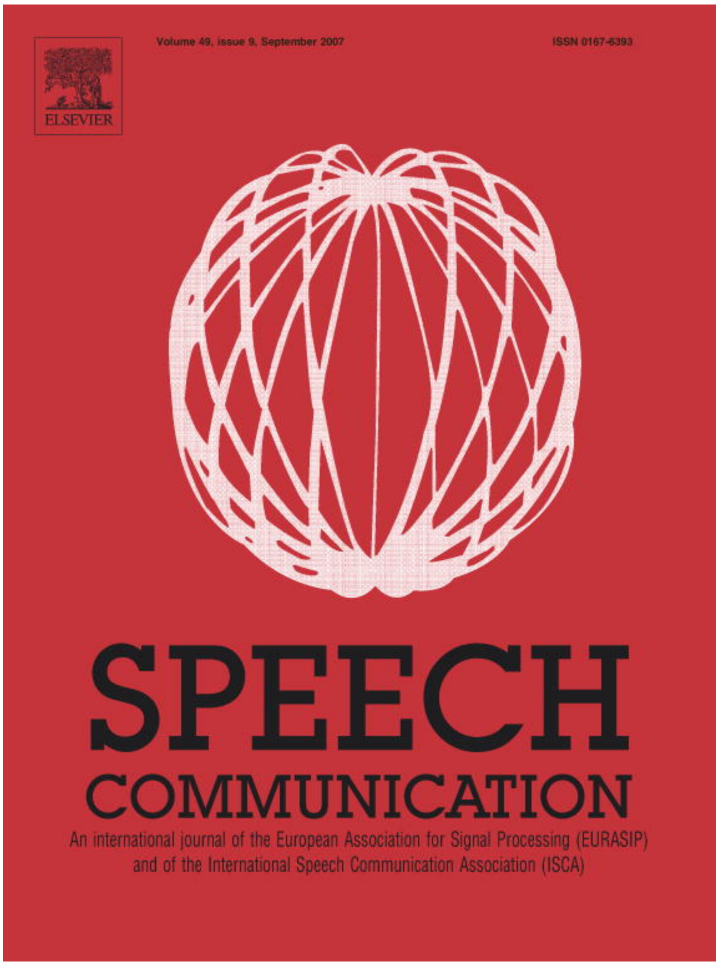
Volume 49, issue 9, September 2007
ISSN 0167-6393
ELSEVIER
SPEECH
COMMUNICATION
An international journal of the European Association for Signal Processing (EURASIP)
and of the International Speech Communication Association (ISCA)

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Speech Communication 49 (2007) 697–724

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Text analysis and language identification for polyglot text-to-speech synthesis

Harald Romsdorfer *, Beat Pfister

Speech Processing Group, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland

Received 14 September 2006; received in revised form 4 April 2007; accepted 13 April 2007

Abstract

In multilingual countries, text-to-speech synthesis systems often have to deal with texts containing inclusions of multiple other languages in form of phrases, words, or even parts of words. In such multilingual cultural settings, listeners expect a high-quality text-to-speech synthesis system to read such texts in a way that the origin of the inclusions is heard, i.e., with correct language-specific pronunciation and prosody. The challenge for a text analysis component of a text-to-speech synthesis system is to derive from mixed-lingual sentences the correct polyglot phone sequence and all information necessary to generate natural sounding polyglot prosody.

This article presents a new approach to analyze mixed-lingual sentences. This approach centers around a modular, mixed-lingual morphological and syntactic analyzer, which additionally provides accurate language identification on morpheme level and word and sentence boundary identification in mixed-lingual texts. This approach can also be applied to 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. Because of its modular design it is easily extensible to additional languages.

PACS: 43.72.Ja; 43.72.−p

Keywords: Polyglot speech synthesis; Mixed-lingual text analysis; Language identification; Morphological and syntactic analysis; Word and sentence boundary identification

1. Introduction

Texts, especially in multilingual countries like Switzerland, become more and more mixed-lingual. This means that texts may contain inclusions from other languages. These foreign inclusions comprise part of words, full word forms, word groups, and complete sentences. In such multilingual cultural settings, listeners expect a high-quality text-to-speech (TTS) synthesis system to read such texts in a way that the origin of the inclusions is heard, i.e., with correct language-specific pronunciation and prosody. The polyglot TTS synthesis system *polySVOX* of ETH Zurich meets exactly these requirements.

For the sake of clarity, we distinguish between multilingual and polyglot TTS synthesis systems:

- A *multilingual* TTS synthesis system can transform sentences in one of several languages into a speech signal. Usually, no language identification is included, but the user of the system has to set the language manually. In general, each language is treated by an independent subsystem and synthesized with a language-specific voice. It is therefore impossible to switch the language within a sentence.
- A *polyglot* TTS synthesis system can process texts with part of words, full words, word groups, or sentences of different languages. For such a system, language

[*] Corresponding author. Tel.: +41 1 632 55 48; fax: +41 1 632 10 35.
E-mail address: romsdorf@tik.ee.ethz.ch (H. Romsdorfer).
URL: www.tik.ee.ethz/~romsdorf (H. Romsdorfer).

0167-6393/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2007.04.006

identification of the text is indispensable. Likewise, all languages must be synthesized with the same voice and it must be possible to switch the language at any position within a sentence.

In order to cope with the high complexity of polyglot TTS synthesis, we constructed our polyglot TTS synthesis system from independent monolingual systems. This approach is feasible provided the architecture of the monolingual systems has been chosen suitably. Our polyglot TTS synthesis system polySVOX shown in Fig. 1 implements such an architecture.

This architecture strictly separates language-independent algorithms from language-dependent linguistic and acoustic data. Furthermore, voice-independent and voice-dependent parts are separated. The voice-independent part includes text analysis and phonological processing. It transforms the input text into the so-called *phonological representation*, i.e., a minimal, voice-independent abstract description of the speech to be synthesized. Concretely, the phonological representation includes the phonetic symbols and the abstract description of prosody of the sentence to be uttered. The voice-dependent part, composed of prosody control and speech signal generation, produces from the phonological representation the speech signal.

The polySVOX system transforms a text paragraph by paragraph in four steps into speech. Fig. 1 illustrates these steps. The applied methods of the four corresponding system components are as follows:

- *Text analysis* derives the morphological structure of the words and the syntactic structure of the sentences, and delivers the phonetic transcription and the language identification of each morpheme. For text analysis, strictly rule-based processing is applied, i.e., a chart parser, which uses word, sentence, and paragraph grammars and two-level rules for lexicon-to-surface mapping implemented as finite state transducers (cf. Traber, 1995; Pfister and Romsdorfer, 2003; Romsdorfer and Pfister, 2006).
- *Phonological processing* applies phonological transformations, like schwa elision, French liaison, or English linking-r, which are formulated using so-called multi-context rules. It also assigns sentence stress and prosodic phrase boundaries based on the syntactic structure of a sentence. This abstract prosodic description, together with the phonetic transcription of each word, constitutes the phonological representation (cf.Romsdorfer and Pfister, 2004; Romsdorfer et al., 2005).
- *Prosody control* generates from the phonological representation the physical prosodic parameters. These are the duration values of all phones and the fundamental frequency contour of a sentence. Phone duration and fundamental frequency control are realized by means of trainable statistical models (artificial neural networks), which directly map the symbols of the phonological representation onto phone duration and fundamental frequency values (cf. Traber, 1995; Riedi, 1997; Romsdorfer and Pfister, 2005).

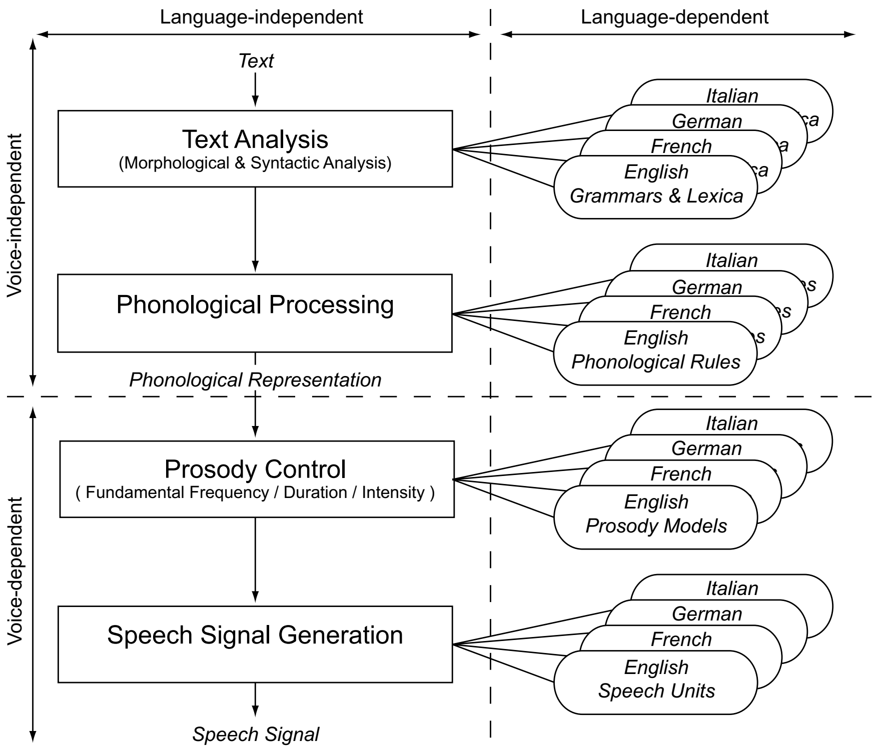

Fig. 1. Architecture of the polyglot TTS synthesis system “polySVOX”.

- *Speech signal generation* is based on concatenation of diphone units extracted from natural speech. Prior to concatenation the diphones have to be prosodically modified such that they match the specified phone duration and fundamental frequency values (cf. Traber, 1995; Traber et al., 1999).

It has to be emphasized that this architecture suits monolingual as well as polyglot TTS synthesis. Basically, the linguistic and acoustic data define the set of languages that can be processed. Thus, the current set of languages of the polySVOX system can easily be expanded to new languages.

In this article, we will focus on the text analysis part of the polySVOX system. This component is responsible for language identification, the generation of the phonetic transcription, and the analysis of the syntactic structure of the text. We first discuss in Section 2 mixed-lingual text analysis and language identification in general. In Section 3 we then describe our approach to mixed-lingual text analysis and language identification. Section 4 illustrates our approach to grapheme-to-phoneme conversion of unknown words in mixed-lingual texts. In Section 5 we present a solution to mixed-lingual word and sentence boundary identification, and in Section 6 we show evaluation results of language identification using our mixed-lingual text analysis.

2. Mixed-lingual text analysis

The task of mixed-lingual text analysis within a polyglot TTS synthesis system is most easily illustrated by means of mixed-lingual example sentences. Table 1 lists some sentences with various foreign inclusions, as they can be found in Swiss newspapers or on Swiss web pages. The inclusions are bounded by brackets and are indexed according to their language either as ($_{E}$English), ($_{F}$French), ($_{G}$German), or ($_{I}$Italian). The sentences themselves are also bounded to indicate the sentence's base language.

2.1. Requirements for mixed-lingual text analysis

The requirements for the text analysis component of a polyglot TTS synthesis arise from the texts, which have to be converted into speech, and from phonological and prosodic requirements of the subsequent synthesis steps. Therefore, we first illustrate the language mixing phenomena typically encountered in published texts and then review the phonological and prosodic requirements.

2.1.1. Language mixing phenomena

The mixed-lingual sentences in Table 1 illustrate basically three major types of foreign inclusions:

Mixed-lingual words that are produced from a foreign stem by means of base language declension or conjugation or by means of compound word formation together with a base language word. Examples for such mixed-lingual words in Table 1 are:

- "($_{G}$($_{E}$up)ge($_{E}$dat)et)" in sentences 12 and 16: some German past participle construction of the English verb "to update".
- "($_{G}$($_{E}$Musical)programm)" in sentence 12: a German compound noun construction of the English noun "musical" and the German noun "Programm".

Table 1
Examples of mixed-lingual sentences with various foreign inclusions

(1)	($_{E}$Asia welcomes ($_{F}$bon ami Chirac).)
(2)	($_{E}$One of French ($_{F}$nouvelle cuisine)'s objectives is to cook foods lightly.)
(3)	($_{E}$She's not really ($_{F}$au fait) with my ideas.)
(4)	($_{F}$Á la mi-mars, le ($_{E}$Tokyo Game Show) sera l'occasion de nouvelles annonces pour ces "($_{E}$world game companies)".)
(5)	($_{F}$Comment avez-vous osé vous attaquer à l'Adagio d'($_{G}$Hammerklavier)!)
(6)	($_{G}$Wird das ($_{F}$Café) nicht von Ihren ($_{E}$Fans) belagert?)
(7)	($_{G}$In 50 m nach links in die ($_{F}$Avenue de l'église) abbiegen!)
(8)	($_{G}$Der Teilnehmer ist ($_{F}$François Lejeune), ($_{I}$via Roggiana) 16, 6945 ($_{I}$Origlio).)
(9)	($_{G}$Die ($_{F}$Femme fatale) ist die zentrale Frauenfigur des ($_{F}$Film noir).)
(10)	($_{G}$Tessiner Städte im ($_{I}$Italianità) ($_{E}$Rating).)
(11)	($_{G}$Die ($_{E}$Greatest Nation) hat die ($_{F}$Grande Nation) als tonangebende Nation abgelöst.)
(12)	($_{G}$Das ($_{E}$Musical)programm ($_{E}$New York's) wurde ($_{F}$en passant) ($_{E}$up)ge($_{E}$dat)et.)
(13)	($_{G}$($_{E}$Lobbying) ($_{F}$à discrétion) vor der Vergabe der Olympischen Spiele von 2012 in Singapur.)
(14)	($_{G}$Geniessen Sie einen ($_{I}$Caffè Latte) oder eine feine italienische Spezialität im ($_{F}$Salon Rouge) des Landesmuseums.)
(15)	($_{G}$Bis Ende März will sich der Kölner Konzern entscheiden, wie es mit dem ($_{E}$Discounter), der Bestandteil der Schweizer Tochter ($_{F}$Bon appétit) ($_{E}$Group) ist, weitergehen soll.)
(16)	($_{G}$($_{F}$Peu à peu) wird der ($_{E}$High Performance) ($_{F}$Fonds) vom ($_{F}$Fonds)($_{E}$manager) ($_{E}$up)ge($_{E}$dat)et.)
(17)	($_{G}$Mit einer ($_{E}$Internet)nutzerschaft von 38% ist die ($_{I}$Svizzera italiana) der kleinste ($_{E}$Internet)markt der drei Regionen.)
(18)	($_{G}$($_{F}$Zidane) plant ($_{E}$Comeback) in der ($_{F}$Equipe Tricolore).)
(19)	($_{G}$Der ($_{E}$Teammanager) ($_{I}$Luigi Riva) sieht im höheren Altersdurchschnitt der ($_{F}$Equipe Tricolore) keinen Vorteil für die ($_{I}$Squadra Azzurra).)
(20)	($_{G}$Die ($_{E}$Air Force 1) landet in Frankfurt.)
(21)	($_{G}$Kunstvolles Dekor im ($_{F}$Louis XIV) Stil.)
(22)	($_{G}$Ich ($_{E}$dat)e das System ($_{E}$up).)
(23)	($_{I}$Come potevo pretendere di condurra la mia ($_{F}$recherche) con lo sfintere?)
(24)	($_{I}$($_{E}$General Motors) pagherà a Fiat 1,55 miliardi di euro per risolvere il ($_{E}$Master Agreement), inclusa la cancellazione della ($_{E}$put option).)

The inclusions are bounded by brackets and are indexed according to their language.

- "(${}_{\mathrm{E}}$(${}_{\mathrm{F}}$cuisine)'s)" in sentence 2: an English s-genitive construction of a French noun.

Full foreign words that are embedded in a base language context. These word forms follow foreign morphology, but possibly disagree syntactically with the base language context. Examples in Table 1 are:

- "(${}_{\mathrm{G}}$das (${}_{\mathrm{F}}$Café))" in sentence 6: a French masculine noun embedded as a German neuter noun.
- "(${}_{\mathrm{I}}$della (${}_{\mathrm{E}}$put option))" in sentence 24: an English neuter compound noun embedded as an Italian feminine noun.

Foreign multi-word inclusions, which are syntactically correct foreign constituents. These foreign constituents are embedded within the base language context according to the base language's syntax. Table 1 also contains examples of this inclusion type, like:

- "(${}_{\mathrm{E}}$New York's)" in sentence 12: an English s-genitive construction, which is embedded in the German sentence, according to the German syntax, after the referent.
- "(${}_{\mathrm{F}}$Avenue de l'église)" in sentence 7: a French noun phrase embedded in the German sentence in place of a German noun.
- "(${}_{\mathrm{E}}$Lobbying) (${}_{\mathrm{F}}$à discrétion)" in sentence 13: a mixed-lingual multi-word inclusion, which consists of an English noun and a French prepositional phrase, embedded as a German noun phrase.

2.1.2. Phonologic and prosodic requirements

Multilingual listeners expect mixed-lingual sentences to be read in a way that the origin of foreign inclusions is heard. Particularly, a polyglot TTS synthesis system must generate the correct language-specific sequence of phones with appropriate prosody. This means that polyglot TTS synthesis must comply with the following phonologic and prosodic requirements:

- *Language-specific pronunciation*: foreign inclusions must be pronounced in a language-specific manner. E.g., in Switzerland the French noun "Avenue" in the German sentence 7 of Table 1 must be pronounced [av(ə)ny] using French phones (and not in a German fashion *[aveːnuə]).
- *Language-specific word stress*: the word stress of foreign inclusions must follow language-specific rules; thus, French nouns in German sentences, like "Avenue" [av(ə)'ny], are end-stressed, even if German nouns are generally front-stressed. Applying a German word stress pattern, e.g., *['av(ə)ny], makes the word difficult to understand.
- *Language-specific phonological phenomena*: phonological transformations within longer foreign inclusions follow the phonological rules of the inclusion language. E.g., the application of German phonological rules onto the French noun phrase "Bon appétit" in the German sentence 14 of Table 1 produces the incorrect transcription *[bõ-ʔa-pe-ti]. This pronunciation sounds strange to Swiss listeners, as it lacks the French liaison consonant and has a German glottal stop inserted (as it is usually done in German sentences before a vowel starting a word). In contrast, the application of French phonological rules, like denasalization and liaison, results in the correct transcription [bɔ-na-pe-ti].
- *Language-specific sentence accentuation*: the intonation of larger, multi-word foreign inclusions, like "world game companies" in sentence 4 of Table 1, follows the foreign sentence accentuation patterns. Thus, "world game companies" is accentuated [[2]wɜːld-[1]ge͜ɪm-[3]kʌm-pə-niz][1] according to English accentuation, and not according to the accentuation of the sentence's base language, French: *[[2]wɜːld-ge͜ɪm-[3]kʌm-pə-[1]niz].
- *Language-specific phrasing*: placement of phrase boundaries within foreign inclusions disobeys in general the base language's phrasing rules; the phrasing rules of the inclusion language specify their correct placement. E.g., in German and English sentences nouns are followed by potential phrase boundaries; in the French inclusion "Salon Rouge" in the German sentence 14 of Table 1, however, no phrase boundary may be placed after the noun "Salon", as the subsequent adjective "Rouge" is part of the French noun phrase.

2.2. Consequences for mixed-lingual text analysis

Text analysis of a TTS synthesis system that has to pronounce sentences like the ones of Table 1 and thereby meet the requirements given in Section 2.1 must fulfill three basic tasks:

- language identification,
- language-dependent phonetic transcription, and
- language-dependent syntactic structure analysis.

We describe these tasks in more detail in the following subsections. Our approach to accomplish these tasks is outlined in Section 3.

2.2.1. Language identification and language-dependent transcription

First of all, mixed-lingual text analysis must be capable to identify the correct language of each portion of the input text. This is necessary in order to transcribe these text portions according to their languages and in order to apply appropriate word stress. The size of such portions, as shown in Table 1, varies from single morphemes to complete sentences.

The task of identifying the correct language of a given portion of text is made additionally difficult by interlingual homographs like, e.g., "hat", which is an English noun as well as a German verb, or "die", which is an English verb

[1] [1] Denotes the main phrase accent; [2] and [3] denote weaker accents.

as well as a German determiner. Certain types of interlingual homographs, like loanwords, logograms, abbreviations, or acronyms, are especially difficult to disambiguate:

> *Loanwords* are strongly assimilated to the base language, not only in morpho-syntactic terms, but also with respect to the pronunciation. Loanwords in mixed-lingual text may, however, raise an additional issue concerning homographs in places where their pronunciation depends on the language context. Consider, e.g., the word "Nation" in sentence 11 of Table 1, which is first pronounced in English as [’ne͜ɪʃən], then in French as [nɑ’sjɔ̃] and finally in German as [na’t͜si̯oːn].
>
> *Logograms* like numbers, Roman numerals, currency units, or special symbols ("%", "& ", etc.) are also a form of interlingual homographs. The correct pronunciation of these logograms depends on the language context. E.g., the two sentences, "(${}_{G}$Die (${}_{E}$Air Force 1) landet in Frankfurt.)" and "(${}_{G}$Kunstvolles Dekor im (${}_{F}$Louis XIV) Stil.)", contain examples of logograms that are part of foreign inclusions and that are therefore pronounced according to their inclusion languages.
>
> *Abbreviations* are short forms of words with or without final period, which are pronounced as the full form they represent. Common abbreviations are often graphemically identical across multiple languages, but are pronounced differently. For example, the abbreviation "dr" is pronounced in English as [’dɒktə], in French as [dɔk’tœːʀ], in German as [’dɔktoːɐ̯] and in Italian as [dot’toːre].
>
> *Acronyms* are short forms of words or phrases, which are either spelled or pronounced. Acronyms are usually composed of the initial letters of the words they symbolize. There exist a lot of graphemically identical acronyms across multiple languages. The rules of pronunciation, however, vary a lot depending on the language: in Italian most acronyms are read, in German and English they are normally spelled, and in French, according to Boula de Mareüil and Floricic (2001), approximately half of the acronyms are read and half are spelled. As an example consider "IRA", which is read in French [i’ʀa] and Italian [’iːra], but spelt in English [ˌa͜ɪɑːr’e͜ɪ] and German [iːɛr’aː].

Most of these interlingual homographs can be disambiguated by syntactic means. In Section 3.5 we illustrate how the polySVOX system accomplishes the disambiguation of such interlingual homographs.

2.2.2. Language-dependent syntactic structure analysis

In the polySVOX TTS synthesis system shown in Fig. 1, the mixed-lingual phonological processing component is responsible for language-dependent application of phonological transformation rules, sentence accentuation, and prosodic phrasing. The mixed-lingual syntactic structure of the input sentence forms the basis for application of sentence accentuation patterns as well as sentence phrasing rules. Likewise, application of most language-dependent phonological rules, like French liaison rules, is restricted to language-dependent syntactic contexts.

It is therefore essential that mixed-lingual text analysis provides the phonological processing component also with the mixed-lingual syntactic structure of the input text.

A prerequisite for such a syntactic structure analysis is the correct identification of syntactic word and sentence boundaries. Syntactic words are the terminal elements of syntactic analysis. In contrast to orthographic words, which are delimited by blank characters and which are therefore easily identified by text preprocessing, syntactic words are more difficult to identify and do not always correspond to orthographic words due to different graphemic phenomena:

- *Word contractions*, e.g., English "he's", "Mary's", German "das ist's" (that's it), or Italian "po'd'acqua" (some water).
- *Cross-line hyphenation of words* at line breaks, e.g., consider English "in-<LF>put" vs. "in-<LF>and output" ('<LF>' is the line end symbol).
- *Multi-word lexemes*, i.e., word forms spanning multiple orthographic words, like 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 at the same time be a full stop to indicate the end of the sentence.
- *Missing designated word separation symbols* in languages like Chinese or Japanese. E.g., Sproat et al. (1996) give a good overview of the problems text analysis for Chinese is confronted with.

We explain our approach to syntactic word and sentence boundary identification in Section 5.

3. Mixed-lingual morphological and syntactic analysis

In this section we describe our approach to mixed-lingual text analysis as implemented in the polySVOX TTS synthesis system. This approach simultaneously solves all three tasks given in Section 2.2: language identification, language-dependent phonetic transcription, and language-dependent syntactic structure analysis.

In order to compare our approach qualitatively with other approaches to language identification, Section 3.1 first gives a survey of other state-of-the-art algorithms for language identification. Section 3.2 explains the concept of our approach in more detail. The subsequent Sections 3.3 and 3.4 describe solutions for all types of inclusion phenomena listed in Section 2.1.1 by applying the polySVOX mixed-lingual text analysis. Finally, Section 3.5 shows a solution to the disambiguation of interlingual homographs.

3.1. Language identification

Language identification from written text is important in different application areas, including mixed-lingual TTS synthesis, multilingual speech recognition (e.g., Tian et al., 2002), and document classification (e.g., Cavnar and Trenkle, 1994). Therefore, numerous approaches towards this task exist. The majority of them is based on statistical information about word and character sequences of the languages in question. These approaches usually apply a character context window of a fixed length onto the input character sequence; an often used window contains three characters in front of and three after the character in question. The most likely language for a given word or a sentence is then calculated employing methods like *n*-grams (Schmitt and October, 1991; Grefenstette, 1995), neural networks (Tian and Suontaustaa, 2004), decision trees (Häkkinen and Tian, 2001), or some combination of them. Some approaches also make use of basic linguistic knowledge, e.g., in form of a heuristic method using language-specific character frequencies plus language-specific lists of function words and word endings (Giguet, 1995). Common to all of these approaches is that the granularity of language identification is either a sentence or at most a word.

As can be verified by the examples of Table 1, language identification on word level is not accurate enough for a polyglot TTS system. Foreign inclusions in mixed-lingual words can be very short, as the English inclusions in the German verb “($_G$($_E$up)ge($_E$dat)et)” demonstrate. Such mixed-lingual words require language identification to be applied on morpheme level. Using statistical information based on a typical character context window, which is larger than the inclusions themselves, it is difficult to identify the language of inclusions in mixed-lingual words correctly.

Additionally, a mixed-lingual sentence, like “($_G$($_F$Peu à peu) wird der ($_E$High Performance) ($_F$Fonds) vom ($_F$Fonds)($_E$manager) ($_E$up)ge($_E$dat)et.)”, demonstrates that there exist words with a majority of characters not belonging to the word’s base language, as well as sentences, in which the sentence’s base language is not the language with the maximum number of words of this sentence. Considering this, simple statistical approaches for identifying the base language of a word or a sentence are too unreliable for language identification in mixed-lingual sentences.

3.2. The polySVOX approach to mixed-lingual text analysis

Investigations of mixed-lingual texts showed that, on the one hand, inclusions of foreign constituents into a context of another language can be restricted by specific bilingual morphological and syntactic rules. And, on the other hand, within the foreign constituents only the foreign monolingual morphological and syntactic rules are relevant.

We want to exemplify the first finding with mixed-lingual German verbs containing English verb stems: mixed-lingual German verbs like

“($_G$($_E$updat)en)”	[ʌp’de͜ɪtn̩]
“($_G$($_E$brows)en)”	[’bra͜ʊzn̩]
“($_G$($_E$scann)en)”	[’skænən]

always contain the present tense form of English verb stems (without silent “e”, but with optional consonant doubling) and follow weak German conjugation. A possible English verb prefix is bound to the English verb stem.

The mixed-lingual German past participle form consists of the German past participle prefix “ge” followed either by the present tense form of the English verb stem plus a German past participle ending or by the complete English past participle. A possible English verb prefix may optionally be separated from the English verb stem and be included in front of the German past participle prefix “ge”. As examples consider the following, equally frequently used mixed-lingual German forms of “updated”:

“($_G$($_E$up)ge($_E$dat)et)”	[’ʔʌpgə,de͜ɪtət]
“($_G$($_E$up)ge($_E$dated))”	[’ʔʌpgə,de͜ɪtɪd]
“($_G$ge($_E$updat)et)”	[gəʔʌp’de͜ɪtət]
“($_G$ge($_E$updated))”	[gəʔʌp’de͜ɪtɪd]

The second finding is illustrated by the mixed-lingual German sentence.

“($_G$Die ($_F$Femme fatale) ist die zentrale Frauenfigur des ($_F$Film noir).)”

As German syntax does not allow adjectives to be placed after the corresponding noun, the syntactic structure of this sentence can only be correctly analyzed if French syntactic rules are applied within the two French noun phrase inclusions. The French noun phrases are then included as noun phrase constituents in the German sentence.

3.2.1. Architecture of the polySVOX text analysis

In the polySVOX system we pursue a modular approach to mixed-lingual text analysis. We strictly separate monolingual analyzers from bilingual inclusion grammars. Each monolingual analyzer contains a monolingual morpheme lexicon as well as a word, a sentence, and a paragraph grammar. A bilingual inclusion grammar contains bilingual grammar rules that describe, which foreign constituents can be mapped as foreign inclusions onto corresponding constituents of the base language. Thus, monolingual grammars need not be modified at all when including new languages. Only small bilingual inclusion grammars are necessary, which are loaded together with the corresponding monolingual grammars. The size of such an inclusion grammar is normally less than five percent of the size of monolingual grammars (e.g., 18 inclusion grammar rules specifying English inclusions in German compared to 797 monolingual German grammar rules).

Fig. 2 illustrates the modular architecture of the polySVOX morphological and syntactic text analysis. It is realized as a bottom-up chart parser for penalty-extended definite clause grammars (DCGs). 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

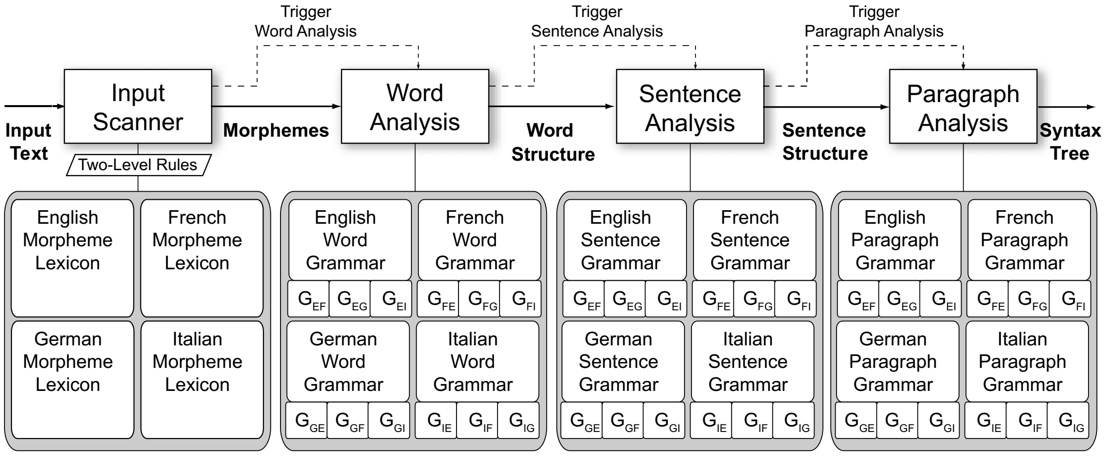

Fig. 2. Architecture of morphological and syntactic analysis of the polySVOX TTS synthesis system. The notation G_{ij} specifies an inclusion grammar that describes inclusions of language j in language i. E, F, G, and I are abbreviations of English, French, German, and Italian, respectively.

looked up in all monolingual morpheme lexica. The chart parser itself operates on three different levels: a word, a sentence, and a paragraph level. Each level is provided with a set of separate monolingual grammars, which are joined by a set of bilingual inclusion grammars. The input scanner triggers word analysis at unambiguous word boundaries in the input stream. Likewise, word analysis starts sentence analysis at unambiguous sentence boundaries. And sentence analysis finally triggers paragraph analysis at unambiguous paragraph boundaries.

The DCG formalism is a natural extension of context-free grammars. This extension is done by augmenting the context-free production rule skeleton with feature terms and the term unification operation. Theoretically, DCGs have the power of a Turing machine, and in that sense are as general as can be (Pereira and Warren, 1980). In the polySVOX system, DCG rules have additionally got a penalty value in order to select the optimal solution among several ambiguous solutions, and they have got optional keywords for controlling the building of parse trees (cp. Traber, 1995) and for identifying syntactic word and sentence boundaries.

The following subsections describe several key aspects of our mixed-lingual text analysis. Section 5 explains the text analysis procedure in detail and illustrates word and sentence boundary identification. Appendix A provides a format specification of lexicon entries and grammar rules, and lists excerpts of the lexica and grammars for English, French, German and Italian.

3.2.2. Inclusion grammars

An inclusion grammar consists of bilingual grammar rules that specify mappings from foreign constituents and their feature terms to corresponding constituents of the base language. Examples of such inclusion grammar rules for English, French, German and Italian are given in Appendix A.

Inclusion grammar rules allow to formulate constraints on including foreign constituents using two basic mechanisms: constituent mapping restrictions and inclusion penalties. These mechanisms are all specified using the extended DCG formalism.

Constituent mapping restrictions allow to map a specific foreign constituent to a base language constituent. E.g., the verb stem inclusion rule R84 in Appendix A.3.4 specifies that only the present tense form of English verb stems VS_E, indicated by the feature value 'pres', may be included as German verb stem VS_G that must additionally follow weak German conjugation.

Inclusion penalties allow to disambiguate interlingual ambiguities. These penalty values are set manually by a linguistic expert according to following guidelines:

- The penalty values of inclusion rules for a given constituent must be generally higher than the overall penalty values of any monolingual analysis of this constituent.
- The penalty values of larger inclusions, e.g., noun phrases, shall be typically lower than the penalty values of smaller inclusions, e.g., nouns or adjectives.
- The penalty values of all inclusion rules for the same constituent must be harmonized across all inclusion languages.

These inclusion grammar rules provide a very accurate description of how foreign inclusions are analyzed in the base language context. The strictly bilingual definition of these rules makes it possible to construct a modular and flexible morphological and syntactic grammar for any desired language combination.

3.2.3. Language switching flag

Inclusion grammars are loaded together with their monolingual grammars. If inclusion grammars of several languages are loaded, cyclic dependencies in bottom-up chart parsing and incorrect analysis results are inevitable.

Cyclic dependencies arise from loading inclusion grammars of two languages specifying inclusions of each other. E.g., loading grammar rule R47 of the English–German inclusion grammar in Appendix A.1.6 and grammar rule R87 of the German–English inclusion grammar in Appendix A.3.4 will result in a cyclic dependency when parsing English or German nouns.

Incorrect analysis results emerge, if certain morphological or syntactic structures that are valid for one language are forbidden in another language, and if inclusion grammar rules exist that specify appropriate mappings between these languages. E.g., the positioning of an adjective after an noun in a French noun phrase, as for example in "le film noir", is forbidden in German noun phrases. The sequence *"der Film schwarze" must therefore not be analyzed as a German noun phrase. However, applying French–German inclusion grammar rules R68 and R69 of Appendix A, and using French sentence grammar rule R64, *"Film schwarze" is analyzed as a French noun phrase that contains two German inclusions. The German–French inclusion grammar rule R101 maps this French noun phrase back to a German noun phrase nucleus. Applying the German sentence grammar rule R79 would result in the incorrect analysis of *"der Film schwarze" as a German noun phrase.

In order to prevent cyclic dependencies in bottom-up chart parsing and incorrect analysis results, all but the first application of inclusion grammar rules for every single lexeme must be inhibited. Therefore, we introduced a so-called "language switching flag" that prevents inclusions of foreign constituents that already contain a foreign inclusion themselves. The flag is basically a Boolean feature term implemented using the DCG formalism. It is therefore completely transparent to the parsing algorithm.

Table 2 shows the application of the language switching flag to the grammar rules which are necessary to analyze the above example noun phrase. Each constituent obtains an additional feature term, which represents the language switching flag. This feature term either has the value `true` or `false`, or is a variable named `LSF1`, `LSF2`, and so forth. Each monolingual grammar rule evaluates this feature by applying Boolean OR to the language switching flags of the constituents of the rule body (cp. the `BOOL_OR` rules, rule R64, and rule R79 in Table 2). Each inclusion grammar rule requires the language switching flag of the foreign constituent to be `false`, and sets the language switching flag of the base language's constituent to `true` (cp. inclusion rules R68, R69, or R101). Thus, the language switching flags of a constituent that contains a foreign inclusion and all constituents derived from it are always set to `true`. Additionally, no inclusion grammar rule can be applied to such a constituent anymore, as this would require their language switching flags to be `false`.

The language switching flag successfully prevents cyclic dependencies in bottom-up chart parsing due to inclusion grammar rules, as it stops possible cycles after the first application of an inclusion grammar rule. The flag also prevents incorrect analysis results like the one illustrated above, as it allows only direct foreign inclusions.

The polySVOX system automatically extends grammar rules by the necessary `BOOL_OR` rules and language switching flag features when loading the grammars. Therefore, the grammar rules listed in Appendix A do not contain language switching flag features or grammar rule extensions.

3.3. Mixed-lingual morphological analysis

The polySVOX approach to mixed-lingual word analysis applies separate monolingual lexica and separate monolingual word grammars plus additional bilingual word inclusion grammars in parallel to parse a given graphemic input sequence morphologically. Appendix A contains example lexica and grammars for the languages English, French, German, and Italian. The central idea when analyzing mixed-lingual input text is to favor always monolingual analysis results over mixed-lingual ones. This is achieved by setting the penalty values of inclusion rules for a given constituent higher than the overall penalty values of any monolingual analysis result of this constituent.

Fig. 3 illustrates on the left side the morphological analysis of the mixed-lingual German word "($_{\mathrm{G}}$($_{\mathrm{E}}$up)ge($_{\mathrm{E}}$dat)et)" (updated) and on the right side the morphological analysis of the monolingual German word "($_{\mathrm{G}}$datiert)" (dated). The stem "dat" is highly ambiguous. It can be an English verb

Table 2
French and German sentence grammar rules showing feature variables `LSF1`, `LSF2`, and `LSF3` as language switching flags

	BOOL_OR (false,false,false)	==>	* 0 :INV
	BOOL_OR (true, true, false)	==>	* 0 :INV
	BOOL_OR (true, false,true)	==>	* 0 :INV
	BOOL_OR (true, true, true)	==>	* 0 :INV
[R64]	NP_F (?N,?P,?G,?**LSF3**)	==>	N_F (?N,?P,?G,?**LSF1**) ADJ_F (n,?N,?G,?**LSF2**) BOOL_OR (?LSF3,?LSF1,?LSF2) *
[R68]	N_F (?NR,?,**true**)	==>	N_G (?NR,?,?,**false**)* 100
[R69]	ADJ_F (?,?N,?,**true**)	==>	ADJ_G (?,?N,?,?,?,**false**) * 100
[R79]	NP_G (?C,?NR,?P,?G,?NT,?**LSF3**)	==>	DET_G (?C,?NR,?G,?F,?TYP,?**LSF1**) NPNUC_G (?C,?NR,?P,?G,?TYP,?NT,?**LSF2**) BOOL_OR (?LSF3,?LSF1,?LSF2) *
[R101]	NPNUC_G (?,?NR,pers3,?,?,?,**true**)	==>	NP_F (?NR,?,?,false) * 90

Language switching flag features are written in bold. Boolean OR is implemented by additional `BOOL_OR` rules. Bilingual inclusion grammar rules toggle the value of the language switching flag feature from `false` to `true`. Monolingual rules evaluate the language switching flags of their subconstituents using the `BOOL_OR` rules. The rule numbers indicate the equivalent grammar rules of Appendix A.

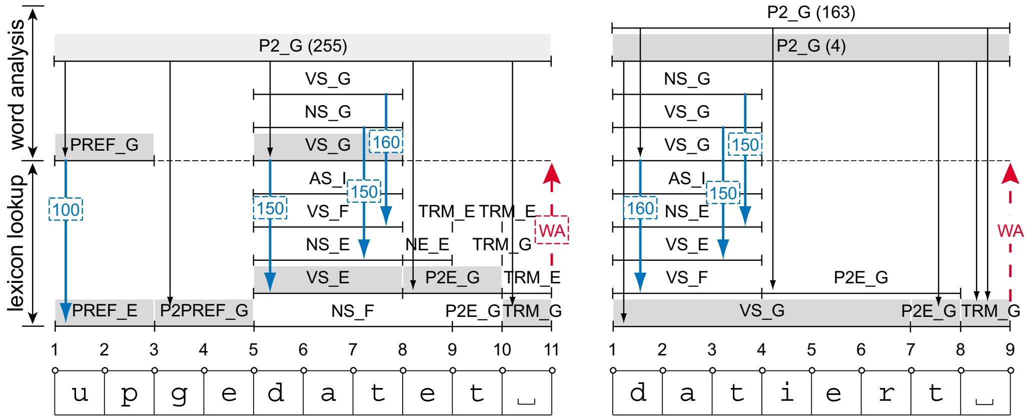

Fig. 3. Representations of the simplified charts resulting from morphological analysis of the mixed-lingual German word "upgedatet" (updated) and the monolingual German word "datiert" (dated). At the bottom the normalized input character sequence is shown. Edges are drawn without constituent feature values. If a set of edges with the same associated constituent but different feature values span the same vertices, only one of these edges is shown here. Important penalty values of edges are shown in parentheses. The "lexicon lookup" section contains edges associated with the lexemes found during lexicon lookup. The "word analysis" section contains edges associated with constituents resulting from word analysis. An arrow with bold line tagged with a penalty value denotes the application of an inclusion grammar rule. An arrow with a dashed line tagged with WA indicates a word analysis trigger event. The constituents of the final morphological parse tree are shown with grey background.

stem VS_E (cp. lexicon entry L34), an English noun stem NS_E (L19), a French verb stem VS_F (L86), or an Italian adjective stem AS_I (L168). Additionally, "datiert" can be analyzed using the German verb stem VS_G "datier" (L127), and "upgedatet" using the French noun stem NS_F "date" (L76).

"upgedatet" on the left side of Fig. 3 demonstrates how language mapping restrictions using inclusion grammar rules are applied in practice. The English and French verb stems can be both analyzed as foreign German verb stems using the inclusion rules of Appendix A.3.4. Inclusion grammar rule R84 maps the English verb stem to a German verb stem with verb class feature value v1. Inclusion grammar rule R86 includes the French verb stem using feature value v12. Another inclusion grammar rule, R85, maps the English prefix "up" to a German prefix. As the verb class feature value of the German past participle ending P2E_G "et" is v1, only the embedded English verb stem can be unified with this ending using the German word grammar rule R75. Thus, the only word constituent that can be analyzed using word grammar rules is a German past participle P2_G with two English inclusions. This is also the correct analysis of "($_G$($_E$up)ge($_E$dat)et)".

"datiert" on the right side of Fig. 3 shows how inclusion rule penalty values are used to disambiguate the correct analysis result of multiple possible results. The German verb stem VS_G "datier" together with the German past participle ending P2E_G "t" forms a monolingual German past participle P2_G with an overall penalty value of 4. Also, all multilingual variants of the stem "dat" are inserted into the chart. The French verb stem VS_F is mapped to a German verb stem with the verb class feature value v12. Using German word grammar rule R76 this stem can be unified with the German past participle ending "iert" to a German past participle with an overall penalty value of 163. As both P2_G constituents are grammatically equal, subsequent sentence analysis will choose the one with the lower penalty value. So, "($_G$datiert)" is correctly analyzed as a monolingual German word.

3.4. Mixed-lingual syntactic analysis

In the polySVOX system syntactic analysis is accomplished in two steps: a sentence analysis and a paragraph analysis. Similar to mixed-lingual word analysis separate monolingual sentence and paragraph grammars plus additional bilingual sentence and paragraph inclusion grammars are applied for mixed-lingual syntactic analysis. The result of syntactic analysis is a mixed-lingual morpho-syntactic parse tree, which describes the syntactic structure of the sentences and the morphological structure of the words. As each constituent is tagged by a suffix indicating the language, the language and the syntactic and morphological structure of foreign inclusions are described as well.

Fig. 4 demonstrates how our approach to syntactic analysis correctly analyzes "($_E$Asia welcomes ($_F$bon ami Chirac.))" This is a mixed-lingual English sentence containing a majority of French words. In this sentence it is also important to correctly analyze the French noun phrase, as this information is necessary for subsequent phonological processing, like the application of French liaison rules, and the generation of a proper prosody. We will discuss three different syntactic analysis results, which are illustrated in Fig. 4:

- S_E (350): Word analysis returns "bon" as French adjective ADJ_F, "ami" as French noun N_F, and "Chirac" as French proper noun PRN_F. These French constituents are mapped onto corresponding English constituents using the inclusion grammar rules R40,

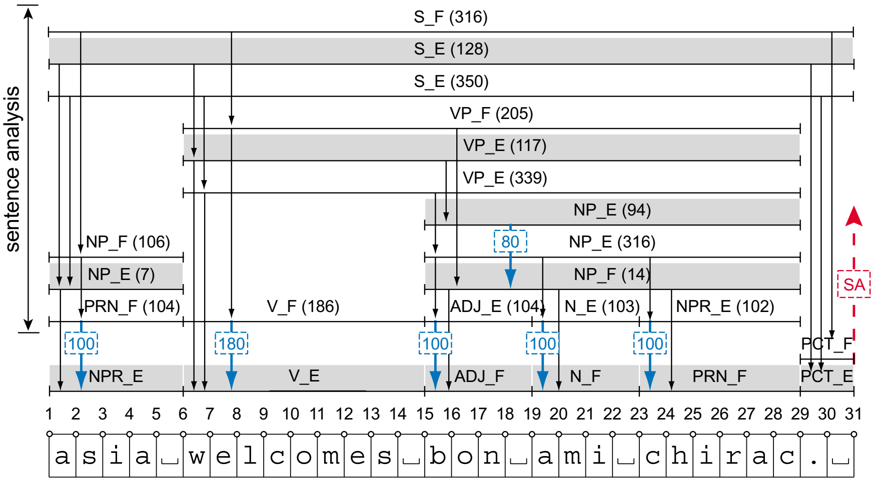

Fig. 4. Representation of the simplified chart resulting from morphological and syntactic analysis of the mixed-lingual English sentence "Asia welcomes bon ami Chirac." The bottom line of constituents comprises word constituents resulting from mixed-lingual word analysis. The "sentence analysis" section contains edges associated with constituents resulting from sentence analysis.

R41, and R42 of Appendix A.1.6. Their inclusion penalties sum to 300. Applying monolingual English sentence grammar rules these embedded French inclusions can be analyzed as an English noun phrase `NP_E` with an overall penalty value of 316. With this English noun phrase the analysis as an English sentence gets an overall penalty value of 350.

- `S_E (128)`: The French adjective, noun, and proper noun can also be analyzed as a French noun phrase `NP_F`. This French noun phrase is then mapped to an English noun phrase with an overall penalty value of 94 using inclusion grammar rule R43. In this case, the inclusion penalty is only 80. The analysis as an English sentence with this embedded French noun phrase results in an overall sentence penalty value of only 128.
- `S_F (316)`: Also, it is possible to analyze the English proper noun "Asia" and the English verb "welcomes" as foreign inclusions within a French sentence. These English constituents are mapped to French constituents using French inclusion rules R66 and R67 of Appendix A.2.4. The summed inclusion penalty is 280. As the inclusion of an English verb within a French sentence is more unlikely, the inclusion grammar rule R67 has a higher penalty value. Overall sentence penalty value of the so analyzed French sentence `S_F` is 316.

Of these three sentence analysis results the one with the lowest overall penalty is finally chosen. This is the English sentence including the complete French noun phrase `NP_F` as a foreign inclusion. The resulting morpho-syntactic parse tree contains the correct identification of the sentence base language and of the language of the foreign multi-word inclusion, the correct mixed-lingual phone sequence, and the correct syntactic structure of the foreign multi-word inclusion.

Fig. 4 shows how our approach to syntactic analysis correctly analyzes the syntactic structure of foreign inclusions. This is achieved inherently by choosing the result with minimal overall penalty, as the inclusion of larger constituent structures adds up fewer inclusion penalties to the overall penalty. The constraints set by monolingual sentence grammars and bilingual sentence inclusion grammars additionally specify how larger foreign inclusions are syntactically analyzed, and restrict thereby the number of possible solutions. The specification of different inclusion penalty values allows to distinguish between common and uncommon foreign inclusions.

If a sentence can not be analyzed using the given sentence grammar rules, an artificial parse tree is created by finding the way through the chart with minimal total penalty (cp. Traber, 1995). An additional edge penalty leads to the preference of higher constituents over the combination of lower constituents. If, e.g., no sentence constituents `S_E` or `S_F` exist in the chart of Fig. 4 after parsing, still a sentence is found as a sequence of `NP_E`, `VP_E`, and `PCT_E`.

3.5. Interlingual homographs

Interlingual homographs, as outlined in Section 2.2.1, are a major problem to language identification. Fig. 5 illustrates by means of the mixed-lingual German sentence "($_{G}$Die ($_{E}$Greatest Nation) hat die ($_{F}$Grande Nation) als tonangebende Nation abgelöst.)" (The Greatest Nation replaced the Grande Nation as leading nation), how mixed-lingual syntactic analysis of the polySVOX system disambiguates such interlingual homographs.

In this sentence the first instance of "nation" is English, the second one is French, and the final one is German. The disambiguation of these homographs alone is impossible, as they all have the same syntactic function (all are singular nouns). For correct disambiguation the language of the neighboring words of these nouns must be considered additionally. The English adjective "greatest" produces an English noun phrase `NP_E` with the English variant of

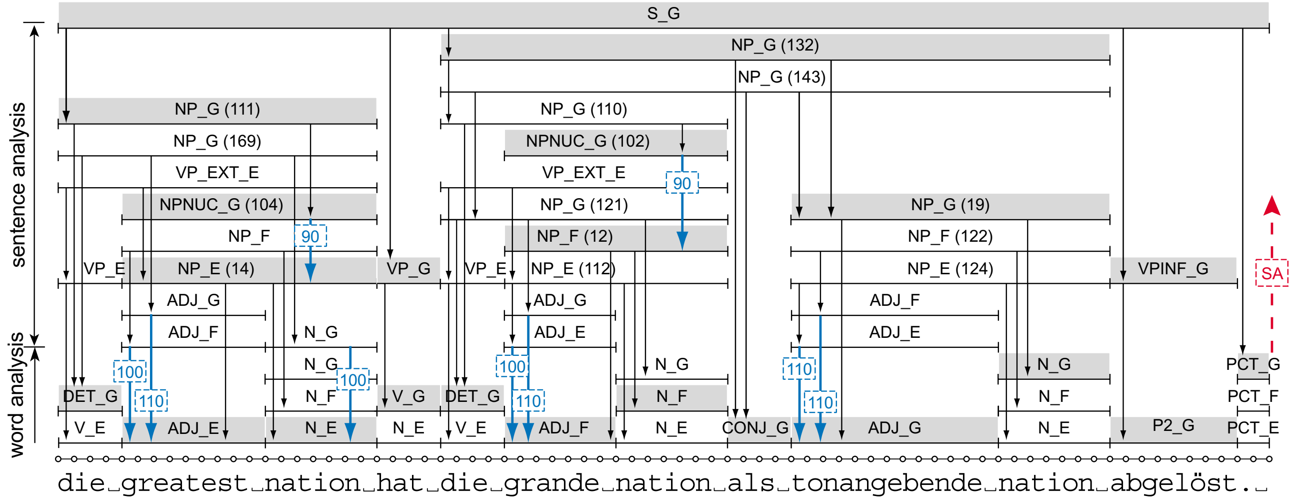

Fig. 5. Representation of the simplified chart resulting from morphological and syntactic analysis of the mixed-lingual German sentence “Die Greatest Nation hat die Grande Nation als tonangebende Nation abgelöst.”

“nation”. Likewise, the French adjective “grande” forms a French noun phrase NP_F with the French variant of “nation”, and the German adjective “tonangebende” a German noun phrase NP_G with the German variant of “nation”. The English and French noun phrases are finally included as foreign noun phrase inclusions within the German sentence.

An alternative analysis of this sentence includes the English and French adjectives as foreign adjective inclusions within German noun phrases. But, as the inclusion of a complete noun phrase is less penalized than the inclusion of a separate adjective (see, e.g., inclusion rules R91 and R92 in Appendix A.3.4), the analysis with complete foreign noun phrase inclusions is preferred. E.g., by including “greatest nation” as a foreign noun phrase the first German noun phrase (NP_G) in Fig. 5 gets an overall penalty of 111 compared to 169 if only the English adjective “greatest” would have been included as foreign inclusion. Likewise, the final German noun phrase gets an overall penalty of 132 when including the complete French noun phrase “grande nation” versus an overall penalty of 143 in case of including only the French adjective “grande”.

Fig. 5 also shows two other interlingual homographs that do not arise from loanwords: one is “hat”, which is an English noun as well as a German verb. The other one is “die”, which is an English verb as well as a German determiner. These interlingual homographs are disambiguated by syntactic means: Using the German variants of these homographs a correct German sentence can be analyzed. With the English variants no syntactically correct sentence is possible.

4. Analysis of unknown words in mixed-lingual sentences

TTS synthesis systems are expected to read any text, even if these texts contain words that are not parseable by word analysis, either as one or more morphemes are missing in the lexicon, or as they are simply misspelled. Such words are often referred to as “unknown” words. Concerning missing lexicon entries, their number can be reduced by utilization of large lexica. Nevertheless, there will always be a remainder of words (especially proper nouns) that are not covered even by very large lexica (cf. Coker et al., 1990). Concerning misspelled words, listing them in lexica is obviously impossible.

Monolingual TTS synthesis systems usually incorporate some form of a rule-based grapheme-to-phoneme mapping by which a monolingual pronunciation of any unknown word is derived in the system’s language.

In mixed-lingual sentences unknown words may additionally be full foreign words or even mixed-lingual words. However, unknown mixed-lingual words occur very seldom, as mixed-lingual words are typically built using common foreign stems, which are anyway included in a reasonably sized foreign lexicon. In order to derive the correct pronunciation of unknown foreign words, polyglot TTS synthesis systems need to comprise a mixed-lingual text analysis component that is able to identify the correct language of unknown words and provide language-dependent grapheme-to-phoneme mappings. The following mixed-lingual sentence illustrates an example of an unknown English proper noun, which is embedded in a German sentence. The unknown word is noted in italics:

“($_{\text{G}}$Er lebt in ($_{\text{E}}$British *Columbia*).)”

The correct pronunciation of this sentence requires the unknown word to be analyzed by the English grapheme-to-phoneme conversion and not, e.g., the one of the sentence’s base language.

To alleviate the complexity of mixed-lingual unknown word analysis, we took the following assumptions as a basis for the design of the polySVOX mixed-lingual grapheme-to-phoneme conversion:

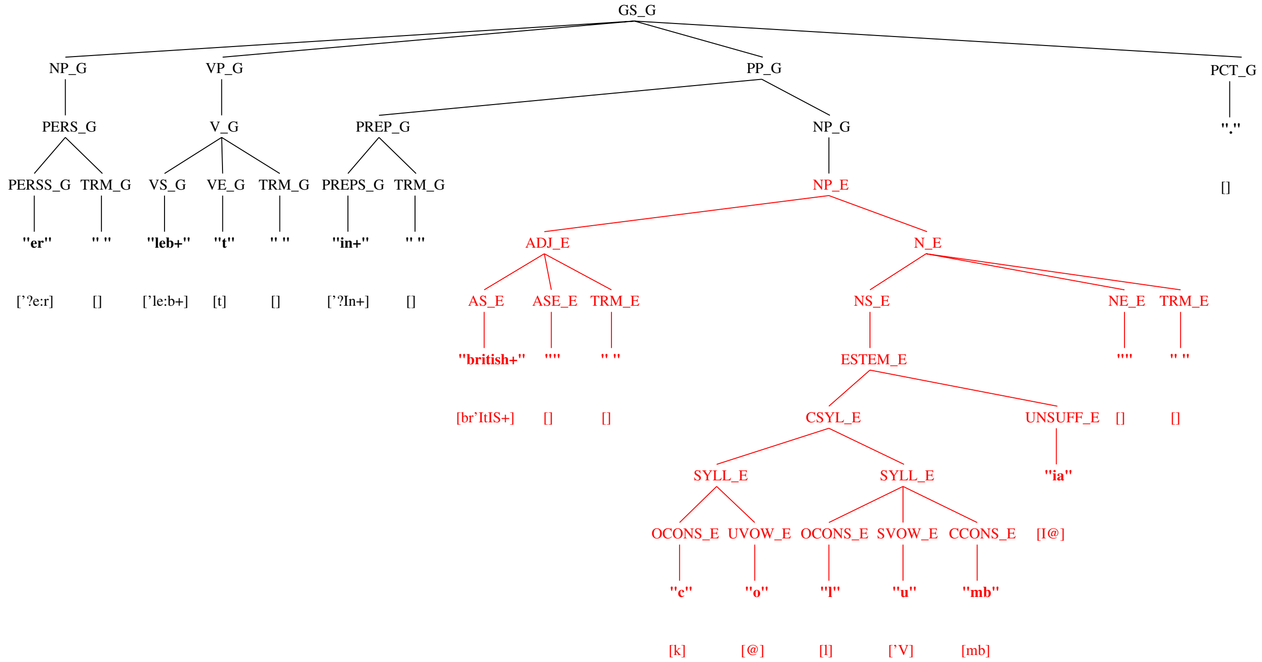

Fig. 6. Morpho-syntactic tree resulting from polySVOX morpho-syntactic analysis of the mixed-lingual German sentence "Er lebt in British *Columbia*." The English proper noun "Columbia" is unknown to the system.

- Closed word categories, like prepositions, conjunctions, determiners, or pronouns, of each language of the system are supposed to be completely listed in the respective monolingual lexica. Thus, the only possible word categories for an unknown word are the open word categories, like nouns, verbs, adjectives, or adverbs.
- An unknown word is analyzed as monolingual word in each of the system's languages in parallel using the respective monolingual grapheme-to-phoneme conversion algorithms.
- Mixed-lingual syntactic analysis finally disambiguates all available multilingual pronunciations of an unknown word by the word categories.

The polySVOX system applies for all four languages the same algorithm to unknown word analysis. This algorithm is based on chart parsing using penalty-extended DCG rules. It implements for each language a separate, monolingual *word stem analysis*, which decomposes unknown words into unknown word stems and known inflectional endings, prefixes, and suffixes, which are part of the monolingual morpheme lexica. Special word grammar rules describe the syllabic structure, word stress assignment, and pronunciation of unknown word stems. For unknown English stems, e.g., these rules define stressed versus unstressed and long versus short pronunciation of vowels, and word initial, word final, and word central pronunciation of consonant clusters. Appendix A.1.3 lists some of the English unknown stem analysis rules (rules R12 to R24).

In addition to the morpheme lexica a separate, monolingual lexicon of grapheme clusters, the so-called submorphemic lexicon, is loaded for each monolingual unknown word analysis. Each submorphemic lexicon contains all possible syllable onsets, codas, and nuclei together with their pronunciation variants of the respective language. Appendix A.1.1 shows some entries of the English submorphemic lexicon. Given specific inflectional endings and suffixes it is possible to make assumptions about the syntactic categories of unknown words and to provide appropriate pronunciations and word stress assignments. Traber (1995) describes in detail this stem analysis for unknown German words.

Fig. 6 shows the morpho-syntactic tree resulting from polySVOX morpho-syntactic analysis of the mixed-lingual example sentence "($_{\mathbb{G}}$Er lebt in ($_{\mathbb{E}}$British Columbia).)" As the English proper noun "Columbia" is unknown, grapheme-to-phoneme mappings in English, French, German, and Italian are applied to this word. Each of these monolingual grapheme-to-phoneme conversions derives pronunciations of "Columbia" for one or more word categories. Of these alternatives mixed-lingual sentence analysis selects the English noun category N_E, as this word category fits best the syntactic structure of the remaining sentence (cp. Section 3.5).

This multilingual unknown word analysis derives the correct pronunciation of most unknown monolingual words or provides at least a pronunciation sufficient for TTS synthesis purposes. For a correct pronunciation of certain unknown proper nouns, however, additional lexical entries are necessary.

5. Mixed-lingual word and sentence boundary identification

In order to correctly identify syntactic words within a graphemic input text, morphological and syntactic knowledge is necessary. Therefore, it is unreasonable to do this identification in some text preprocessing step. We better integrate identification of syntactic words into morphological and syntactic text analysis.

Fig. 7 illustrates our approach to word and sentence boundary identification with a morpho-syntactic analysis of the English sentence: “It’s in St. Mary’s St.” The correct pronunciation of this sentence [ɪts ɪn sənt meəriz striːt] requires to identify the dot in the first “St.” as part of the abbreviation and the dot in the second “St.” as a fullstop terminating the sentence. This can be achieved by syntactic means, which 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 describe the application of the main processing steps of our text analysis to this example sentence. These processing steps are also shown in Fig. 2:

Text normalization generates out of the graphemic input text or input stream a well-defined character stream. Note that also punctuation characters, the blank character, carriage return, the newline character, and other special characters are included as separate tokens. Text normalization primarily takes care that all capital letters are converted to lowercase letters, that all sequences of contiguous space characters are reduced to one space character, and that all input characters not defined in one of the language-specific sets of legal 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 lexemes of the morpheme lexicon. For each matching lexeme, a corresponding edge is inserted into the chart. These edges are shown in the “lexicon lookup” section in Fig. 7. In the morpheme lexicon the keyword ‘:WORD_END’ indicates a possible word boundary after the lexeme, as can be seen, e.g., in the lexicon entries L1, L2, L3, or L4 in Appendix A.1.1.

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, see, e.g., Appendix A.1.3. The word analysis results are inserted into the chart. These constituents are shown in the “word analysis” section in Fig. 7.

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

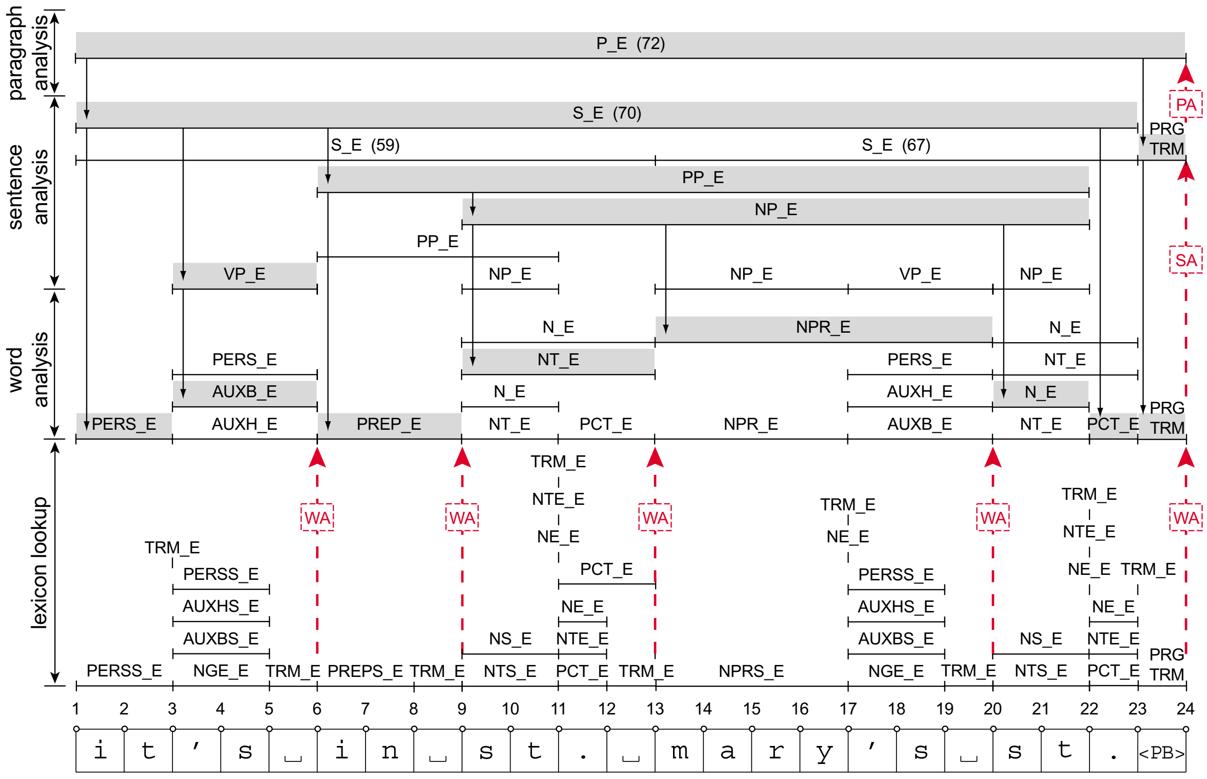

Fig. 7. Representation of the simplified chart resulting from morphological and syntactic analysis of the sentence “It’s in St. Mary’s St.”

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, e.g., in the grammar rules R2 or R1 in Appendix A.1.3. 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" in Fig. 7.

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. grammar rule R25 in Appendix A.1.4. 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.

5.1. Analysis of contracted word forms

The approach presented here allows to correctly analyze ambiguous contracted word forms. The basic idea is to include in morphological analysis beside of blank characters also empty characters as word delimiters. E.g. for English, these delimiters are listed as TRM_E in the morpheme lexicon in Appendix A.1.1 and are used in the word grammar rules in Appendix A.1.3 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 a higher penalty, cf. lexicon entry L5. Additionally, specific word categories like abbreviations can use separate empty word delimiters with a lower penalty value, e.g., lexicon entry L6. These empty word delimiters are without a ':WORD_END' tag, 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 by some examples. One example is the token sequence "'s" in the sentence in Fig. 7, which 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, L10, L33, L39, and L40 match "'s" and are inserted into the chart. For "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). For "mary's␣" the word grammar rule R5 additionally allows a morphological analysis of the complete orthographic word as possessive form of a proper noun NPR_E.

As can be verified in Fig. 7 this input sequence can 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".

Paragraph grammar rules, as shown for English in Appendix A.1.5, that define a paragraph as a sequence of sentences, prevent that 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 $(2 + 70)$.

5.2. Analysis of cross-line hyphenations

Texts may also contain line breaks and hyphens that indicate a word split at a line break. Line breaks within the input text are encoded as '<LF>' by text normalization. By default, text analysis replaces every '<LF>' by the blank symbol '␣' before lexicon lookup by applying a two-level rule (Rule T4 in Table 3 implements this default transformation). A hyphen preceding '<LF>' may indicate a word split at the line break. In order to correctly look up such words in the lexicon, text analysis must merge the two word parts again. Merging them, however, may lead to ambiguities concerning the treatment of hyphens in the following situations:

- *Words split by a hyphen at a line break*: in order to derive the correct word, the sequence "-<LF>" must be deleted from the input sequence. E.g., consider the input sequence "in-<LF>put" and the corresponding lexeme "input". In German texts, however, some rules are still in use that additionally change the orthographic form of words split at line breaks: E.g., one rule states that "ck" is split into "k-<LF>k". Thus, "Zucker" (sugar) is split into "Zuk-<LF>ker". Such graphemic modifications must also be considered by text analysis.

Table 3
Two-level rules needed for analysis of hyphens at the end of text lines

[T1]	@:-	⇒	_ ?: <LF>
[T2]	@:<LF>	⇒	?:- _
[T3]	c:k	⇒	_ @:- @:<LF>
[T4]	␣:<LF>	⇔	_

Rule T4 denotes the default replacement of <LF> by a blank. This rule is applied after application of rules T1, T2, and T3. A character pair *x*:*y* denotes an association between a lexical character *x* and a surface character *y*. *x* and *y* may also be empty characters denoted by @. ? specifies an arbitrary character. A rule of the form *x*:*y* ⇒ *L*_*R* (where _ denotes the position of the character pair in question, and *L* and *R* the left and the right context) states that the character pair *x*:*y* must not occur in any other context than *L*_*R*. A rule of the form *x*:*y* ⟺ *L*_*R* additionally specifies that in the context *L*_*R* the lexicon character *x* must not be associated with any other surface character than *y*.

- *Hyphens being part of a split compound*: in this case, only the symbol <LF> must be deleted from the input sequence. E.g., the deletion of <LF> from “n-<LF>tuple” results in the correct word “n-tuple”.
- *Hyphens indicating word ellipsis*: here, text analysis must replace the input symbol <LF> by a ␣ before lexicon lookup. This is the default behavior, if no hyphen precedes <LF>. E.g., replacing the symbol <LF> in the input sequence “in-<LF>and␣output” gives “in-␣and␣output”.

These examples show that such hyphens can not simply be lexically annotated. As during lexicon lookup the correct treatment of the hyphen is unknown yet, the polySVOX text analysis applies all three graphemic conversions in parallel on the input stream using the two-level rules in Table 3: Rule T1 optionally removes a hyphen in front of <LF>. Rule T2 denotes optional deletion of <LF> after a hyphen. Rule T3 is an example of considering special graphemic modifications like German “k-<LF>k”; it optionally replaces a “k” by a “c” in front of “-<LF>”.

Lexicon lookup is then done for each of the resulting alternative sequences. All matching lexemes from these lookups are inserted into the chart. Finally, morphological analysis parses the correct word forms as defined by the word grammar. Fig. 8 illustrates two-level rules application, lexicon lookup, and morphological analysis for the input sequence “in-<LF>put”.

5.3. 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”.

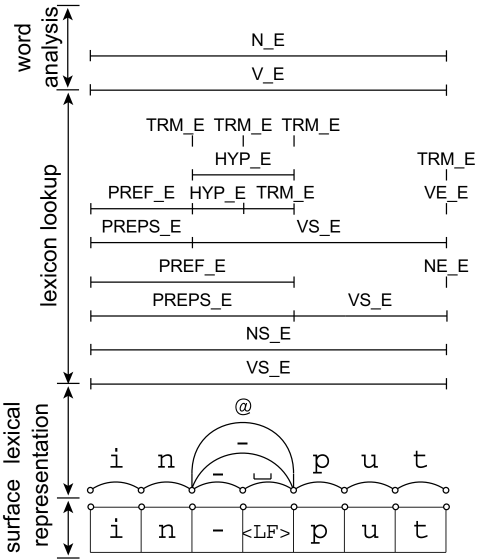

Fig. 8. Representation of the simplified chart resulting from morphological analysis of the sequence “in-<LF> put”. Two-level rule application is shown in form of a transition network over the surface input sequence. Each path from left to right through the transition network forms a possible lexical representation. @ denotes an empty transition.

To describe “in front of” as a multi-word lexeme is convenient for syntactic analysis, whereas it is irrelevant for pronunciation. For other word forms, like the adverb “in fine”, pronounced as [ɪn ˈfa͜ɪni], multi-word analysis is a necessity to disambiguate it from the preposition “in” [ɪn] followed by the adjective “fine” [fa͜ɪn]. E.g., consider the sentence “He’s in fine condition in fine.”. Using multi-word lexemes, the final “in fine” is correctly analyzed as an adverb.

5.4. Sentence end identification

The correct identification of the end of a sentence in case of ambiguous punctuation symbols is an important issue in TTS synthesis, as sentence end is an important feature in prosody generation. Therefore, numerous approaches have already been presented, including simple heuristic ones, as detecting capitalized words following periods (cp. McAllister, 1989; Dutoit, 1993; Liberman and Church, 1991), probabilistic ones (e.g., Riley, 1989), and an elaborated, morphology based approach presented by Traber (1995). All of these approaches, however, are applied in a preprocessing step and therefore lack syntactic disambiguation capabilities.

Similar to the identification of syntactic words, the identification of sentence ends also requires morphological and syntactic knowledge. In our approach we integrated sentence end identification into morphological and syntactic analysis and analyze punctuation symbols as a special form of syntactic words. The following points summarize the central ideas in sentence end identification:

- In case of *unambiguous sentence-final punctuation symbols*, sentence analysis is 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 containing the punctuation symbol are inserted into the chart. Sentence analysis is started only at the next unambiguous sentence end.
- At the *end of a paragraph*, indicated by the paragraph boundary symbol "<PB>", sentence analysis is always started.

Fig. 9 illustrates the first two situations: In case of "street.␣", as presented on the left side, word analysis returns an English noun N_E, which contains an empty noun ending NE_E and an empty word delimiter TRM_E. This noun is followed by an unambiguous sentence end

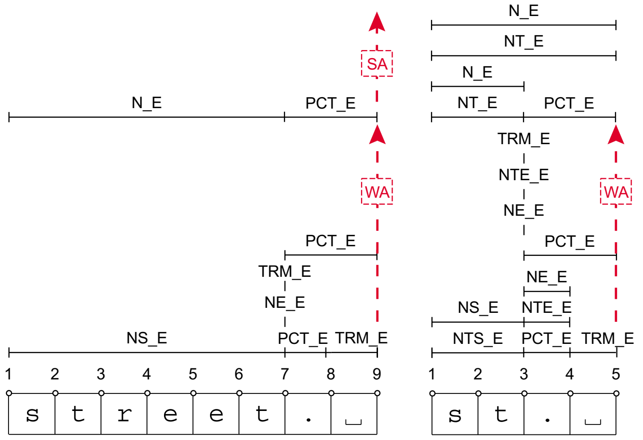

Fig. 9. For the input text “Street.” word analysis returns a noun N_E followed by an unambiguous sentence end PCT_E. Thus, sentence analysis is started at chart vertex 9. In case of the input text “St.” the period is ambiguous. It is either a punctuation symbol PCT_E, a part of a noun N_E, or a noun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

PCT_E, which spans the period and the blank character. The corresponding morpheme lexicon entries are listed in Appendix A.1.1.

In contrast to this, the right side of Fig. 9 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 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 syntactic analysis. Fig. 7 illustrates such a disambiguation. As sentence end decision in chart vertex 13 is ambiguous (two word category edges without ‘:SENT_END’ keyword end in this vertex), sentence analysis is started only after 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 5.1.

5.5. 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 Fig. 7, the resulting input sequence is "it’sinst.mary’sst.". Fig. 10 illustrates a simplified chart from morphological and syntactic analysis of this sequence.

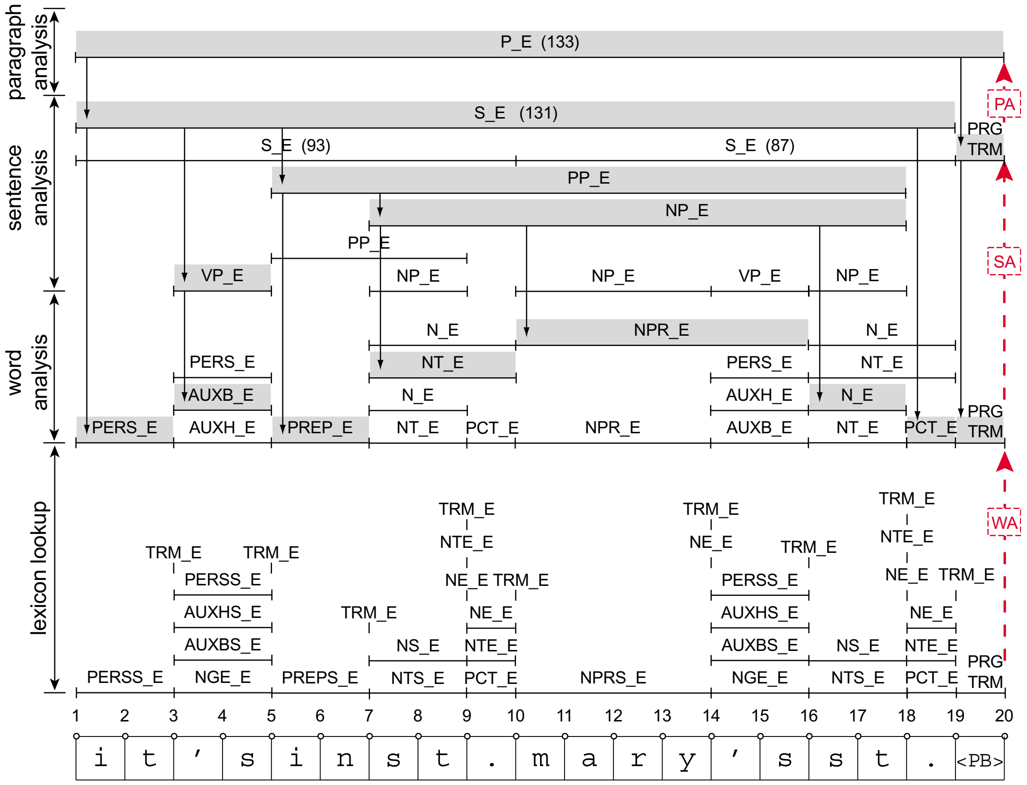

Fig. 10. Representation of the simplified chart resulting from morphological and syntactic analysis of the sentence “it’sinst.mary’sst.”

It is easy to verify that the syntactic parse tree and the word constituents in the “word”, “sentence”, and “paragraph analysis” sections of Fig. 10 are exactly the same as the ones of Fig. 7. Thus, regardless of the presence of word separation characters text analysis derives the same phonological representation. Using the existing grammars, however, equal syntactic parse trees can not be guaranteed for an arbitrary English sentence.

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. Sproat et al., 1996). As an example, consider the Chinese character sequence 研究生 which forms a complete noun in the sentence

研究生	一般	年龄	大
yan2-jiu1-sheng1	yi1-ban1	nian2-ling2	da4
’Graduate student’	’generally’	’age’	’old’

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

他	在	研究	生 命起源
ta1	zai4	**yan2-jiu1**	**sheng1**-ming4-qi3-yuan2
’He’	’doing’	’research’	’the origin of life’

As long as such character sequences are only lexically ambiguous, the text analysis approach 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, and foreign English inclusions. Such sentences can be analyzed using the mixed-lingual text analysis approach presented in Section 3.

6. Experiments

6.1. Mixed-lingual sentence Corpus

In order to evaluate language identification accuracy of our approach we collected a test corpus of 612 mixed-lingual sentences. The majority of these sentences comes from Swiss newspapers in German (“Neue Zürcher Zeitung”, “Blick”, and “20 Minuten”) and in French (“Le Matin” and “Tribune de Genève”). Additionally, the example sentences of Table 1 were included in the corpus. Some of the English sentences, finally, were taken from articles found on the internet.

This test corpus contains 36 English, 35 French, and 541 German mixed-lingual sentences, which together comprise 8511 words. Punctuation symbols are not counted. Table 4 shows detailed word number statistics of the corpus. 1903 (22.4%) of all words are foreign inclusions. These inclusions consist of 1593 full foreign words and 310 mixed-lingual words.

Table 4
Number of words of the mixed-lingual sentence corpus

Sent.	Word	Base Lang.	Full Incl.	Mixed Incl.	Sum
ENG	ENG	410 (30)	–	3 (0)	413 (33)
	FRE	–	58 (3)	0	58 (3)
	GER	–	12 (0)	0	12 (0)
	ITA	–	28 (0)	0	28 (0)
	Sum	410 (30)	98 (3)	3 (0)	511 (33)
FRE	ENG	–	109 (13)	0	109 (13)
	FRE	412 (34)	–	0	412 (34)
	GER	–	16 (7)	0	16 (7)
	ITA	–	13 (1)	0	13 (1)
	Sum	412 (34)	138 (21)	0 (0)	550 (55)
GER	ENG	–	861 (117)	0	861 (117)
	FRE	–	420 (53)	0	420 (53)
	GER	5786 (327)	–	307 (28)	6093 (355)
	ITA	–	76 (17)	0	76 (17)
	Sum	5786 (327)	1357 (187)	307 (28)	7450 (542)
Sum		6608 (391)	1593 (211)	310 (28)	8511 (630)

This corpus contains 36 English, 35 French, and 541 German sentences with English, French, German, and Italian inclusions. The numbers are grouped row-wise according to the base language of the sentence (`Sent.`) and the language of the words (`Word`). The columns contain the numbers of monolingual words of the sentence base language (`Base Lang.`), of full foreign inclusions (`Full Incl.`), and of mixed-lingual words (`Mixed Incl.`) For every category, the number of unknown words is given in parentheses beside the number of all words.

The complete, manually tagged mixed-lingual corpus together with the analysis results of polySVOX is available online on our web site <http://www.tik.ee.ethz.ch/%7Espr/SVOX/polysvoxdemo/>.

6.2. Sentence base language identification

For this evaluation we analyzed all sentences of the mixed-lingual corpus separately without any context sentences. Still, the base language of all 612 sentences but one were correctly identified. This one German mixed-lingual sentence, “Weltcup-Leader Simon Schoch out” – a heading in a Swiss German newspaper, was analyzed as an English mixed-lingual sentence.

As “Schoch” is an unknown word in our system and “Simon” is an English as well as a German forename, the analysis as an English sentence, “($_{E}$($_{G}$Welt)cup-Leader Simon Schoch out)”, with only one foreign inclusion gets lower penalty than the analysis as a German sentence, “($_{G}$Welt($_{E}$cup)-($_{E}$Leader) Simon Schoch ($_{E}$out))”, containing three foreign inclusions.

However, as this sentence is the heading of a German article, paragraph analysis (cp. Section 5) would finally chose the correct German reading of this sentence.

6.3. Language identification of words

Table 5 shows the word language confusion matrix grouped by the sentence base languages. From this table one can easily verify that the language of 8314 of all 8511

Table 5
Word language confusion matrix of English, French, and German mixed-lingual sentences

	ENG				FRE				GER			
	ENG	FRE	GER	ITA	ENG	FRE	GER	ITA	ENG	FRE	GER	ITA
ENG	412	1	0	0	105	3	1	0	801	2	58	0
FRE	3	55	0	0	2	409	1	0	17	379	24	0
GER	0	0	12	0	0	0	16	0	54	19	6017	3
ITA	0	0	0	28	0	0	0	13	0	0	9	67

The rows show the reference language of a word, the columns show the language assigned to a word by polySVOX.

Table 6
Precision and recall results of word language identification given in percent for English, French, and German mixed-lingual sentences separately and for the mixed-lingual corpus in total

Word language	ENG		FRE		GER		Total	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
ENG	99.3	99.8	98.1	96.3	91.9	93.0	95.3	94.6
FRE	98.2	94.8	99.3	99.3	94.8	90.2	97.1	94.7
GER	100	100	88.9	100	98.5	98.8	98.5	98.8
ITA	100	100	100	100	95.7	88.2	97.3	92.3
Base Lang.	99.3	99.8	99.3	99.3	98.5	98.8	*98.6*	*98.8*
Foreign Lang.	99.0	96.9	97.1	97.1	95.6	94.9	*95.8*	*95.1*

At the bottom precision and recall results for word language identification of full foreign inclusions and of sentence base language words are shown.

words (97.7%) was correctly identified. However, this number is not really representative, as it largely depends on the rate of sentence base language words. Therefore we present separate numbers for word language identification of sentence base language words and of foreign inclusions.

Table 6 gives detailed word language identification results in terms of precision and recall for sentence base language words and full foreign inclusions grouped by the sentence base languages and in total for the whole mixed-lingual corpus. In total, the language of full foreign inclusions was identified with a balanced *F-score of 95.5%*. The language of sentence base language words was identified with an *F-score of 98.7%*.

F-scores for language identification of foreign inclusions within the English and French sentences sets separately are even higher, i.e., 98.0% and 97.1% resp. The language of words of the sentence base language was identified with *F*-scores of 99.6% for English and 99.3% for French mixed-lingual sentences. These results verify the authors' experience that foreign inclusions in English or French sentences are easier to identify than in German sentences. However, the results for English and French sentences alone must be read with some care as the number of English and French sentences is rather limited (i.e., 36 English and 35 French sentences).

6.4. Language identification of inclusions in mixed-lingual words

Mixed-lingual words are mainly found in German sentences. Table 4 shows that 307 of 1664 foreign inclusion (18.4%) in the German sentences are mixed-lingual words. In the English test sentences only three mixed-lingual words were found. These are actually full foreign inclusions with English plural or s-Genitive suffixes, i.e., "(${}_{I}$cappucino)s", "(${}_{I}$lasagna)s", and "(${}_{F}$cuisine)'s". In the French test sentences no mixed-lingual words exist.

All English mixed-lingual words and 260 of the 307 German mixed-lingual words, *84.8%* of the mixed-lingual words in total, are correctly analyzed. The erroneous analyses of German mixed-lingual words originate from four main sources:

(1) *Analysis as a full foreign word:* the polySVOX analysis prefers full, foreign monolingual noun inclusions that syntactically agree over mixed-lingual inclusions. Therefore, a mixed-lingual word that contains a German word part which is ambiguous to a word of the inclusion language is analyzed as a monolingual foreign word (cp. the examples in Table 7). This is the most common error of mixed-lingual word analysis.
(2) *Analysis as monolingual base language word:* as our analysis prefers monolingual words of the sentence base language over any foreign inclusion, every mixed-lingual word containing a foreign inclusion that is ambiguous to a base language morpheme is always analyzed as monolingual word in the sentence base language.
(3) *Ambiguous base language and foreign morphemes:* the polySVOX analysis prefers ambiguous base language morphemes that accord with the morphological rules over foreign morphemes. Exceptions are incorrectly analyzed, as shown in Table 7.
(4) *Ambiguous foreign morphemes of multiple languages:* ambiguous English and French morphemes are very common in German mixed-lingual words. As English

Table 7
Examples of incorrectly analyzed mixed-lingual words of the test corpus

	Reference analysis	Analysis by polySVOX
1	(${}_{G}$(${}_{F}$Gourmet)-(${}_{E}$Festival)) (${}_{G}$(${}_{E}$Bluetooth)-System) (${}_{G}$Index(${}_{F}$fonds)) (${}_{G}$Auto(${}_{E}$rowdy)) (${}_{G}$(${}_{E}$Lifestyle)-Hotel)	(${}_{F}$Gourmet-Festival) (${}_{E}$Bluetooth-System) (${}_{F}$Indexfonds) (${}_{E}$Autorowdy) (${}_{E}$Lifestyle-Hotel)
2	(${}_{G}$(${}_{E}$Hacker)attacken)	(${}_{G}$Hackerattacken)
3	(${}_{G}$Firmen-(${}_{E}$Website)) (${}_{G}$(${}_{E}$All-In-One)-Navigationsgerät)	(${}_{G}$Firmen-Web(${}_{E}$site)) (${}_{G}$All-In-(${}_{E}$One)-Navigationsgerät)
4	(${}_{G}$Feinschmecker-(${}_{F}$Restaurants)) (${}_{G}$(${}_{F}$Amateur)truppe) (${}_{G}$Viertel(${}_{F}$final))	(${}_{G}$Feinschmecker-(${}_{E}$Restaurants)) (${}_{G}$(${}_{E}$Amateur)truppe) (${}_{G}$Viertel(${}_{E}$final))

The number in the left column indicates the description number in Section 6.4.

morphemes are more often used in our cultural setting the polySVOX analysis prefers English morphemes over ambiguous French morphemes. Table 7 shows exceptions, in which the French pronunciation is more common. These are incorrectly analyzed using the English pronunciation.

6.5. Language identification of unknown words

Two hundred and thirty nine (12.6%) of the foreign inclusions and 391 (5.9%) of the sentence base language words are unknown (cp. Table 4). This means, they must be analyzed using the procedure described in Section 4. Note, that our strict morphological analysis of words reduces the number of unknown words already considerably when compared to full form lexicon lookup.

Table 8 shows the word language identification results of unknown words and the inclusion language identification results of unknown mixed-lingual words. The polySVOX system assigns the correct language to 95.1% of the unknown words in the sentence base language, and to *72.5% of the unknown foreign inclusions*.

As our system analyzes unknown words in a monolingual fashion (cp. Section 4), unknown mixed-lingual words are analyzed incorrectly by default. However, polySVOX analyzes unknown mixed-lingual hyphenated compounds as a sequence of hyphenated monolingual words. Thus, polySVOX is still able to identify the correct inclusion languages within 11 of 28 (39.3%) unknown mixed-lingual words. In total, the language of *85.1% of 630 unknown words* was correctly identified.

Table 8
Results of word language identification of unknown words

	Unknown words	Correct	Correct (%)
Base Lang.	391	372	95.1
Full Incl.	211	153	72.5
Mixed Incl.	28	11	39.3
Total	630	536	85.1

The rows show the total number of words, and the number and percentage of correctly identified words of unknown words of the sentence base language (`Base Lang.`), of unknown full foreign inclusions (`Full Incl.`), and of unknown mixed-lingual words (`Mixed Incl.`).

7. Conclusion

In this article, we have presented a new and accurate analyzer for mixed-lingual texts, which maintains a strict separation of the linguistic databases for each language. This kind of morpho-syntactic analyzer meets all requirements of a polyglot TTS synthesis system that has to pronounce mixed-lingual text in a way that the origin of foreign inclusions can be heard, i.e., with correct language-specific pronunciation and prosody. Each inclusion, even if it is a part of a word only, is pronounced according to its originating language.

A mixed-lingual text analysis component of a TTS system is often confronted with ambiguous word and sentence boundaries. This 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.

An evaluation of this analyzer using a mixed-lingual sentence corpus comprising 8511 words in total showed that 98.7% of 6608 base language words, 95.5% of 1593 foreign word inclusions, and the foreign inclusions in 84.8% of 310 mixed-lingual words were correctly analyzed. Also, for 85.1% of 630 unknown words a correct analysis was found.

With our approach to morpho-syntactic text analysis we achieve both, precise language identification and accurate structure determination. Such polyglot TTS synthesis systems are especially necessary in multilingual countries, like Switzerland, where people generally speak several languages fluently.

Acknowledgements

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.

Appendix A. Grammars and lexica

A grammar is a collection of grammar rules. Each grammar rule consists of a head, which denotes a constituent, the production symbol '`==>`', and a body, which denotes a list of subconstituents.

```
headcons==>[{subcons}]*[penalty] [{keywords}]
```

An empty subconstituent list denotes the empty production. The body is terminated by the '*' symbol. A grammar rule can optionally be followed by an integer penalty value. If this penalty value is missing, a default value of 1 is assumed. These penalty values are added during rule application in the parser and are used to select the optimal solution out of a number of ambiguous solutions.

The keywords ':INV', ':WORD_END', ':SENT_END', and ':PARA_END' may optionally be set after the penalty value or the '*' sign. ':INV' makes the corresponding node of a rule invisible in the resulting syntax tree. ':WORD_END', ':SENT_END', and ':PARA_END' are used for the identification of syntactic word, sentence, and paragraph boundaries, respectively, as explained in Section 5.

Each constituent is composed of a constituent identifier and a list of feature terms associated with the constituent. Language-specific constituent identifiers are terminated with '_X', where 'X' specifies the language. Feature terms can be atoms or variables. Variables start with a '?' followed by an identifier. The variable '?' itself is the "anonymous variable", which is usually applied as a "don't care" marker. Atoms are constants, whose identifiers must not start with a '?'. Term unification operates on all variables with identical identifiers within one grammar rule.

A lexicon is a collection of so-called preterminal constituents together with their associated terminal elements (i.e., the individual words or morphemes). Each lexicon entry consists of a constituent followed by the graphemic and phonemic representation of the terminal element. Analogously to grammar rules, a lexicon entry can optionally be followed by a penalty value and keywords.

```
cons graphem_repr phonem_repr [penalty] [{keywords}]
```

The following subsections list all English, French, German and Italian grammar rules and lexicon entries that are necessary to understand the examples given in this article. The grammars as well as the lexica presented here are by far not complete. The complete, quadrilingual set of all grammars and lexica of the polySVOX system comprises currently about 2800 grammar rules and about 21 500 lexicon entries.

A.1. English lexicon and grammars

A.1.1. English lexicon

```
[L1]   PRGTRM ()                          "<PB>"  ""  0          :WORD_END
[L2]   PCT_E (f,s)                        "."     ""             :WORD_END
[L3]   PCT_E (f,s)                        ". "    ""             :WORD_END
[L4]   TRM_E (?)                          " "     ""  0          :WORD_END
[L5]   TRM_E (?)                          ""      ""  100
[L6]   TRM_E (abbr)                       ""      ""
[L7]   HYP_E ()                           "-"     ""
[L8]   HYP_E ()                           "- "    ""             :WORD_END

[L9]   PERSS_E (sg,p3,n,s)                            "it"       "'It"
[L10]  PERSS_E (pl,pl,n,o)                            "'s"       "z"
[L11]  PREPS_E ()                                     "in"       "'In"
[L12]  PREF_E ()                                      "in+"      "'In-+"
[L13]  PREF_E ()                                      "up+"      "'Vp-+"

[L14]  NTS_E (ntcll)                                  "saint"    "s'@nt+"
[L15]  NTS_E (abbr)                                   "st"       "s'@nt+"
[L16]  NTE_E (ntcll)                                  ""         ""
[L17]  NTE_E (abbr)                                   "."        ""
[L18]  NTE_E (abbr)                                   ""         ""

[L19]  NS_E (ncl7,sgenl,n)                            "dat+"     "d'e_It+"
[L20]  NS_E (ncll,sgenl,n)                            "hat+"     "h'qt+"
[L21]  NS_E (ncll,sgenl,n)                            "input+"   "'InpUt+"
[L22]  NS_E (ncll,sgenl,n)                            "nation+"  "n'e_IS(@)n+"
[L23]  NS_E (ncll,sgenl,n)                            "street+"  ”str’i:t+”
[L24]  NS_E (abbr,nosgen,n)                           "st"       "str'i:t+"
[L25]  NPRS_E (ncll,sgenl,n)                          "asia+"    "'e_IS@+"
[L26]  NPRS_E (ncll,sgenl,f)                          "mary+"    "m'e_@ri+"
[L27]  NE_E (ncll,sg)                                 ""         ""
```

```
[L28]  NE_E (ncl1,pl)                              "s"        "s"
[L29]  NE_E (ncl7,sg)                              "e"        ""
[L30]  NE_E (ncl7,pl)                              "es"       "s"
[L31]  NE_E (abbr,sg)                              ""         ""
[L32]  NE_E (abbr,sg)                              "."        ""
[L33]  NGE_E (sgen1,sg)                            "’s"       "z"

[L34]  VS_E (emutel,pres)                          "dat+"     "d’e_It+"
[L35]  VS_E (emutel,pres)                          "di+"      "d’a_I+"
[L36]  VS_E (s,pres)                               "input+"   "’InpUt+"
[L37]  VS_E (s,pres)                               "put+"     "p’Ut+"
[L38]  VS_E (emutel,pres)                          "welcom+"  "w’elk@m+"
[L39]  AUXBS_E (sg,p3,ind,pres,yes)                "’s"       "z"
[L40]  AUXHS_E (sg,p3,ind,pres,yes)                "’s"       "z"
[L41]  VE_E (emutel,pres,ind,pres,sg,p1)           "e"        ""
[L42]  VE_E (emutel,pres,ind,pres,sg,p2)           "e"        ""
[L43]  VE_E (emutel,pres,ind,pres,sg,p3)           "es"       "z"
[L44]  AS_E (as1)                                  "great"    "gr’e_It"
[L45]  ASE_E (as1,pos)                             ""         ""
[L46]  ASE_E (as1,comp)                            "er"       "@(r)"
[L47]  ASE_E (as1,sup)                             "est"      "@st"
```

A.1.2. English submorphemic lexicon

```
[L48]  OCONS_E (s,?)                               "b"        "b"
[L49]  OCONS_E (s,?)                               "c"        "k"
[L50]  OCONS_E (s,?)                               "l"        "l"
[L51]  OCONS_E (s,?)                               "m"        "m"
[L52]  CCONS_E (s,?)                               "b"        "b"
[L53]  CCONS_E (s,?)                               "m"        "m"
[L54]  CCONS_E (m,nf)                              "mb"       "mb"
[L55]  CCONS_E (s,f)                               "mb"       "m"
[L56]  SVOW_E (ln,?)                               "a"        "’e_I"
[L57]  SVOW_E (sh,?)                               "a"        "’q"
[L58]  SVOW_E (ln,?)                               "i"        "’a_I"
[L59]  SVOW_E (sh,?)                               "i"        "’I"
[L60]  SVOW_E (ln,?)                               "o"        "’@_U"
[L61]  SVOW_E (sh,?)                               "o"        "’Q"
[L62]  SVOW_E (ln,?)                               "u"        "j’u:"
[L63]  SVOW_E (sh,?)                               "u"        "’V"
[L64]  UVOW_E ()                                   "a"        "@"
[L65]  UVOW_E ()                                   "i"        "I"
[L66]  UVOW_E ()                                   "o"        "@"
[L67]  UVOW_E ()                                   "u"        "j@"
[L68]  UNSUFF_E (n,p,nf)                           "ia"       "I@"
```

A.1.3. English word grammar

```
[R1]   PRGTRM ()                                   ==>        PRGTRM () * :SENT_END
[R2]   PCT_E (?F,?T)                               ==>        PCT_E (?F,?T) * :SENT_END
[R3]   N_E (?N,?G,?SG)                             ==>        NOUN_E (?N,?G,?SG,?NCL)
                                                              NGE_OPT_E (?SG,?N)
                                                              TRM_E (?NCL) *
[R4]   NOUN_E (?N,?G,?SG,?NCL)                     ==>        NS_E (?NCL,?SG,?G)
                                                              NE_E (?NCL,?N) * :INV
```

```
[R5]   NPR_E (?N,?G,?SG)                              ==>        NPRS_E (?NCL,?SG,?G)
                                                                 NE_E (?NCL,?N)
                                                                 NGE_OPT_E (?SG,?N)
                                                                 TRM_E (?NCL) *
[R6]   NGE_OPT_E (?SG,?N)                             ==>        * 0 :INV
[R7]   NGE_OPT_E (?SG,?N)                             ==>        NGE_E (?SG,?N) * 0 :INV
[R8]   NT_E ()                                        ==>        NTS_E (?NTCL)
                                                                 NTE_E (?NTCL)
                                                                 TRM_E (?NTCL) *
[R9]   AUXB_E (?N,?P,?M,?T,?POS)                      ==>        AUXBS_E (?N,?P,?M,?T,?POS)
                                                                 TRM_E (std) *
[R10]  AUXH_E (?N,?P,?M,?T,?POS)                      ==>        AUXHS_E (?N,?P,?M,?T,?POS)
                                                                 TRM_E (std) *
[R11]  PERS_E (?NR,?P,?G,?C)                          ==>        PERSS_E (?NR,?P,?G,?C)
                                                                 TRM_E (std) * 0

[R12]  NS_E (ncl1,?,?)                                ==>        ESTEM_E (n) *
[R13]  ESTEM_E (n)                                    ==>        CSYL_E (?,m,sh,r,i,nf)
                                                                 UNSUFF_E (n,p,nf) *
[R14]  CSYL_E (?O,?C,?PR,r,?IP,?FP)                   ==>        SYLL_E (?O,n,ln,u,?IP,nf)
                                                                 SYLL_E (s,?C,?PR,s,ni,?FP) *
[R15]  SYLL_E (?O,?C,?PR,s,?IP,?FP)                   ==>        ONSCL_E (?O,?IP)
                                                                 SVOW_E (?PR,?FP)
                                                                 CODCL_E (?C,?FP) *
[R16]  SYLL_E (?O,?C,?PR,u,?IP,?FP)                   ==>        ONSCL_E (?O,?IP)
                                                                 UVOW_E ()
                                                                 CODCL_E (?C,?FP)*
[R17]  ONSCL_E (n,?)                                  ==>        * 1 :INV
[R18]  ONSCL_E (s,?P)                                 ==>        OCONS_E (s,?P) * 1 :INV
[R19]  ONSCL_E (d,?P)                                 ==>        OCONS_E (d,?P) * 1 :INV
[R20]  ONSCL_E (m,?P)                                 ==>        OCONS_E (m,?P) * 2 :INV
[R21]  CODCL_E (n,?)                                  ==>        * 0 :INV
[R22]  CODCL_E (s,?P)                                 ==>        CCONS_E (s,?P) * 2 :INV
[R23]  CODCL_E (d,?P)                                 ==>        CCONS_E (d,?P) * 2 :INV
[R24]  CODCL_E (m,?P)                                 ==>        CCONS_E (m,?P) * 2 :INV
```

A.1.4. English sentence grammar

```
[R25]  PRGTRM ()                                      ==>        PRGTRM () * :PARA_END
[R26]  S_E (?T)                                       ==>        PERS_E (?N,?P,?,s)
                                                                 VP_E (ind,?T,?N,?P,?,fin)
                                                                 PP_E ()
                                                                 PCT_E (f,s) *
[R27]  S_E (?T)                                       ==>        NP_E (?N,?G)
                                                                 VP_E (ind,?T,?N,?P,?,fin)
                                                                 NP_E (?,?)
                                                                 PCT_E (f,s) *
[R28]  VP_E (inf,?T,?N,?P,?,?)                        ==>        AUXB_E (?N,?P,inf,?T,pos) *
[R29]  PP_E ()                                        ==>        PREP_E (?)
                                                                 NP_E (?,?) *
[R30]  NP_E (?N,?G)                                   ==>        NPRP_E (?,?)
                                                                 N_REP_E (?N,?G) *
[R31]  N_REP_E (?N,?G)                                ==>        N_E (?N,?G,?) * :INV
[R32]  N_REP_E (?N,?G)                                ==>        N_E (?,?,?)
                                                                 N_REP_E (?N,?G) * :INV
[R33]  NPRP_E (?N,?G)                                 ==>        NT_E (?)
```

```
                                                    NPR_REP_E (?N,?G) * :INV
[R34]  NPR_REP_E (?N,?G)                   ==>      NPR_E (?N,?G,?) * :INV
[R35]  NPR_REP_E (?N,?G)                   ==>      NPR_E (?,?,?)
                                                    NPR_REP_E (?N,?G) * :INV
```

A.1.5. English paragraph grammar

```
[R36]  P_E ()                              ==>      S_REP_E () PRGTRM () *
[R37]  S_REP_E ()                          ==>      S_E (?) * :INV
[R38]  S_REP_E ()                          ==>      S_E (?)
                                                    S_REP_E () * 5 :INV
```

A.1.6. English inclusion grammars

```
[R39]  NOUN_E (?NR,?,?,?)                  ==>      NOUN_F (?NR,?) * 150
[R40]  N_E (?NR,?,?)                       ==>      N_F (?NR,?) * 100
[R41]  NPR_E (?,?,?)                       ==>      PRN_F () * 100
[R42]  ADJ_E (?)                           ==>      ADJ_F (?,?,?) * 100
[R43]  NP_E (?NR,?G)                       ==>      NP_F (?NR,?,?G) * 80
[R44]  NP_E (?NR,?)                        ==>      NP_F (?NR,?,?) * 90
[R45]  N_E (?,?,?)                         ==>      NP_F (?,?,?) * 90
[R46]  NOUN_E (?NR,?,?,?)                  ==>      NOUN_G (?,?NR,?) * 120
[R47]  N_E (?NR,?,?)                       ==>      N_G (?NR,?,?) * 100
[R48]  NPR_E (?,?,?)                       ==>      PRN_G (?,?,?) * 100
[R49]  NPR_E (?,?,?)                       ==>      NP_G (?,?,?,?,?) * 110
[R50]  ADJ_E (?)                           ==>      ADJ_G (?,?,?,?,?) * 110
[R51]  NP_E (?,?)                          ==>      NP_G (?,?,?,?,?) * 90
[R52]  N_E (?,?,?)                         ==>      NP_G (?,?,?,?,?) * 90
```

A.2. French lexicon and grammars

A.2.1. French lexicon

```
[L69]   PRGTRM ()                        "<PB>"  ""  0           :WORD_END
[L70]   PCT_F (f,s)                      "."     ""              :WORD_END
[L71]   PCT_F (f,s)                      ". "    ""              :WORD_END
[L72]   TRM_F (?)                        " "     ""  0           :WORD_END
[L73]   TRM_F (?)                        ""      ""  100
[L74]   TRM_F (abbr)                     ""      ""  1

[L75]   NS_F (sgml,sgfl,plml,plfl)                   "ami+"      "ami+"
[L76]   NS_F (non,sgf2,non,plfl)                     "date+"     "dat(@)+"
[L77]   NS_F (non,sgf2,non,plfl)                     "femme+"    "fam+"
[L78]   NS_F (sgml,non,plml,non)                     "film+"     "film+"
[L79]   NS_F (non,sgf2,non,plfl)                     "nation+"   "nasj~o+"
[L80]   PRNS_F ()                                    "chirac+"   "SiRak+"
[L81]   NESG_F (m,sgml)                              ""          ""
[L82]   NESG_F (f,sgfl)                              "e"         ""
[L83]   NESG_F (f,sgf2)                              ""          ""
[L84]   NEPL_F (m,plml)                              "s"         ""
[L85]   NEPL_F (f,plfl)                              "s"         ""

[L86]   VS_F (gl,sclla,nonrefl,?,non)                "dat+"      "dat+"
[L87]   VE_F (gl,sclla,ind,pres,sg,persl)            "e"         "(@)"
[L88]   VE_F (gl,sclla,ind,pres,sg,pers2)            "es"        "(@)"
[L89]   VE_F (gl,sclla,ind,pres,sg,pers3)            "e"         "(@)"
[L90]   VE_F (gl,sclla,ind,pres,pl,persl)            "ons"       "~o"
```

```
[L91]   VE_F (g1,sc11a,ind,pres,pl,pers2)        "ez"       "e"
[L92]   VE_F (g1,sc11a,ind,pres,pl,pers3)        "ent"      "(@)"

[L93]   AS_F (v,sgm1,sgf7,plm1,plf1,non)         "bon+"     "b~o+"
[L94]   AS_F (n,sgm1,sgf1,plm1,plf1,adv1)        "fatal+"   "fatal+"
[L95]   AS_F (a,sgm1,sgf3,plm1,plf1,non)         "grand+"   "gR~a+"
[L96]   AS_F (n,sgm1,sgf1,plm1,plf1,non)         "noir+"    "nwaR+"
[L97]   AE_F (sg,m,sgm1)                         ""         ""
[L98]   AE_F (sg,f,sgf1)                         "e"        "(@)"
[L99]   AE_F (sg,f,sgf3)                         "e"        "d(@)"
[L100]  AE_F (sg,f,sgf7)                         "ne"       "n(@)"
[L101]  AE_F (pl,m,plm1)                         "s"        ""
[L102]  AE_F (pl,f,plf1)                         "s"        ""
```

A.2.2. French word grammar

```
[R53]   PRGTRM ()                ==>      PRGTRM () * :SENT_END
[R54]   PCT_F (?F,?T)            ==>      PCT_F (?F,?T) * :SENT_END
[R55]   N_F (?NR,?G)             ==>      NOUN_F (?NR,?G)
                                          TRM_F (?) *
[R56]   NOUN_F (sg,?G)           ==>      NS_F (?SGM,?,?,?)
                                          NESG_F (?G,?SGM) * :INV
[R57]   NOUN_F (sg,?G)           ==>      NS_F (?,?SGF,?,?)
                                          NESG_F (?G,?SGF) * :INV
[R58]   NOUN_F (pl,?G)           ==>      NS_F (?SGM,?,?PLM,?)
                                          NESG_F (?G,?SGM)
                                          NEPL_F (?G,?PLM) * :INV
[R59]   NOUN_F (pl,?G)           ==>      NS_F (?,?SGF,?,?PLF)
                                          NESG_F (?G,?SGF)
                                          NEPL_F (?G,?PLF) * :INV
[R60]   ADJ_F (?POS,?N,?G)       ==>      AS_F (?POS,?SGM,?,?,?,?)
                                          AE_F (?N,?G,?SGM)
                                          TRM_F (?) *
[R61]   ADJ_F (?POS,?N,?G)       ==>      AS_F (?POS,?,?SGF,?,?,?)
                                          AE_F (?N,?G,?SGF)
                                          TRM_F (?) *
[R62]   ADJ_F (?POS,?N,?G)       ==>      AS_F (?POS,?SGM,?,?PLM,?,?)
                                          AE_F (?,?G,?SGM)
                                          AE_F (?N,?G,?PLM)
                                          TRM_F (?) *
[R63]   ADJ_F (?POS,?N,?G)       ==>      AS_F (?POS,?,?SGF,?,?PLF,?)
                                          AE_F (?,?G,?SGF)
                                          AE_F (?N,?G,?PLF)
                                          TRM_F (?) *
```

A.2.3. French sentence grammar

```
[R64]   NP_F (?N,?P,?G)          ==>      N_F (?N,?P,?G)
                                          ADJ_F (n,?N,?G) *
```

A.2.4. French inclusion grammars

```
[R65]   ADJ_F (?,?,?)                 ==>      ADJ_E (?) * 100
[R66]   PRN_F ()                      ==>      NPR_E (?,?,?) * 100
[R67]   V_F (?,?,?NR,?P,?,?,non)      ==>      V_E (?,?,?NR,?P) * 180
[R68]   N_F (?NR,?)                   ==>      N_G (?NR,?,?) * 100
[R69]   ADJ_F (?,?N,?)                ==>      ADJ_G (?,?N,?,?,?) * 100
```

A.3. German lexicon and grammars

A.3.1. German lexicon

```
[L103] PRGTRM ()                         "<PB>"  ""  0 :WORD_END
[L104] PCT_G (f,s)                       "."     ""  :WORD_END
[L105] PCT_G (f,s)                       ". "    ""  :WORD_END
[L106] TRM_G (?)                         " "     ""  0 :WORD_END
[L107] TRM_G (?)                         ""      ""  100
[L108] TRM_G (abbr)                      ""      ""  1

[L109] NS_G (sk3,pk7,m)                              "film+"     "'fIlm+"
[L110] NS_G (sk1,pk4,f)                              "nation+"   "na't_sˆio:n+"
[L111] NS_G (sk2,non,m)                              "ton+"      "'to:n+"
[L112] NES_G (sk1,?)                                 ""          ""
[L113] NES_G (sk2,n)                                 ""          ""
[L114] NES_G (sk2,g)                                 "s"         "s"
[L115] NES_G (sk2,d)                                 ""          ""
[L116] NES_G (sk2,a)                                 ""          ""
[L117] NES_G (sk3,n)                                 ""          ""
[L118] NES_G (sk3,g)                                 "es"        "@s"
[L119] NES_G (sk3,d)                                 ""          ""
[L120] NES_G (sk3,d)                                 "e"         "@"
[L121] NES_G (sk3,a)                                 ""          ""
[L122] NEP_G (pk4,?)                                 "en"        "@n"
[L123] NEP_G (pk7,n)                                 "e"         "@"
[L124] NEP_G (pk7,g)                                 "e"         "@"
[L125] NEP_G (pk7,d)                                 "en"        "@n"
[L126] NEP_G (pk7,a)                                 "e"         "@"

[L127] VS_G (v1,a,v,non)                             "datier+"   "da'ti:r+"
[L128] VS_G (v7,a,v,non)                             "geb+"      "'ge:b+"
[L129] VS_G (v6,a,auxh,non)                          "hab+"      "'ha:b+"
[L130] VS_G (v6,b,auxh,non)                          "ha+"       "'ha+"
[L131] VS_G (v1,a,v,a)                               "lös+"      "'l2:z+"
[L132] VE_G (v1,a,ind,pres,sg,pers1)                 "e"         "@"
[L133] VE_G (v1,a,ind,pres,sg,pers2)                 "st"        "st"
[L134] VE_G (v1,a,ind,pres,sg,pers3)                 "t"         "t"
[L135] VE_G (v6,a,ind,pres,sg,pers1)                 "e"         "@"
[L136] VE_G (v6,b,ind,pres,sg,pers2)                 "st"        "st"
[L137] VE_G (v6,b,ind,pres,sg,pers3)                 "t"         "t"
[L138] VE_G (v12,a,ind,pres,sg,pers1)                "iere"      "'i:r@"
[L139] VE_G (v12,a,ind,pres,sg,pers2)                "ierst"     "'i:rst"
[L140] VE_G (v12,a,ind,pres,sg,pers3)                "iert"      "'i:rt"
[L141] P1SUFF_G ()                                   "end+"      "@nd+"
[L142] P2PREF_G ()                                   "ge+"       "g@+"
[L143] P2E_G (v1)                                    "et"        "@t"
[L144] P2E_G (v1)                                    "t"         "t"
[L145] P2E_G (v12)                                   "iert"      "'i:rt"

[L146] AS_G (pos,non,non)                            "schwarz+"  "'Svart_s+"
[L147] AE_G (typ2,n,sg,f)                            "e"         "@"
[L148] AE_G (typ2,g,sg,f)                            "en"        "@n"
[L149] AE_G (typ2,d,sg,f)                            "en"        "@n"
[L150] AE_G (typ2,a,sg,f)                            "e"         "@"
[L151] ARTDEFS_G (n,sg,m)                            "der"       "'de:r"
[L152] ARTDEFS_G (n,sg,f)                            "die"       "'di:"
[L153] PREF_G (v,p3,sep)                             "ab+"       "'?ap+"
```

```
[L154]  PREF_G (v,p3,sep)                    "an+"       "'?an+"
[L155]  CONJS_G (sub,front,c)                "als"       "'?als"
```

A.3.2. German word grammar

```
[R70]   PRGTRM ()                            ==>         PRGTRM () * :SENT_END
[R71]   PCT_G (?F,?T)                        ==>         PCT_G (?F,?T) * :SENT_END
[R72]   N_G (?C,?NR,?G)                      ==>         NOUN_G (?C,?NR,?G)
                                                         TRM_G (?) *
[R73]   NOUN_G (?C,sg,?G)                    ==>         NS_G (?SGCL,?,?G)
                                                         NES_G (?SGCL,?C) * :INV
[R74]   NOUN_G (?C,pl,?G)                    ==>         NS_G (?,?PLCL,?G)
                                                         NEP_G (?PLCL,?C) * :INV
[R75]   P2_G (?,?)                           ==>         PREF_OPT_G (v,?,?)
                                                         P2PREF_G ()
                                                         VS_G (?VCL,?,v,?)
                                                         P2E_G (?VCL)
                                                         TRM_G (?) *
[R76]   P2_G (?,?)                           ==>         PREF_OPT_G (v,?,?)
                                                         VS_G (?VCL,?,v,?)
                                                         P2E_G (?VCL)
                                                         TRM_G (?) *
[R77]   PREF_OPT_G (?U,?T,?S)                ==>         * 0 :INV
[R78]   PREF_OPT_G (?U,?T,?S)                ==>         PREF_G (?U,?T,?S) * 0 :INV
```

A.3.3. German sentence grammar

```
[R79]   NP_G (?C,?NR,?P,?G,?NT)              ==>         DET_G (?C,?NR,?G,?F,?TYP)
                                                         NPNUC_G
                                                         (?C,?NR,?P,?G,?TYP,?NT) *
[R80]   NP_G (?C,?NR,?P,?G,?NT)              ==>         DET_G (?C,?NR,?G,?F,?TYP)
                                                         ADJ_G (?C,?NR,?G,?GR,?TYP)
                                                         NPNUC_G
                                                         (?C,?NR,?P,?G,?TYP,?NT) *
```

A.3.4. German inclusion grammars

```
[R81]   AS_G (p)                             ==>         AS_E (?,pos) * 150
[R82]   NS_G (sk2,pk1,?)                     ==>         NS_E (?,?,?) * 150
[R83]   NS_G (sk2,pk2,?)                     ==>         NS_E (?,?,?) * 150
[R84]   VS_G (v1,a,v,?)                      ==>         VS_E (?,pres) * 150
[R85]   PREF_G (?,p3,sep)                    ==>         PREF_E () * 100

[R86]   VS_G (v12,a,v,?REF)                  ==>         VS_F (?,?,?REF,?,non) * 160

[R87]   N_G (?NR,?,?)                        ==>         N_E (?NR,?,?) * 100
[R88]   PRN_G (?,?,?)                        ==>         NPR_E (?,?,?) * 100
[R89]   PRN_G (?,?NR,?)                      ==>         NP_E (?NR,?) * 110
[R90]   V_G (?,?,?,?,?)                      ==>         V_E (?,?,?,?) * 100
[R91]   ADJ_G (?,?,?,?,?)                    ==>         ADJ_E (?) * 110
[R92]   NP_G (?,?NR,?,?,?)                   ==>         NP_E (?NR,?) * 80
[R93]   NPNUC_G (?,?NR,pers3,?,?,?)          ==>         NP_E (?NR,?) * 90
[R94]   PP_G (?,?,?,?)                       ==>         PP_E () * 100

[R95]   N_G (?NR,?,?)                        ==>         N_F (?NR,?) * 110
[R96]   PRN_G (?,?,?)                        ==>         PRN_F () * 110
[R97]   PRN_G (?,?NR,?)                      ==>         NP_F (?NR,?,?) * 110
```

```
[R98]   V_G (?,?,?,?,?)                          ==>       V_F (?,?,?NR,?,?,?,non) *
                                                           120
[R99]   ADJ_G (?,?,?,?,?)                        ==>       ADJ_F (?,?,?) * 110
[R100]  NP_G (?,?NR,?,?,?)                       ==>       NP_F (?NR,?,?) * 90
[R101]  NPNUC_G (?,?NR,pers3,?,?,?)              ==>       NP_F (?NR,?,?) * 90
[R102]  PP_G (?,?NR,?,?)                         ==>       PP_F (?NR,?) * 110
[R103]  N_G (?,?,?)                              ==>       N_I (?,?) * 110
[R104]  PRN_G (?C,?NR,?G)                        ==>       NPR_I (?) * 110
[R105]  PRN_G (?,?NR,?)                          ==>       NP_I (?NR,?,?) * 110
[R106]  ADJ_G (?,?,?,?,?)                        ==>       ADJ_I (?,?) * 110
```

A.4. Italian lexicon and grammars

A.4.1. Italian lexicon

```
[L156]     PRGTRM ()          "<PB>"     ""     0             :WORD_END
[L157]     PCT_I (f,s)        "."        ""                   :WORD_END
[L158]     PCT_I (f,s)        ". "       ""                   :WORD_END
[L159]     TRM_I (?)          " "        ""     0             :WORD_END
[L160]     TRM_I (?)          ""         ""     100
[L161]     TRM_I (abbr)       ""         ""     1

[L162]     NS_I (null,m)                        "caff‘e+"     "kaf_f’E+"
[L163]     NS_I (e,m)                           "latt+"       "l’at_t+"
[L164]     NE_I (e,sg,m)                        "e"           "e"
[L165]     NE_I (e,pl,m)                        "i"           "i"
[L166]     NE_I (null,sg,m)                     ""            ""
[L167]     NE_I (null,pl,m)                     ""            ""

[L168]     AS_I (o)                             "dat+"        "d’a:t+"
[L169]     AE_I (o,sg,m)                        "o"           "o"
[L170]     AE_I (o,pl,m)                        "i"           "i"
[L171]     AE_I (o,sg,f)                        "a"           "a"
[L172]     AE_I (o,pl,f)                        "e"           "e"
```

A.4.2. Italian word grammar

```
[R107]     PRGTRM ()                            ==>           PRGTRM () * :SENT_END
[R108]     PCT_I (?F,?T)                        ==>           PCT_I (?F,?T) * :SENT_END
[R109]     N_I (?N,?G)                          ==>           NOUN_I (?N,?G)
                                                              TRM_I (?) *
[R110]     NOUN_I (?N,?G)                       ==>           NS_I (?CL,?G)
                                                              NE_I (?CL,?N,?G) * :INV
[R111]     ADJ_I (?N,?G)                        ==>           AS_I (?CL)
                                                              AE_I (?CL,?N,?G)
                                                              TRM_I (?) *
```

References

Boula de Mareüil, P., Floricic, F., September 2001. On the pronunciation of acronyms in French and Italian. In: Proc. Eurospeech 2001, Aalborg, Denmark, pp. 1923–1926.

Cavnar, W.B., Trenkle, J.M., April 1994. N-gram based text categorization. In: 3rd Annual Symp. on Document Analysis and Information Retrieval, pp. 161–169.

Coker, C.H., Church, K.W., Liberman, M.Y., September 1990. Morphology and rhyming: two powerful alternatives to letter-to-sound rules for speech synthesis. In: Proc. ESCA Workshop on Speech Synthesis, Autrans, France, pp. 83–86.

Dutoit, T., October 1993. High quality text-to-speech synthesis of the French language. Ph.D. thesis, Faculté Polytechnique de Mons.

Giguet, E., September 1995. Categorization according to language: a step toward combining linguistic knowledge and statistic learning.

In: 4th Internat. Workshop of Parsing Technologies, Prague, Czech Republic.

Grefenstette, G., December 1995. Comparing two language identification schemes. In: Proc. 3rd Internat. Conf. on the Statistical Analysis of Textual Data, Rome, Italy, pp. 1–6.

Häkkinen, J., Tian, J., 2001. *N*-gram and decision tree based language identification for written words. In: IEEE Workshop on Automatic Speech Recognition and Understanding, Italy, pp. 335–338.

Liberman, M., Church, K., 1991. Text analysis and word pronunciation in text-to-speech synthesis. In: Furui, S., Sondhi, M. (Eds.), Advances in Speech Signal Processing, pp. 791–831 (chapter 24).

McAllister, M., September 1989. The problems of punctuation ambiguity in fully automatic text-to-speech conversion. In: Proc. Eurospeech 89, Vol. 1, Paris, France, pp. 538–541.

Pereira, F.C.N., Warren, D.H.D., 1980. Definite clause grammars for language analysis – a survey of the formalism and a comparison with augmented transition networks. Artif. Intell. 13, 231–278.

Pfister, B., Romsdorfer, H., September 2003. Mixed-lingual text analysis for polyglot TTS synthesis. In: Proc. Eurospeech'03, Geneva, Switzerland, pp. 2037–2040.

Riedi, M., September 1997. Modeling segmental duration with multivariate adaptive regression splines. In: Proc. Eurospeech'97, pp. 2627–2630.

Riley, M., 1989. Some applications of tree-based modelling to speech and language. In: Proc. DARPA Speech and Natural Language Workshop, Cape Cod, MA, pp. 339–352.

Romsdorfer, H., Pfister, B., October 2004. Multi-context rules for phonological processing in polyglot TTS synthesis. In: Proc. Interspeech 2004 – ICSLP, Jeju Island, Korea, pp. 737–740.

Romsdorfer, H., Pfister, B., 2005. Phonetic labeling and segmentation of mixed-lingual prosody databases. In: Proc. Interspeech 2005, Lisbon, Portugal, pp. 3281–3284.

Romsdorfer, H., Pfister, B., April 2006. Character stream parsing of mixed-lingual text. In: ISCA Tutorial and Research Workshop on Multilingual Speech and Language Processing (MultiLing 2006), Stellenbosch, South Africa.

Romsdorfer, H., Pfister, B., Beutler, R., 2005. A mixed-lingual phonological component which drives the statistical prosody control of a polyglot TTS synthesis system. In: Bengio, S., Bourlard, H. (Eds.), Machine Learning for Multimodal Interaction. Springer-Verlag, Heidelberg, pp. 263–276.

Schmitt, J.C., October 1991. Trigram-based method of language identification. US Patent number: 5062143.

Sproat, R., Chilin, S., Gale, W., Chang, N., 1996. A stochastic finite-state word-segmentation algorithm for Chinese. In: Computational Linguistics No. 22, Vol. 3, pp. 377–404.

Tian, J., Häkkinen, J., Riis, S., Jensen, K.J., September 2002. On text-based language identification for multilingual speech recognition systems. In: Proc. ICSLP 2002. Denver, Colorado, USA.

Tian, J., Suontaustaa, J., May 2004. Scalable neural network based language identification from written text. In: Proc. ICASSP 2004, Montreal, Canada.

Traber, C., March 1995. SVOX: the implementation of a text-to-speech system for German. Ph.D. thesis, No. 11064, Computer Engineering and Networks Laboratory, ETH Zurich (TIK-Schriftenreihe Nr. 7, ISBN 3 7281 2239 4).

Traber, C., Pfister, B., et al., September 1999. From multilingual to polyglot speech synthesis. In: Proc. Eurospeech'99, pp. 835–838.