MULTILINGUAL TEXT ANALYSIS FOR TEXT-TO-SPEECH SYNTHESIS

Richard Sproat

Speech Synthesis Research Department
Bell Laboratories, Murray Hill, NJ, USA

ABSTRACT

We present a model of text analysis for text-to-speech (TTS) synthesis based on weighted finite-state transducers, which serves as the text-analysis module of the multilingual Bell Labs TTS system. The transducers are constructed using a lexical toolkit that allows declarative descriptions of lexicons, morphological rules, numeral-expansion rules, and phonological rules, inter alia. To date, the model has been applied to eight languages: Spanish, Italian, Romanian, French, German, Russian, Mandarin and Japanese.

1. INTRODUCTION

The first task faced by any text-to-speech (TTS) system is the conversion of input text into a linguistic representation. This is a complex task since the written form of any language is at best an imperfect representation of the corresponding spoken forms. Among the problems that one faces in handling ordinary text are the following:

1. Some languages, such as Chinese, do not delimit words with whitespace. One is therefore required to 'reconstruct' word boundaries in TTS systems for such languages.
2. Digit sequences need to be expanded into words, and more generally into well-formed number names: so *243* in English would generally be expanded as *two hundred and forty three*.
3. Abbreviations must be expanded into full words. This can involve some amount of contextual disambiguation: so *kg.* can be either *kilogram* or *kilograms*, depending upon the context.
4. Ordinary words and names need to pronounced. In many languages, this requires morphological analysis: even in languages with fairly 'regular' spelling, morphological structure is often crucial in determining the pronunciation of a word.
5. Prosodic phrasing is only sporadically marked (by punctuation) in text, and phrasal accentuation is almost never marked.

In many TTS systems the first three tasks — word segmentation, and digit and abbreviation expansion — would be classed under the rubric of *text normalization* and would generally be handled prior to, and often in a quite different fashion from the last two problems, which fall more squarely within the domain of linguistic analysis (but see [12], which treats numeral expansion as an instance of morphological analysis). One problem with this approach is that in many cases the selection of the correct linguistic form for a 'normalized' item cannot be chosen before one has done a certain amount of linguistic analysis. A particularly compelling example can be found in Russian, where deciding how to read a percentage denoted with '%' depends on complex contextual factors, unlike the situation in English. The first decision that needs to be made is whether or not the number-percent string is modifying a following noun. Russian in general disallows noun-noun modification, so that an adjectival form of *procent* 'percent' must be used: *20% skidka* 'twenty percent discount' is rendered as *dvadcati-procent***naja** *skidka* (twenty$_{[\mathrm{gen}]}$-percent+**adj**$_{[\mathrm{nom,sg,fem}]}$ discount$_{[\mathrm{nom,sg,fem}]}$). Not only does *procent* have to be in the adjectival form, but as with any Russian adjective it must also agree in number, case and gender with the following noun. Observe also that the word for 'twenty' must occur in the genitive case. In general, numbers which modify adjectives in Russian must occur in the genitive case. If the percentage expression is not modifying a following noun, then the nominal form *procent* is used, but this form appears in different cases depending upon the number it occurs with. With numbers ending in *one* (including compound numbers like *twenty one*), *procent* occurs in the nominative singular. After so-called paucal numbers — *two*, *three*, *four* and their compounds — the genitive singular *procenta* is used. After all other numbers one finds the genitive plural *procentov*. So we have *odin procent* (one percent$_{[\mathrm{nom,sg}]}$), *dva procenta* (two percent$_{[\mathrm{gen,sg}]}$), and *pyat' procentov* (five percent$_{[\mathrm{gen,pl}]}$). However, all of this presumes that the percentage expression as a whole is in a non-oblique case. If the expression is in an oblique case, then both the number and *procent* show up in that case, with *procent* being in the singular if the number ends in *one*, and the plural otherwise: *s odnym procentom* (with one$_{[\mathrm{instr,sg,masc}]}$ percent$_{[\mathrm{instr,sg}]}$) 'with one percent'; *s pjat'ju procentami* (with five$_{[\mathrm{instr,pl}]}$ percent$_{[\mathrm{instr,pl}]}$) 'with five percent.' As with the adjectival forms, there is nothing peculiar about the behavior of the noun *procent*: all nouns exhibit identical behavior in combination with numbers. The complexity, of course, arises because the written form *%* gives no indication of what linguistic form it corresponds to. Furthermore, there is no way to correctly expand this form without doing a substantial amount of analysis of the context, including some analysis of the morphological properties of the surrounding words, as well as an analysis of the relationship of the percentage expression to those words.

The obvious solution to this problem is to delay the decision on how exactly to transduce symbols like or '%' until one has enough

information to make the decision in an informed manner. This suggests is a model where, say, an expression like '20%' in Russian is transduced into all possible renditions, and the correct form selected from the lattice of possibilities by filtering out the illegal forms. An obvious computational mechanism for accomplishing this is the *finite-state transducer* (FST). Indeed, since FSTs can be used to model (most) morphology and phonology [5, 8], as well as to segment words in Chinese text [11], and for performing other text-analysis operations such as numeral expansion [9], this suggests a model of text-analysis that is entirely based on regular relations. We present such a model below. More specifically we present a model of text analysis for TTS based on *weighted* FSTs (WFSTs) [7], which serves as the text-analysis module of the multilingual Bell Labs TTS system. To date, the model has been applied to eight languages: Spanish, Italian, Romanian, French, German, Russian, Mandarin and Japanese. One property of this model that distinguishes it from most TTS text-analyzers is that such tasks as numeral expansion and word-segmentation are *not* logically prior to other aspects of linguistic analysis. There is therefore no distinguished 'text-normalization' phase.

2. OVERALL ARCHITECTURE

Let us start with the example of the lexical analysis and pronunciation of ordinary words, taking again an example from Russian. Russian orthography is often described as morphophonemic, meaning that the orthography represents not a surface phonemic level of representation, but a more abstract level. This is description is correct, but from the point of view of predicting word pronunciation, it is noteworthy that Russian, with a well-defined set of lexical exceptions, is almost completely phonemic in that one can predict the pronunciation of most words in Russian based on the spelling of those words — provided one knows the placement of lexical stress, since several Russian vowels undergo reduction to varying degrees depending upon their position relative to stressed syllables. The catch is that lexical stress usually depends upon knowing lexical properties of the word, including morphological class information. To take a concrete example, consider the word *kostra* (Cyrillic костра) (bonfire+genitive.singular). This word belongs to a class of masculine nouns where the lexical stress is placed on the inflectional ending, where there is one. Thus the stress pattern is *kostr′a*, and the pronunciation is /kəstr′ʌ/, with the first /o/ reduced to /ə/. Let us assume that the morphological representation for this word is something like *kostr*{*noun*}{*fem*}{*inan*}+′*a*{*sg*}{*nom*}, where for convenience we represent phonological and morphosyntactic information as part of the same string.[1] Assuming a finite-state model of lexical structure [5, 8], we can easily imagine a set of transducers M that map from that level into a level that gives the minimal morphologically-motivated annotation (MMA) necessary to pronounce the word. In this case, something like *kostr′a* would be appropriate. Call this *lexical-to-MMA* transducer L_{word}; such a transducer can be constructed by composing the lexical acceptor D with M so that $L_{word} = D \circ T$. A transducer that maps from the MMA to the standard spelling *kostra* (костра) would, among other things, simply delete the stress marks: call this transducer S. The composition $L_{word} \circ S$, then computes the mapping from the lexical level to the surface orthographic level, and its inverse $(L_{word} \circ S)^{-1} = S^{-1} \circ L_{word}^{-1}$ computes the mapping from the surface to all possible lexical representations for the text word. A set of pronunciation rules compiled into a transducer P [2, 6], maps from the MMA to the (surface) phonological representation; note that by starting with the MMA, rather than with the more abstract lexical representation, the pronunciation rules do not need to duplicate information that is contained in L_{word} anyway. Mapping from a single orthographic word to its possible pronunciations thus involves composing the acceptor representing the word with the transducer $S^{-1} \circ L_{word}^{-1} \circ L_{word} \circ P$ (or more fully as $S^{-1} \circ M^{-1} \circ D \circ M \circ P$).

For text elements such as numbers, abbreviations, and special symbols such as '%', the model just presented seems less persuasive, because there is no aspect of a string, such as '25%' that indicates its pronunciation: such strings are purely *logographic* (or even *ideographic*) representing nothing about the phonology of the words involved.[2] For these cases we presume a direct mapping between all possible forms of *procent*, and the symbol '%': call this transducer L_{perc}, a subset of $L_{special\ symbol}$. Then L_{perc}^{-1} maps from the symbol '%' to the various forms of *procent*. In the same way, the transducer $L_{num}^{-1} \frown L_{perc}^{-1}$ maps from numbers followed by the sign for percent, into various possible (and some impossible) lexical renditions of that string — the various forms to be disambiguated using contextual information, as we shall show below. Abbreviations are handled in a similar manner: abbreviations such as *kg* (кг) in Russian show the same complexity of behavior as *procent*.

So far we have been discussing the mapping of single text words into their lexical renditions. The construction of an analyzer to handle a whole text is based on the observation that a text is simply constructed out of one or more instances of a text word coming from one of the models described above — either an ordinary word, an abbreviation, a number, a special symbol, or some combination of numbers with a special symbol; with each of these tokens separated by some combination of whitespace or punctuation. The structure of this model of lexical analysis is summarized in Figure 1. We pre-

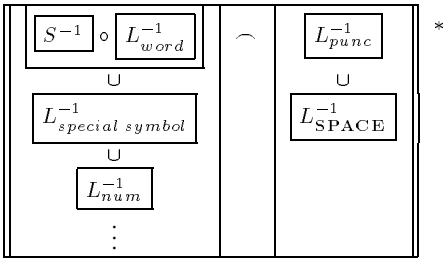

Figure 1: Structure of lexical analysis. Note that SPACE corresponds to whitespace in (e.g.) Russian, but ϵ in (e.g.) Chinese.

[1] Conversion between a 'flattened' representation of this kind and a hierarchical representation more in line with standard linguistic models of morphology and phonology, is straightforward and we will not dwell on this issue here.

[2] This is certainly not completely true in all such cases, as 'mixed' representations such as *1st* and *2nd* suggest. But such cases are most easily treated as also being logographic, at least in the present architecture.

sume two models for space and punctuation. The model L^{-1}_{SPACE}, maps between interword SPACE and its potential lexical realizations, usually a word boundary, but in some cases a higher-level prosodic phrase boundary. Interword SPACE is parameterized so that in European languages, for example, it corresponds to actual whitespace, whereas in Chinese or Japanese, it corresponds to ϵ. Similarly, the model L^{-1}_{punc} maps between punctuation marks (possibly with flanking whitespace) and the lexical realization of those marks: in many, though not all, cases the punctuation mark may correspond to a prosodic phrase boundary.

The output of the lexical analysis WFST diagrammed in Figure 1 is a lattice of all possible lexical analyses of all words in the input sentence. Obviously in general we want to remove contextually inappropriate analyses, and to pick the 'best' analysis in cases where one cannot make a categorical decision. This is accomplished by a set of one or more *language model* transducers — henceforth Λ — which are derived from rules or other expressions that examine contexts wider than the lexical word. Phrasal accentuation and prosodic phrasing are also handled by the language model transducers.[3] The output of composing the lexical analysis WFST with Λ is a lattice of contextually disambiguated lexical analyses. The lowest-cost path of this lattice is then selected, using a Viterbi *best path* algorithm. Weights on the lattice may be weights hand-selected to disfavor certain lexical analyses — see the Russian percentage example detailed in the next section; or they may be genuine data-derived weight estimates, as in the case of the Chinese lexical analysis WFST, where the weights correspond to the negative log (unigram) probability of a particular lexical entry [11]. Given the best lexical analysis, one can then proceed to apply the phonological transducer (or set of transducers) P to the lexical analysis, or more properly to the lexical analysis composed with the lexical-to-MMA map M, as we saw above. Although the lexical-to-MMA map M was introduced as mapping from the lexical analyses of ordinary words to their MMA, if the map is constructed with sufficient care it can serve as the transducer for lexical analyses coming from any of the text-word models.

The construction of the WFSTs depends upon a lexical toolkit, which allows for the description of linguistic generalizations in linguistically sensible human-readable form: the toolkit is thus similar in spirit to the Xerox tools [4, 3], though the latter do not allow weights in the descriptions. Considerations of space do not allow me to describe the toolkit here: the reader is referred to the longer version of this paper in [9], and also [6] and [10] for a description.[4]

3. RUSSIAN PERCENTAGES

Let us return to the example of Russian percentage terms. Assume that we start with a fragment of text such as с 5% скидкой *s 5% skidkoj* (with 5% discount) 'with a five-percent discount'. This is

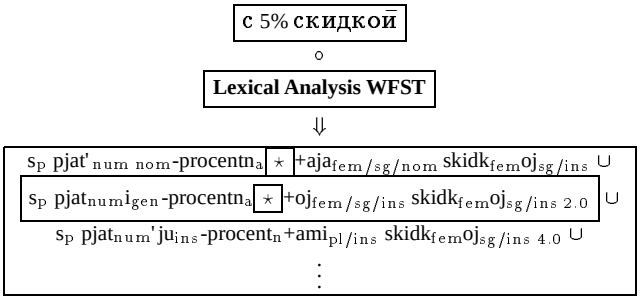

Figure 2: Composition of с 5% скидкой *s 5% skidkoj* 'with a 5% discount' with the lexical analysis WFST to produce a range of possible lexical renditions for the phrase.

first composed with the lexical analysis WFST to produce a set of possible lexical forms; see Figure 2. By default the lexical analyzer marks the adjectival readings of '%' with '$\star$', meaning that they will be filtered out by the language-model WFSTs, if contextual information does not save them. Weights on analyses mark constructions — usually oblique case forms — that are not in principle ill-formed but are disfavored except in certain well-defined contexts. The correct analysis (boxed in Figure 2), for example, has a weight of 2.0 which is an arbitrary weight assigned to the oblique instrumental adjectival case form: the preferred form of the adjectival rendition '%' is masculine, nominative, singular if no constraints apply to rule it out. Next the language model WFSTs Λ are composed with the lexical analysis lattice. See Figure 3. The WFSTs Λ include transducers compiled from rewrite rules that ensure that the adjectival rendition of '%' is selected whenever there is a noun following the percent expression (the first block in Figure 3), and rules that ensure the correct case, number and gender of the adjectival form given the form of the following noun (the second block in Figure 3). In addition, a filter expressible as $\neg(\Sigma^* \boxed{\star} \Sigma^*)$ removes any analyses containing the tag $\star$ (the third block in Figure 3). The lowest-weight analysis among the remaining analyses is then selected. Finally, the lexical analysis is composed with $M \circ P$ to produce the phonemic transcription.

4. SIZE AND SPEED ISSUES

Table 1 gives the sizes of the lexical analysis WFSTs for the languages German, Spanish, Russian and Mandarin. To a large extent, these sizes accord with our intuitions of the difficulties of lexical processing in the various languages. So Russian is very large, correlating with the complexity of the morphology in that language. German is somewhat smaller. Mandarin has a small number of states, correlating with the fact that Mandarin words tend to be simple in terms of morphemic structure; but there are a relatively large number of arcs, due to the large character set involved. Sizes for the Spanish transducer are misleading since the current Spanish system includes only minimal morphological analysis: note that morphological analysis is mostly unnecessary in Spanish for correct word pronunciation. While the transducers can be large, the performance (on, e.g., an SGI Indy) is acceptably fast for a TTS application. Slower performance is certainly observed, however, when the sys-

[3] To date our multilingual systems have rather rudimentary lexical-class based accentuation rules, and punctuation-based phrasing. Thus these components of the systems are not as sophisticated as the equivalent components of our English system; see [1, 13]. This is largely because the relevant research has not been done for most of the languages in question, rather than for technical problems in fitting the results of that research into the model.

[4] The toolkit is built on top of a WFST toolkit built by Michael Riley, Fernando Pereira and Mehryar Mohri of AT&T Research.

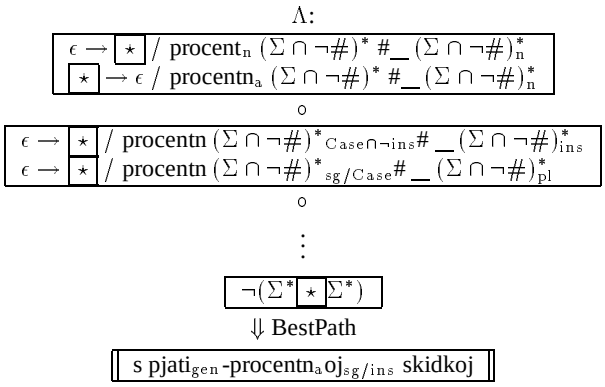

Figure 3: Portion of the language model WFSTs related to the rendition of percentages. The (correct) output of this sequence of transductions for the lattice from Figure 2 is shown at the bottom.

	States	Arcs
German	77295	207859
Russian	139592	495847
Mandarin	48015	278905
Spanish	8602	17236

Table 1: Sizes of lexical analysis WFSTs for selected languages.

tem is required to explore certain areas of the network, as for example in the case of expanding and disambiguating Russian number expressions. To date, no formal evaluations have been performed on the correctness of word-pronunciation in the various languages under development, since there remains work to be done before the systems can be called complete. An evaluation of the correctness of word segmentation in the Mandarin system is reported in [11].

5. SUMMARY AND FUTURE WORK

The system for text analysis presented in this paper is a complete working system that has been used in the development of text-analyzers for several languages. In addition to German, Spanish, Russian and Mandarin, a system for Romanian has been built, and work on French, Japanese and Italian is underway. From the point of view of previous research on linguistic applications finite-state transducers some aspects of this work are familiar, some less so. Familiar, of course, are applications to morphology, phonology, and syntax, though most previous work in these areas has not made use of *weighted* automata. More novel are the applications to text 'preprocessing', in particular numeral expansion and word segmentation. From the point of view of text-analysis models for text-to-speech the approach is quite novel since, as described in the introduction, most previous work treats certain operations, such as word segmentation or numeral expansion in a preprocessing phase that is logically prior to the linguistic analysis phase; we have argued here against this view. Areas of future work include incorporating decision-tree-based models of phrasing [13] and decision-list-based sense-disambiguation methods [14] into the text-analysis model using the tree compiler described in [10] and similar tools.

6. REFERENCES

1. Julia Hirschberg. Pitch accent in context: Predicting intonational prominence from text. *Artificial Intelligence*, 63:305–340, 1993.
2. Ronald Kaplan and Martin Kay. Regular models of phonological rule systems. *Computational Linguistics*, 20:331–378, 1994.
3. Lauri Karttunen. Finite-state lexicon compiler. Technical Report P93–00077, Xerox Palo Alto Research Center, 1993.
4. Lauri Karttunen and Kenneth Beesley. Two-level rule compiler. Technical Report P92–00149, Xerox Palo Alto Research Center, 1992.
5. Kimmo Koskenniemi. *Two-Level Morphology: a General Computational Model for Word-Form Recognition and Production*. PhD thesis, University of Helsinki, Helsinki, 1983.
6. Mehryar Mohri and Richard Sproat. An efficient compiler for weighted rewrite rules. In *34rd Annual Meeting of the Association for Computational Linguistics*, Morristown, NJ, 1996. Association for Computational Linguistics.
7. Fernando Pereira, Michael Riley, and Richard Sproat. Weighted rational transductions and their application to human language processing. In *ARPA Workshop on Human Language Technology*, pages 249–254. Advanced Research Projects Agency, March 8–11 1994.
8. Richard Sproat. *Morphology and Computation*. MIT Press, Cambridge, MA, 1992.
9. Richard Sproat. Multilingual text analysis for text-to-speech synthesis. In *Proceedings of the ECAI-96 Workshop on Extended Finite State Models of Language*, Budapest, Hungary, 1996. European Conference on Artificial Intelligence.
10. Richard Sproat and Michael Riley. Compilation of weighted finite-state transducers from decision trees. In *34rd Annual Meeting of the Association for Computational Linguistics*, Morristown, NJ, 1996. Association for Computational Linguistics.
11. Richard Sproat, Chilin Shih, William Gale, and Nancy Chang. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3), 1996.
12. Christof Traber. SVOX: The implementation of a text-to-speech system for German. Technical Report 7, Swiss Federal Institute of Technology, Zurich, 1995.
13. Michelle Wang and Julia Hirschberg. Automatic classification of intonational phrase boundaries. *Computer Speech and Language*, 6:175–196, 1992.
14. David Yarowsky. Homograph disambiguation in text-to-speech synthesis. In Jan van Santen, Richard Sproat, Joseph Olive, and Julia Hirschberg, editors, *Progress in Speech Synthesis*. Springer, New York, 1996.