languages often contain characters of multiple alphabets within one sentence like traditional Han characters, modern Latin characters plus foreign English inclusions. Such sentences can be analyzed using the mixed-lingual text analysis approach of Section 4.

6. Conclusions

The text analysis component of a TTS system is confronted with ambiguous word and sentence boundaries. For certain languages and especially in the case of mixed-lingual texts, the ambiguity problem makes word token-based parsing virtually impossible. The approach presented here solves most of the ambiguity problems and particularly allows to correctly analyze contracted word forms, multi-word lexemes and sentence ends in mixed-lingual sentences as long as they can be disambiguated by morphological or syntactic means.

We have analyzed a corpus of 50 mixed-lingual sentences containing English, French, German and Italian inclusions using the approach presented in this paper. These sentences including morphological and syntactic analysis results are available on our web site
<http://www.tik.ee.ethz.ch/~spr/SVOX/polysvoxdemo/>.

7. Acknowledgments

We cordially thank Alexis Wilpert and Yan Bi for providing the Chinese example sentences.

This work was partly supported by the Swiss National Science Foundation in the framework of NCCR IM2.

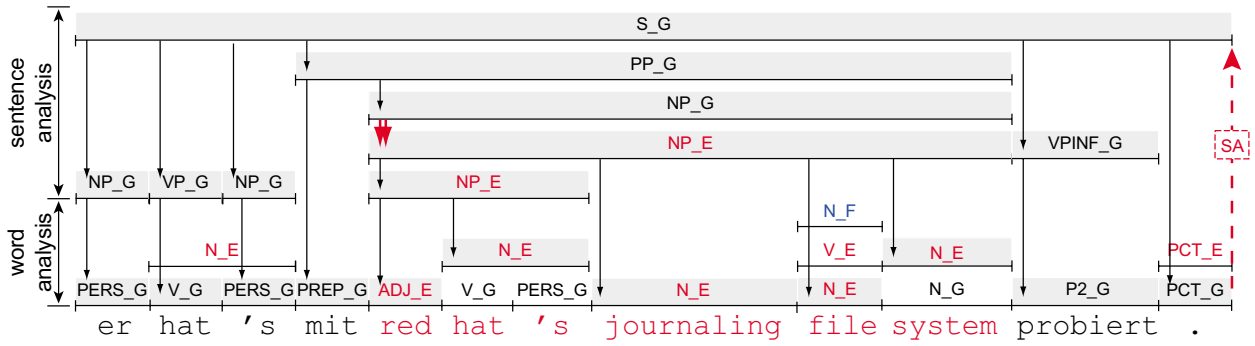

Figure 3: Representation of the simplified chart resulting from mixed-lingual syntactic analysis of the sentence ”Er hat’s mit Red Hat’s File System probiert.”: At the bottom the normalized input character sequence is shown. Edges are drawn without constituent feature values. Arrows with dashed lines indicate trigger events. A doubled arrow indicates a production of an inclusion grammar rule. The constituents of the final syntactic parse tree are shown with grey background.

8. References

[1] B. Pfister and H. Romsdorfer. Mixed-lingual text analysis for polyglot TTS synthesis. In *Proceedings of Eurospeech'03*, pages 2037–2040, Geneva, Switzerland, September 2003.

[2] H. Romsdorfer and B. Pfister. Multi-context rules for phonological processing in polyglot TTS synthesis. In *Proceedings of Interspeech 2004 – ICSLP*, pages 737–740, Jeju Island (Korea), October 2004.

[3] C. Traber. *SVOX: The Implementation of a Text-to-Speech System for German*. PhD thesis, No. 11064, Computer Engineering and Networks Laboratory, ETH Zurich (TIK-Schriftenreihe Nr. 7, ISBN 3 7281 2239 4), March 1995.

[4] R. Sproat. Multilingual text analysis for text-to-speech synthesis. In *Proceedings of ICSLP'96*, Philadelphia, October 1996.

[5] R. Sproat, S. Chilin, W. Gale, and N. Chang. A stochastic finite-state word-segmentation algorithm for chinese. In *Computational Linguistics*, 1996.

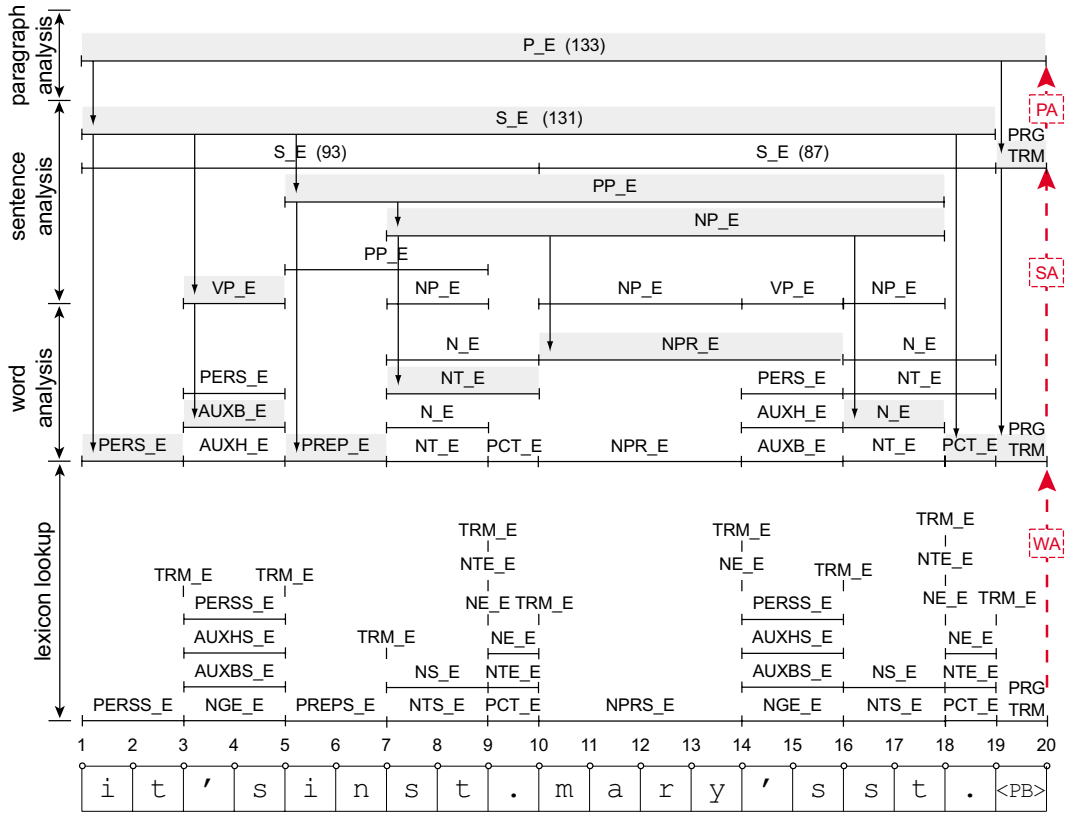

Figure 4: Representation of the simplified chart resulting from morphological and syntactic analysis of the sentence "it'sinst.mary'sst.": At the bottom the normalized input character sequence is shown. Edges are drawn without constituent feature values. For a set of edges with the same associated constituent but different feature values that span the same vertices only one edge is shown. The "lexicon lookup" section contains edges associated to the lexemes found during lexicon lookup. The "word analysis", "sentence analysis" and "paragraph analysis" sections contain edges associated to constituents resulting from the respective analysis levels. The minimal penalty values of sentence and paragraph constituents are denoted in parenthesis at their associated edges. Arrows with dashed lines indicate trigger events. The constituents of the final syntactic parse tree are shown with grey background.