ELSEVIER

Speech Communication 18 (1996) 47–63

SPEECH COMMUNICATION

Reversible letter-to-sound/sound-to-letter generation based on parsing word morpology ☆

Helen Meng [a], Sheri Hunnicutt [b], Stephanie Seneff [a], Victor Zue [a,*]

[a] *Spoken Language Systems Group, MIT Laboratory for Computer Science, 545 Technology Square, NE43-642, Cambridge, MA 02139, USA*

[b] *Department of Speech Communication and Music Acoustics, Royal Institute of Technology, Stockholm, Sweden*

Received 6 September 1994; revised 20 June 1995

Abstract

This paper describes a bi-directional letter/sound generation system based on a strategy combining data-driven techniques with a rule-based formalism. Our approach provides a hierarchical analysis of a word, including stress pattern, morphology and syllabification. Generation is achieved by a probabilistic parsing technique, where probabilities are trained from a parsed lexicon. Our training and testing corpora consisted of spellings and pronunciations for the high frequency portion of the Brown Corpus (10,000 words). The phonetic labels are augmented with markers indicating morphology and stress. We will report on two distinct grammars representing a historical perspective. Our early work with the first grammar inspired us to modify the grammar formalism, leading to greater constraint with fewer rules. We evaluated our performance on letter-to-sound generation in terms of whole word accuracy as well as phoneme accuracy. For the unseen test set, we achieved a word accuracy of 69.3% and a phoneme accuracy of 91.7% using a set of 52 distinct phonemes. While this paper focuses on letter-to-sound generation, our system is also capable of generation in the reverse direction, as reported elsewhere (Meng et al., 1994a). We believe that our formalism will be especially applicable for entering unknown words orally into a recognition system.

Zusammenfassung

Dieser Artikel beschreibt ein orthographisch–phonetisches Umsetzungssystem, das in zwei Richtungen arbeitet. Es beruht auf einer Strategie, die die auf Daten basierende Technik mit dem Formalismus eines Regelwerks kombiniert. Unser Ansatz liefert die hierarchische Analyse der Wörter, die die Akzentuierung, die Morphologie und die Silbenstruktur bestimmt. Die Umsetzung wird mit einer auf Wahrscheinlichkeitstechnik beruhenden syntaktischen Analyse durchgeführt, bei der die Wahrscheinlichkeiten anhand eines Lexikons trainiert werden. Der

☆ This paper is based on a communication presented at the ESCA Conference EUROSPEECH-93 and has been recommended by the EUROSPEECH-93 scientific program committee.

* Corresponding author. E-mail: hmmeng@goldilocks.lcs.mit.edu.

SSDI 0167-6393(95)00032-1

Trainings- und Testkorpus besteht aus der Buchstabierung und der Aussprache von den 10000 meistgebräuchlichen Wörtern des Brown Corpus. Die phonetische Markierung wurde durch morphologische und akzentuelle Marker erweitert. Die Resultate werden von zwei verschiedenen Grammatiken geliefert, die zwei verschieden Entwicklungsstadien der Arbeit entsprechen. Unsere früheren Arbeiten mit der ersten Grammatik haben zu einer Modifizierung des Formalismus geführt, mit größeren Einschränkungen und weniger Regeln. Die Leistungsfähigkeit des Systems wurde sowohl auf Wort- als auch auf Phonemebene getestet. Bei dem Testkorpus wurde auf Wortebene eine Präzision von 69,3% erreicht und auf Phonemebene 91,7% unter Verwendung von 52 Phonemen. Auch wenn dieser Artikel sich hauptsächlich mit der orthographisch–phonetischen Umsetzung befaßt, so ist unser System doch auch zu der gegensätzliche Umsetzung fähig, wie bei Meng et al. (1994a) berichtet. Unserer Überzeugung nach ist dieses System besonders für die mündliche Eingabe unbekannter Wörter in ein Spracherkennungssystem geeignet.

Résumé

Cet article décrit un système de transcription orthographique–phonétique bidirectionnel basé sur une stratégie combinant des techniques basées sur les données et un formalisme de règles. Notre approche fournit une analyse hiérarchique des mots, incluant la position de l'accent, sa morphologie et sa structure syllabique. La génération est réalisée par une technique d'analyse syntaxique probabiliste où les probablilitiés sont apprises à partir d'un lexique. Nos corpus d'apprentissage et de test sont constitués d'épellations et de prononciations des 10000 mots les plus fréquents du Brown Corpus. L'étiquetage phonétique a été enrichi par des marqueurs indiquant la morphologie et l'accentuation. Les résultats sont fournis sur deux grammaires distinctes, correspondant à deux stades d'évolution du travail. Notre travail antérieur avec la première grammaire nous a incité à modifier le formalisme de la grammaire, ce qui a abouti à des contraintes plus fortes avec moins de règles, Nous avons évalué les performances de notre système tant au niveau du mot entier que du phonème. Pour le corpus de test, la précision atteinte au niveau du mot est de 69.3%; elle est de 91.7% au niveau du phonème, en utilisant un répertoire de 52 phonèmes. Bien que cet article traite essentiellement de la transcription orthographique–phonétique, notre système est également un système de génération dans l'autre sens, comme décrit dans un article antérieur (Meng et al., 1994a). Nous pensons que notre formalisme sera applicable en particulier pour entrer vocalement des mots inconnus dans un système de reconnaissance.

Keywords: Reversible letter/sound generation; Phonological parsing; Morphology

1. Introduction

English is a language that is characterized by a rich mixture of words inherited from multiple languages. As a consequence of these borrowings, letter/sound mappings in English can be highly irregular. However, it is apparent that many regularities exist: those who are knowledgeable in English are usually able to pronounce with high accuracy words not known to them.

Much research has been done with the goal of achieving reliable automatic letter-to-sound generation. Speech output is especially applicable in reading machines and for generating proper name pronunciations for telephone directories. In addition, the text input for the letter-to-sound system may potentially be the output of an optical character recognition system or a handwriting recognition system. In this case, the text may be one of the many hypotheses corresponding to a document image, and therefore may be associated with a confidence level and may contain errors. Thus it will be desirable if the letter-to-sound generation system can handle uncertainty in the text input, since it will be capable of interfacing pen-based systems with speech-based systems. There is also an increasing need for the reverse process of sound-to-letter generation in spoken language systems, as in the automatic addition of new words to the vocabulary of an existing sys-

tem. A spoken language system cannot be expected to fully specify its active vocabulary based on a static initial set. Users should be able to enter new words orally, by providing a (spoken, typed or handwritten) spelling and/or pronunciation. The system might also be able to look up proper nouns in an on-line "yellow-pages". It should then be able to automatically produce pronunciations from input spellings, or spellings from input pronunciations, and dynamically update the recognizer's vocabulary accordingly.

With these needs in mind, we established a goal of creating a bi-directional letter-to-sound and sound-to-letter framework that could handle uncertainties in the input, and provide a *set* of candidate outputs with associated probabilities. Our approach represents a cross between explicit rule-driven strategies and strictly data-driven approaches. Extensive experience with morphological analysis for text-to-speech conversion led us to the hypothesis that the spelling and pronunciation of most English words could be specified simply and explicitly using only a few morphological and phonological units. The pronunciation of a number of words can only be specified with the help of morphological analysis. For example, a morphological boundary causes the letter sequence "sch" in "discharge" to be realized differently from that in "school" or "scheme." Stress shift brought about by suffixation changes the identity of vowels in a word, e.g., "define" versus "definition". Word class can also affect pronunciation. The noun and verb forms of "record", for example, are pronounced differently. Syllabic, phonotactic and graphemic constraints also play a part in letter-to-sound mappings. Therefore, it seems that an English word can be represented by a *hierarchical* structure with several linguistic levels, each being characterized as a sequence of pairwise transitions. In order to produce such hierarchical structures in a parse tree format, the natural language framework, TINA (Seneff, 1992) seems to provide a particularly suitable formalism. TINA parses with a set of probabilistic context-free rules which are automatically trainable. The basic probability unit in TINA predicts the right sibling based on the left context and the parent, with probabilities throughout the parse tree acquired automatically from a parsed training corpus. In our case, the terminal entries are dual in nature, representing phones in one path and letters in the other. Thus there is a direct symmetry between the letter-to-sound and the sound-to-letter tasks, with just the input/output specification being swapped. This paper concentrates on letter-to-sound generation; details for sound-to-letter can be found in (Meng et al., 1994a).

We have organized the paper to reflect a historical progression in our work spanning the last three years. We began with the original version of TINA, which attaches probabilities to sibling-sibling transitions in context-free rules, and uses the generic [START] category as the left context for the probability model to predict the beginning of a rule. While this appeared to be adequate in parsing sentences with a semantic grammar, it turned out to present a problem to our word-level parser in that it led to a great deal of overgeneration. We could compensate somewhat by writing a large number of explicit rules, but this made rule specification more tedious and less well-defined. Also it led to a heavy computational load in exploring partial theories that would fail later. In fact, it was these problems that arose in our letter-to-sound task that led us to reformulate TINA in a paradigm of "layered bigrams" (Seneff et al., 1992). With this new framework, the left context of a right sibling was specified as the left sibling at its layer, regardless of whether that left sibling shared the same parent. The effect can be broadly characterized as producing context dependent rule production probabilities.

Once we had established that this external left sibling was useful for increasing constraint at the sentence level, we then turned our attention back to the problem of parsing words to obtain their morphological decomposition, stress contours and phonological structure. It became apparent that it would be better to keep the category labels very generic in the higher levels of the parse tree, and to organize the hierarchical layers in a very regular fashion, with each layer reflecting a particular aspect of word structure. We also noted that if we include as left context the entire column above the left sibling rather than simply the left sibling

itself, we could gain a great deal more constraint, which would then make it feasible to operate with a simpler rule set.

The rest of the paper is organized as follows. Section 2 begins with a brief survey of some previous efforts conducted in the area. Section 3 discusses the data used in our experiments, including how the words are labelled based on etymology and phonology. Section 4 describes our initial approach, and is followed by Section 5 where a more refined approach is described. Our system's performance is reported in Section 6 and, finally, Section 7 presents some discussion of the results and of possible future directions aimed at improving overall system performance and coverage of unseen test data.

2. Previous work

A myriad of approaches have been applied to the problem of letter-to-sound generation. Excellent reviews can be found in (Damper, Forthcoming; Golding, 1991; Klatt, 1987). The various approaches have given rise to a wide range of letter-to-sound generation accuracies. Many of these accuracies are based on different corpora, and some corpora may be more difficult than others. Furthermore, certain systems are evaluated by human subjects, while others have their pronunciation accuracies reported on a per phoneme or per letter basis. Insertion errors or stress errors may be included in some cases, and ignored in others. There are also systems which look up an exceptions dictionary prior to generation, and the performance accuracies of these sytems tend to increase with the use of larger dictionaries. Due to the above reasons, we should be careful when comparing different systems based on quoted performance values.

The approaches adopted for letter-to-sound generation include the rule-based approach, which uses a set of hand-engineered, ordered rules for transliteration. Transformation rules may also be applied in multiple passes in order to process linguistic units larger than the phoneme/grapheme, e.g., morphs. The rule-based approaches have by far given the best generation performance. A classic example of the rule-based approach is the system MITalk (Allen et al., 1987) which uses a 12,000-word morph lexicon together with a morphological analysis algorithm as the major source of word pronunciation. If no analysis resulted, the word spelling was transformed to a string of phonemes with stress marks by a set of about three hundred to six hundred ordered cyclical rules (Hunnicutt, 1976). Using the MITalk rules alone gave word accuracies ranging from 66% to 76.5% (all phonemes and stress pattern correct). However, the combination of the morph lexicon, the analysis algorithm, and the set of ordered rules achieved a word accuracy of about 97%. Although the process of developing this combination is tedious and time consuming, the high performance level of 97% has not yet been matched by other more automated techniques.

Since the generation of rules is a difficult and complicated task, several research groups have attempted to acquire letter-to-sound generation systems through automatic or semi-automatic data-driven techniques. In the following we will provide a brief sketch. The goal is to provide as little a priori information as possible – ideally, only a set of pairings of letter sequences with corresponding (aligned or unaligned) phone sequences. Training algorithms are then used to produce a mechanism that is applied to predict the most "promising" pronunciation. For example, the induction approach, which attempts to infer letter-to-sound rules from a body of training data, was adopted in (Hochberg et al., 1991; Klatt and Shipman, 1982; Lucassen and Mercer, 1984; Oakey and Cawthorne, 1981; Segre et al., 1983; Van Coile et al., 1992). Hidden Markov modeling was used in (Parfitt and Sharman, 1991; Luk and Damper, 1993). Neural networks were used in the well-known NETtalk system (Sejnowski and Rosenberg, 1987; Lucas and Damper, 1992). Psychological approaches, based on models proposed in the psychology literature, were used in (Dedina and Nusbaum, 1991; Sullivan and Damper, 1992). The case-based reasoning approaches, which generate a pronunciation of an input word based on similar exemplars in the training corpus, was adopted in (Coker et al., 1990; Stanfill, 1987;

Lehnert, 1987; Golding, 1991; Van den Bosch and Daelemans, 1993). Generally speaking, phoneme accuracies of the data-driven systems hover around the low 90 percentages. This roughly translates to $(0.9)^6 = 53\%$ word accuracy, if we assume that an average word is 6 letters long, and the probability of pronouncing each letter correctly in a word is independent of the other letters.

3. Labelling the lexicon

Our experimental data consist of the 10,000 most frequent words appearing in the Brown Corpus (Kucera and Francis, 1967). A *single* phonemic transcription is provided for each word, and the transcription may contain up to three levels of stress: primary ('), secondary (") and reduced (unmarked). The words were also marked morphologically by using special symbols to identify prefix boundaries (=), root-root boundaries (#), derivational suffix boundaries (+), and inflectional suffix boundaries (++). Affixes as well as roots were identified on an etymological basis, by reference to a dictionary (Websters, 1984) in cases of uncertainty. A (non-contiguous) excerpt from the resulting lexicon is shown in Fig. 1. The first part of each entry is the word spelling; the second part is the phonemic transcription; and the third part is the morphological composition.

```
BANKERS B'aenxKerZ * BANK++ER++S ##
BANKRUPTCY B'aenxKRahPTSiy * BANK#RUPT+CY ##
BANKS B'aenxKS * BANK++S ##
BANNER B'aeNer * BANNER ##
BAPTIST B'aePTixST * BAPT+IST ##
BAPTIZE B'aePT"ayZ * BAPT+IZE ##
BAPTIZED B'aePT"ayZD * BAPT+IZe++ED ##
...
BEAUTIFULLY B'yuTihFaxliy * BEAUT+y+FUL+LY ##
...
BEGGED B'ehGD * BEGg++ED ##
...
BINOMIAL B"ayN'owMiyaxl * BI=NOM+IAL ##
...
CHROMATOGRAPHY KR"owMaeT'aaGRaeFiy *
  CHROMATO=GRAPH+Y ##
...
GOSPEL G'aaSPehl * GOd#SPEL ##
GOSSIP G'aaSihP * GOd#sSIP ##
...
HANDFUL hh'aeNDFuhl * HAND#FULl ##
HANDICAP hh'aeNDiyK"aeP * HAND#In#CAP ##
```

Fig. 1. Excerpts from the lexicon.

An example of conformity to etymological principles in data preparation can be noted by comparing the words "banker" and "banner". While the inflectional suffix "er" in the word "banker" denotes someone (in a position of authority) working at a bank and also appears in similar words such as "baker" and "builder", the most common usage of the word "banner" does not have a meaning similar to "someone who bans". Rather, the word is derived from the Middle English "banere" and Old French "banere, baniere", and is therefore identified as an integral root.

Information about changes in spelling due to morph composition is also retained in the lexical entry by the use of lower case letters. The final "e" of the derivational suffix "ize" in the word "baptized" is given in lower case since it is redundant with the "e" of the inflectional suffix "ed". Somewhat rarer instances of deletion are seen in the word "handful", which comes from "hand" and "full", and in the word "handicap", which comes from the three words "hand", "in" and "cap". On the contrary, in the word "begged", the "g" in lower case appears in the spelling of the word but not in the spelling of the root.

Prefixes are later distinguished as frequent (usually one-syllable) prefixes, such as "bi" indicating "two" in the word "binomial" or as root-derived prefixes such as "chromato" in "chromatography". Root-derived prefixes can be automatically relabelled as a root morph followed by a "connecting vowel". Connecting vowels serve the purpose of encoding stress difference. As an example, contrast the stress patterns of the words "speed", "meter" and "speedometer".

The markings in the lexicon, however, are not fully stable. Often it is difficult to determine exactly how a word should be marked, because word segmentation according to morphological decomposition disagrees with that according to syllabification. This is particularly true when determining the left or right syllable affiliation of ambisyllabic consonants. Since markings according to syllabification can be quite clearly specified by rules such as the Maximal Onset Principle and

Stress Resyllabification [1], we try to incorporate special markings to preserve the underlying morphological decomposition. For example, the letter "c" in "dedicated" belongs to the root of the word, but is assigned to the following syllable through a Stress Resyllabification rule.

4. Initial approach

As was mentioned previously, our initial approach involves specifying some structural regularities of English word morphology and syllabification in terms of a simple rule set. These grammar rules are then used by a parser to analyze the structure of a given English word.

Word and sub-word structures were defined from an analysis of the root and affix morphs for all the entries in the lexicon. Roots were defined as having one or two syllables. Roots with more than two syllables are treated as compound words. First syllables were found to have the form onset-nucleus-coda where only the nucleus was a compulsory element. This nucleus was usually a single vowel phonemically (a vowel or vowel digraph orthographically) optionally followed by a liquid. The optional second syllable of a root was sonorant: an enumeration of such sonorant sequences was collected. Prefixes, as mentioned previously, could be of two types: there is an enumerable set of single-syllable prefixes, as well as a set of multi-syllable prefixes each of which is formed by adjoining a "connecting vowel" to a root or whole word structure. Suffixes were enumerated, and could begin with a connecting vowel. Units were specified as members of the sets of onsets, nuclei, coda, sonorant syllables, enumerable (non-root-derived) prefixes, enumerable affixes and connecting vowels. Some examples are given in Table 1.

Table 1
Examples of units in root and prefix structures

Structure	Unit	Word examples
one syllable root	onset-nuc-coda	br-ea-k, b-ar-k
	onset-nuc	p-ie
	nuc-coda	e-gg
	nuc	i, a
two syllable root	(first syl.) – (sonorant syl.)	bott-le, bann-er, pill-ow
single syllable prefix	bi	bi-nomial
root-derived prefix	bio	bi-o-graphy

Possible correspondences of English spelling and pronunciation for these units were then collected from the data and expressed as definitions. A formalism was defined that allowed alternative spellings/pronunciations including optional letters/phonemes to be given by a single sequence.

Possible transitions in the grammar were then defined as pair-wise unit-to-unit sequences. Sequential constraints were facilitated by the definition of special categories of phonemes (e.g., back vowel, lax vowel, digraph, consonant cluster) and affixes (e.g., adjectival suffix, non-final suffix, derivational suffix).

The original version of the TINA natural language framework was used to parse either the letter string or the phoneme string corresponding to a word. TINA augments the context-free rules with other probabilistic constraints which are automatically trainable from training data. In a typical context-free rule, *Parent → Left-sibling Right-sibling*, the probabilities are attached to the sibling–sibling transition, so that the right sibling in a rule can be predicted based on the left context (left sibling) and the parent. The generic [START] category is used as the left context for the probability model to predict the beginning of a rule.

Parsing the orthographic representation of the 10,000 words from which the morphs and subsyllabic units were drawn resulted in 96.2% coverage. The remaining non-parsable words were mainly compounds (2.3%) and names (1.0%). Words which failed to parse due to sparse train-

[1] The Maximal Onset Principle states that the number of consonants in the onset position should be maximized when phonotactic constraints permit, and Stress Resyllabification refers to maximizing the number of consonants in stressed syllables.

ing data constituted only 0.4% of the lexicon. Although many pronunciations were unique and correct, these figures do not reflect the success of a single unique pronunciation in many other cases.

A major problem with this approach is that while TINA is adequate for parsing sentences using a semantic grammar, using it as a word-level parser led to a great deal of overgeneration. Consequently, the task of applying phonological rules at a higher level in the form of filters was initiated to remove inappropriate correspondences. One class of filters, for example, was designed to choose the correct vowel quality (and thus correct stress) in words containing stress-affecting suffixes. Such a filter would, for example, eliminate all but the lax form of the phoneme corresponding to the vowel in the syllable preceding the derivational suffixes +ic and +ity as in the words "atomic" and "authenticity" (where the voiced version of the consonant in +ic could be filtered out before a front vowel as well). This task, however, proved to be quite formidable, since it made rule specification more tedious and less well-defined. Also, this configuration led to a heavy computational load in exploring partial theories that would later fail. Therefore, a better solution was sought.

5. Refined approach

Through our attempts to develop the initial grammar for letter-to-sound productions, it became evident that in order to simplify our rule set and avoid the use of complex filters, a more constraining probability model was needed to control overgeneration problems during parsing. We found that, if we could arrange the parse trees in a regular layered structure, then it would be straightforward to incorporate *across-rule* constraints by conditioning probabilities on the left sibling (instead of the [START] category) even when that sibling did not share the same parent. We reported on a successful application of this new "layered bigrams" formalism to sentence level parsing in (Seneff et al., 1992). In the remainder of this paper we will report on an alternative approach to the letter-to-sound problem which makes use of a greatly simplified rule set in conjunction with a very powerful probability model derived from this "layered bigrams" strategy. This new approach enforces an extremely regular parse tree structure, with each layer in the hierarchy being assigned to a particular linguistic role. Furthermore, the left-context used in the probability model includes not only the left-sibling, but also its complete history [2], to provide additional constraints for avoiding overgeneration.

5.1. The new representation

A variety of linguistic knowledge sources are collectively used to describe English orthographic-phonological regularities. Each knowledge source only involves a small inventory of labels. These linguistic levels and their associated labels are defined from top to bottom in the hierarchy as follows:

1. top-level ([WORD] category, but may encode part-of-speech information);
2. morphs (prefix, root, suffix);
3. syllable stress (primary [SSYL1], secondary [SSYL2], reduced [SYL]);
4. subsyllabic units (onset, nucleus, coda);
5. manner classes (stops, fricatives, vowels, etc.);
6. place and voicing classes encoded as phonemes;
7. letters.

As an illustration, Fig. 2 shows the representation of the word "dedicated". The higher levels in the hierarchy encode longer distance constraints, while the lower levels carry more local constraints. The layer of terminal nodes, which currently represent letters in a word spelling, can conceivably represent phones in the word pronunciation as well. As such, the task of letter-to-sound generation should involve deriving the higher levels in the hierarchy based on an input spelling, and subsequently generating a phone sequence (pronunciation) while making use of all available hierarchical constraints. This process

[2] The set of parse tree categories contained in the path up to the top-level root node.

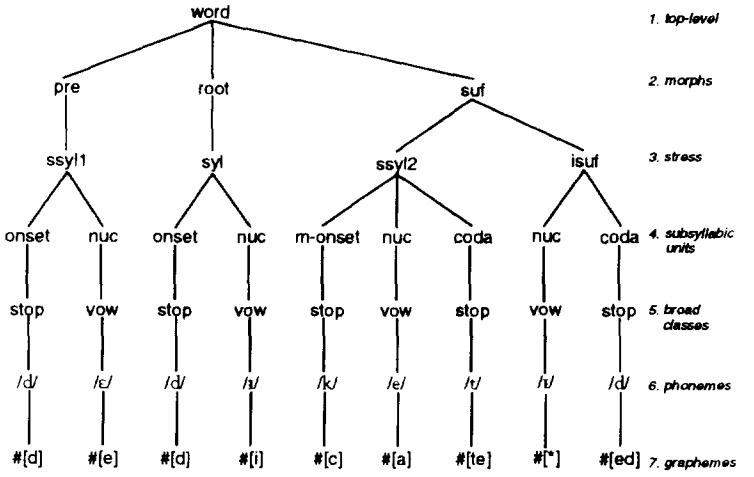

Fig. 2. Parse tree for "dedicated" with different linguistic layers indicated numerically.

can also be used for sound-to-letter generation, with the exception of swapping the input/output specifications. Our current system does not involve phones, because our experimental corpus provides phonemic transcriptions only. However, if the terminal nodes were actually dual in nature, the hierarchical representation should be able to capture phonological rules between layers 6 (phonemes) and 7 (phones).

Although most of our labels are easily understood, a couple of special annotations that appear in Fig. 2 should be explained. The graphemic "place-holder", (*) in layer 7, is introduced here to maintain consistency between the representations of the words "dedicate" and "dedicated", where the letter "e" in the inflectional suffix [ISUF] "ed" is dropped when it is attached after the final silent "e" in "dedicate." Also noteworthy is the special [M-ONSET] category, which signifies that the letter "c" should belong to the root "-dic-", [3] but has become a moved onset of the next syllable due to syllabification principles such as the Maximal Onset Principle and the Stress Resyllabification Principle.

5.2. The layered bigrams paradigm

In order to generate hierarchical representations for words, we adopt a probabilistic parsing algorithm based on the layered bigrams (Seneff et al., 1992). The layered bigrams have previously been used to parse sentences in the Air Travel Information Service, or ATIS (Hirschman et al., 1993) domain, where parsing takes place in a top-down and left-to-right fashion. The new version of the layered bigrams is adapted to parse bottom-up left-to-right for the current subword application. The two main changes in the new algorithm were (1) to allow the left context to take into account the entire column above the left sibling in question, and (2) to predict the right siblings along a given column from bottom to top rather than from top to bottom. [4] The algorithm is able to capture long distance constraints through enforcing local constraints hierarchically–sonority sequencing in syllable layers (3, 4 and 5), phonotactics in the phoneme layer (layer 6), and graphemic constraints in the letter layer (layer 7). If the terminals (layer 7) were to

[3] According to Webster's New World Dictionary, the root of "dedicated" is "-dic-", which is derived from the Latin Word "dicare".

[4] A bottom-up procedure is difficult when parsing sentences into linguistic structures due to movement phenomena (e.g., gaps and unification features) which do not occur below the word level.

Table 2
A few examples of generalized context-free rules for different levels in the hierarchy

word	→ [prefix] root [suffix]
root	→ ssyl [syl]
stressed-syl.	→ [onset] nucleus [coda]
nucleus	→ vowel
nasal	→ (/m/, /n/, /ŋ/)
/m/	→ ("m" "me" "mn" "mb" "mm" "mp")

represent phones also, phonological rules should be captured as well. A significant portion of the higher level layers of the parse tree is shared by large groups of words, leading to efficient parsing.

In order to train the probabilities in the layered bigrams, the algorithm calls for a set of training parse trees. About a hundred context free rules were written to parse the entire training corpus of 8,000 words into the format shown in Fig. 2. As can be seen from the examples in Table 2, the rules are very general and straightforward. It should be noted that the terminal rules simply enumerate all possible spellings that can associate with a given letter or letter sequence, without regard to context. Context conditions are learned automatically through the probabilistic training step.

The linguistic knowledge encoded in these context-free rules, together with other regularities present in the training parse trees but not explicitly specified, are automatically deduced by the training procedure and transformed into a set of trained probabilities. Details of the training and testing procedures are provided in the following.

5.2.1. *Training procedure*

The layered bigrams system trains on a set of parse trees, each of which corresponds to a word in the training corpus. The generation of these parse trees is accomplished by boot-strapping with the TINA parser (Seneff, 1992) operating with the set of generalized context-free rules. These context-free rules serve to incorporate linguistic knowledge into the parser. Fig. 3 shows the parse tree representation of the word "predicted", while Fig. 4 shows the representation in "layered bigrams" format.

The basic 4-tuple used in the analysis of the training data is

1. Left-History (LH),
2. Left-Sibling (LS),
3. Right-Parent (RP),
4. Right-Sibling (RS).

Using Fig. 4 as an example, if we regard the current node (or RS) as the terminal node r in column 2, then the entire column 1 constitutes its left-history, i.e., [WORD, PREFIX, UNSTRESSED SYLLABLE, ONSET, STOP, /p/, p]. The left-parent and left-sibling are, respectively, the phoneme /p/ and the letter "p" in column 1, while the right-parent is the phoneme /r/ in column 2. Generally, the left and right parents may or may not be identical. Notice that in this example the left-parent is different from the right-parent. This 4-tuple derives from two context-free rules: /p/ → "p", and /r/ → "r".

As another example, consider columns 4 and 5 in Fig. 4. If we regard RS = NUCLEUS in column 5, then LH = [WORD, ROOT, STRESSED SYLLABLE, ONSET] in column 4, LS = ONSET, LP = STRESSED SYLLABLE and RP = SAME (as LP). Notice that in this case, the left-parent "holds" (i.e., LP = RP), because this 4-tuple is derived from a single context-free rule, namely, STRESSED SYLLABLE → (ONSET) NUCLEUS

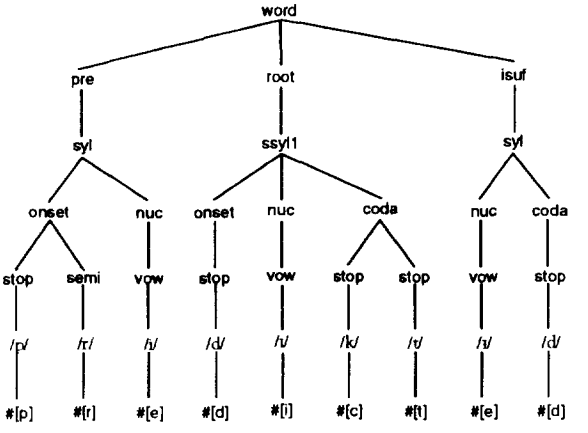

Fig. 3. A parse tree generated by TINA for the word "predicted".

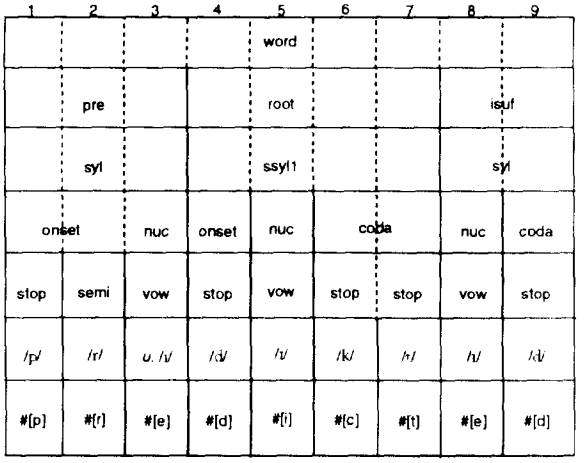

Fig. 4. A parse tree for the word "predicted" in layered bigrams format.

(CODA). In other words, a 4-tuple encodes "within-rule" constraints in a derivation if LP = RP. Otherwise, if LP ≠ RP, the 4-tuple corresponds to "across-rule" constraints.

The set of probabilities that we compute is intended to capture the constraints provided by the context-free rules as well as other regularities inherent in the training parse trees but not explicitly specified. These training probabilities are computed by tallying counts and then normalizing them by the total counts. Each word in the lexicon is counted as a single occurrence. [5] The set of training probabilities includes:

1. *start terminal unigram* $P_{\text{StartUnigram}}$ – this is the unigram probability over all the terminals that can start a word. In the letter-to-sound generation case, the start terminal is a grapheme, e.g., the letter "p" starts the word "predicts", and the grapheme "ph" starts the word "philosophy".
2. *start column prediction probability* P_{StartCol} – this is the bottom-up prediction probability given that we are at the start column of the word. It is the product of the start terminal unigram and all the bottom-up prediction probabilities in the start column, i.e.,

$$P_{\text{StartCol}} = P_{\text{StartUnigram}} \times Pr(RP \mid RS = \text{the start terminal}, LH = START) \times \cdots \times Pr(RP = WORD \mid RS, LH = START).$$

3. *column advance probability* $P_{\text{ColAdvance}}$ – this is the bigram probability over all the terminals than can follow the current column, i.e., Pr(RS = next terminal | LH). The next terminal may be an END node.
4. *column probability* P_{Col} – this this is the product of the column advance probability and all the bottom-up prediction probabilities in the current column. Prediction in the layered bigrams is modified to drive entirely bottom-up so that the "within-rule" statistics and "across-rule" statistics are shared. The bottom-up prediction probability Pr(RP | RS, LH) makes a prediction using the entire left-history, and the current (right-sibling) category as its context.

$$P_{\text{Col}} = P_{\text{ColAdvance}} \times Pr(RP \mid RS = \text{current terminal}, LH) \times \cdots Pr(RP = SAME \mid RS, LH).$$

Notice that we stop accumulating column prediction probabilities once we reach RP = SAME. This is because from then on the right-history merges with structures which are already in place in the left-history due to previous derivations from the context-free rules.

5.2.2. Testing procedure

Given a test spelling or pronunciation, we attempt to construct a layered parse structure in a bottom-to-top, left-to-right fashion. A theory is first proposed by constructing the start column and computing the corresponding start column probability, and then it is pushed on the stack to become a partial theory with a stack score. At each iteration of the algorithm, the stack is sorted,

[5] Another possibility is to take the word frequencies into account.

and the partial theory with the highest stack score is popped off the stack and advanced by one column. At the advancement, we check to see if the last column of the partial theory is valid top-down, and if it can reach the next terminal to follow. If either one of these conditions is not satisfied, the partial theory is eliminated. Otherwise, this newly expanded partial theory is pushed onto the stack with an updated stack score. The iteration continues until one or more complete theories (theories whose last column contains an END node) is popped off the stack.

It can be seen that the layered bigrams algorithm attempts to construct the entire parse tree structure based on some very local probabilistic constraints within adjacent layers. With the various different layers in the hierarchy, long-distance constraints are *implicitly* enforced. The layout of the hierarchical structure also enables us to *explicitly* enforce additional long-distance constraints through the use of filters. To illustrate this, consider the example of bisyllabic words with a [PREFIX]–[ROOT] morphology, where the nouns often have a STRESS–UNSTRESS pattern, while the verbs have an UNSTRESS–STRESS pattern (consider "permit", "record", etc.). There are also several known "stress-affecting suffixes" which tend to alter the stress contour of a word in a predictable way (consider "combine" versus "combination"). For these examples, we can potentially utilize stress filters to eliminate any partial (or complete) theories which do not have the correct stress pattern. Similarly, it is possible to use morph filters to eliminate theories which contain illegal morph combinations.

Apart from allowing additional constraints to be enforced, the flexibility of the layered bigrams algorithm also enables us to relax some constraints. This can be done by sharing probabilities among different left histories, to legitimize reasonable sibling–sibling transitions which were not foreseen in the training data. More specifically, probability sharing is accomplished by conditioning the column advancement probabilities on a partial left-history (e.g. disregarding the grapheme terminal) instead of the entire left-history. In this way, we are able to increase the coverage of the parser.

5.2.3. An efficient search algorithm

During the parsing process, all partial theories are sorted in a stack, and the search for the best-scoring theory is guided by the stack score. Consequently, the evaluation function for computing the stack score is important. In order to avoid expensive computation to obtain a tight upper bound for the look-ahead score of a partial theory, we are currently using an evaluation function which invokes a score normalization mechanism. This mechanism aims at generating stack scores within a certain numeric range, and thus strives to achieve a fair comparison on the goodness of a partial path between the shorter partial paths and the longer ones. Scoring normalization may be accomplished by an *additive* correction factor in some cases, and a *multiplicative* correction factor in others. In our implementation, we use a "fading" scheme as shown in

$$\hat{f}(n) = \alpha \hat{f}(n') + (1-\alpha)p(n', n), \qquad (1)$$

where $\hat{f}(c)$ is the stack score from the [START] column to the current column c, c' is the column preceding c in the parse tree, $p(c', c)$ is the log-likelihood associated with extending the parse tree from c' to c, and α is some fading factor $(0 < \alpha < 1)$. The idea is to have the stack score carry short term memory, where the current node always contributes towards a certain portion of the stack score (according to some pre-set weight of $\alpha = 0.95$), while the remaining portion associated with the past gradually fades away exponentially, so that the distant past contributes less to the stack score than the recent history, and the score tends to remain quite stable over time. The outcome of this search is that the ordering tends to place together parse theories with similar distant columns and different recent columns.

If multiple hypotheses are desired, the algorithm can terminate after a desired number of complete hypotheses have been popped off the stack. In addition, a limit is set on the maximum number of theories (partial and complete) popped off the stack. The complete theories are subsequently reranked according to their actual parse score (no fading). Though our search is inadmissible, we are able to obtain multiple hypotheses

inexpensively. Our performance will be reported in the next section.

6. Experimental results

Our experimental corpus consisting of the 10,000 most frequent words appearing in the Brown Corpus (Kucera and Francis, 1967), have been divided into 3 subsets. The words were arranged alphabetically, and every tenth word was set aside as a future test set. Subsequently, every tenth word of the remaining set was set aside for a development test set. The rest of the words (about 8,000 in total) are used for training. The preliminary results given in this paper are based on the development test set only. Subsequent improvements to the systems and the future development of a robust parsing mechanism for handling nonparsable words will ensue. The best configuration of our system, based on the attainment of the highest performance and broadest coverage on the development test set, will ultimately be tested on the future test set.

The two criteria which we use for evaluating spelling-to-pronunciation accuracies are similar to those used in the other systems reported previously:

1. *Word accuracy.* In the case of spelling-to-pronunciation generation, one can perform a match between a generated phoneme string and the reference phoneme string from the lexical entry of the word. Our experimental corpus provides only a *single* reference pronunciation per word. Generation is correct only if there are no discrepancies between the two phoneme sequences. In the case of pronunciation-to-spelling generation, a similar match is performed between the two letter sequences. This is a strict evaluation criterion which permits only a single pronunciation per word and does not allow the generated output to contain any deviation from the reference output.
2. *Phoneme accuracy.* In addition to whole-word accuracy, it is also possible to measure phoneme/letter accuracies. Phoneme accuracy should be a good evaluation criterion to use for spelling-to-pronunciation generation. [6] Such measures can be computed using a dynamic programming algorithm to align the generated string with the "correct" string (provided by the lexicon). The alignment selected minimizes the number of insertion, deletion and substitution operations necessary to transform the generated string to the reference string. The accuracy is computed by subtracting the sum of the insertion (I), deletion (D) and substitution (S) error rates from 100%, i.e.,

$$\text{accuracy} = 100\% - (I + D + S)\%. \qquad (2)$$

This evaluation criterion is the one adopted by NIST (The United States National Institute of Standards and Technology) for measuring the performance of speech recognition systems.

As was mentioned earlier, one should be careful when comparing the performance of different systems based on the given reported accuracy values. This is because the training and test sets, as well as the inventory of phonemes/phones, vary from one system to another. Discrepancies also occur in the evaluation methods. Multiple pronunciations may be provided for a single word in some cases, and insertion errors or stress errors may or may not be taken into account. In addition, phoneme/phone accuracies may be computed per phoneme/phone or per letter (in measuring the accuracy per letter, silent letters are regarded as mapping to a [NULL] phoneme/phone). We expect the accuracy per letter to be generally higher than accuracy per phoneme/phone, because there are generally more letters than phonemes/phones per word, and the letters mapping to the generic category [NULL] would usually be correct. Although our results reported in the following are based on the *per phoneme* criterion, empirical results comparing accuracy per letter and accuracy per phoneme/phone will also be provided.

Our system uses an inventory of 52 unique phonemes, which includes several unstressed

[6] Similarly, for pronunciation-to-spelling generation, letter accuracy should be used.

Table 3
Experimental results – word and phoneme accuracies for training and testing data

Accuracy		top choice correct	top 5 correct	top 10 correct
train	word	77.3%	93.7%	95.7%
	phoneme	94.2%	–	–
test	word	69.3%	86.2%	87.9%
	phoneme	91.7%	–	–

vowels and pseudo diphthongs such as /or/. [7] Stress errors are accounted for to a certain extent, since we distinguish between the stressed and unstressed realizations of some of our vowels. The parser was set to terminate after obtaining 30 complete hypotheses, and the maximum number of theories (partial and complete) popped off the stack is constrained to 330. Following this, we re-ranked the theories according to their actual parse score. [8] Experimental results are shown in Table 3. "Top-choice" word accuracy is the percentage of words for which the top-ranking complete theory contains the correct phoneme sequence. "Top N" word accuracy refers to the percentage of words for which the correct phoneme sequence is contained in one of the first N complete theories. Fig. 5 is a plot of cumulative percent correct of whole word theories as a function of the ranks of the theories after re-scoring. Although 30 complete theories were generated for each word, no correct theories occur beyond $N = 15$ after resorting. There were 46 non-parsable words in the development test set – these are words for which the parser cannot generate any complete theories. By incorporating a simple "backoff" mechanism, [9] we managed to reduce the number of non-parsable words to 18, containing approximately equal proportions of

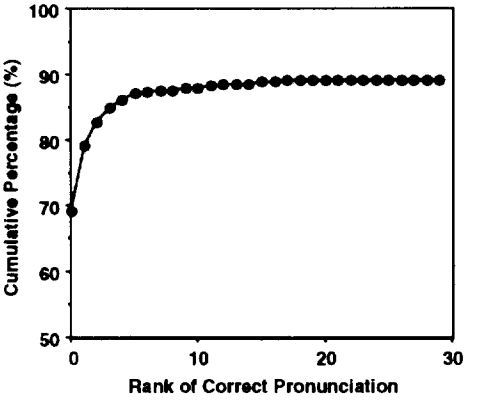

Fig. 5. Percent correct whole-word theories as a function of N-best depth for the test set.

compound words, proper names and words that failed due to sparse data problems. [10]

Notice that the asymptote of the cumulative plot in Fig. 5 lies around 88%. Computation of the cumulative percentage includes words for which the generated output is "correct", as well as other words for which the "correct" theory does not surface with top rank, but is among the N-best. In some cases, different parse trees in the N-best pool may give the same output pronunciation, but with different higher level linguistic analyses. Therefore another possible method of N-best rescoring is to sum the independent probabilities of the different parse theories with identical phonemes and re-rank the generated pronunciations. In order to retrieve the "correct" hypothesis from the N-best pool, we can perhaps use a better stack criterion to target admissibility and curb search errors. We can also eliminate systematic errors and refine generation outputs by post-processing with additional contextual information. The fraction of the test set lying above the asymptote (about 12%) in Fig. 5 consists of words for which a correct pronunciation did not emerge as one of the output complete theories.

[7] The complete set of phonemes is included in Appendix A.

[8] Since our search is inadmissable, the best scoring theory is not always first.

[9] In this case, the column advances probability is conditioned only on the *partial* left-history, where the letter terminal is excluded.

[10] A more recent backoff algorithm achieves 100% coverage (Meng et al., 1994).

We grossly classified the errors into 4 categories. Some generated pronunciations have subtle differences from the given pronunciations, and are essentially correct. Another category has unusual pronunciations due to influences from foreign languages, e.g., "aborigines", "champagne" and "debris". Still another category has generated pronunciations which agree with the regularity of English letter-to-phoneme mappings, but were nevertheless incorrect, e.g., "cushion" /k ʌ šɪn/, "elite" /ɪ l ɑ^{y} t/ and "viscosity" /v i s k o^{w} s ɪ t i^{y}/. The final category are errors attributable to sparse data problems, such as "que" in "braque".

We have also run an experiment comparing the *per letter* and *per phoneme* accuracies by dividing the training set into two portions, training on one and testing on the other. Phoneme accuracy per letter is computed using the alignment provided by the training parse trees. We found that the per letter measurement led to approximately a 10% reduction in error rate.

7. Discussion and future work

Our current work demonstrates the use of a hierarchical framework, which is relatively rich in linguistic knowledge, for bi-directional letter/sound generation. This paper focuses on letter-to-sound generation. The results obtained on a disjoint test set, 69.3% word accuracy and 91.7% phoneme accuracy, seems competitive when compared to the other automatic approaches described previously in Section 2.

In order to demonstrate the bi-directional generation capability, we have also conducted a formal evaluation on sound-to-letter generation. A detailed report can be found in (Meng et al., 1994a). Using the same training and test sets as described here, the word accuracy and letter accuracy obtained are approximately 52% and 89%, respectively. There is much overlap between the problematic words in letter-to-sound generation and the corresponding set in sound-to-letter generation, possibly due to the symmetry imposed by our approach. The overlap implies that improvements made in one generative direction should carry over to the opposite direction as well.

The hierarchical lexical representation used in our approach advocates the use of higher level linguistic knowledge for letter/sound generation. Therefore, it is important for us to assess the relative contribution of each linguistic layer in the hierarchy towards generation, and the relative merits of the overall hierarchical design. We have run a series of letter-to-sound generation experiments which are reported in (Meng et al., 1994b). Our results show that while the broad class layer promotes more sharing of the training data and consequently relaxes some constraints for increased coverage, all the other layers seem to provide additional constraints which are important for generation. Furthermore, comparison with an alternative flat structure shows that the hierarchical representation provides a parsimonious description of English letter-to-sound mappings. Once again, due to the symmetry between letter-to-sound generation and sound-to-letter generation, we expect these findings to be applicable to the sound-to-letter generation task as well. The task of bi-directional letter/sound generation is particularly useful for handling out-of-vocabulary words encountered by speech recognition systems, by generating reasonable pronunciations/spellings for new words and incorporating them into the vocabulary. Our hierarchical lexical representation, being rich in linguistic knowledge, is also potentially useful for other applications in speech synthesis, recognition and understanding. It could serve as a lexical representation for large vocabulary recognition tasks, providing extensive structural sharing among all words. The morphology layer allows us to automatically derive regular inflectional and derivational forms of words, and can potentially provide basic semantic and syntactic information. The layer of manner features can be used for rapid lexical access (fast match) by narrowing down word candidates to a short list which belong to the same cohort (Shipman and Zue, 1982). The framework would also be suitable as a constraining language model in a character recognition task. Viewing the layered bigrams as a predictor of the next character, we can measure the perplexity [11] per character (letter) of the test set to determine its effectiveness. We obtained a perplexity of 8.3 as contrasted with

11.3 for a standard bigram language model, having trained and tested on the same data sets. A lower perplexity implies the provision of more constraints, which is often conducive towards better performance.

Furthermore, the multi-level framework should also be applicable to languages other than English. For example, it should be possible to apply our system methodologies to multilingual systems (Zue et al., 1993) whenever letter-sound correspondences and context interact in the same way as in our current monolingual system (Goddeau, 1994), e.g., generating English name pronunciations in terms of the Japanese Katakana pronunciation alphabet.

Some resemblances can be found between our formalism and another used for Dutch, known as the Speech Maker Formalism (Van Leeuwen, 1993). Speech Maker is used for text-to-speech synthesis, and supports a multi-level synchronized data structure called a *grid*. The grid contains *streams*, or levels which contain information about word accent, word class, and the morphemes, syllables, phonemes and graphemes constituting the word. Each stream is also synchronized with other streams in the grid by sync marks, i.e. the vertical bars as illustrated in Fig. 6, which shows a portion of a grid. Generation in the Speech Maker is achieved by complex, *two-dimensional* rules which can operate on more than one stream at a time, modelled after the DELTA language (Hertz et al., 1973). An example is shown in Fig. 7, which is a rule stating that an "a", followed by any sequence of characters pronounced as a single consonant, followed by an "e" which is root final, should be pronounced as /e/.

Our approach differs from the Speech Maker in that each rule in our rule set specifies the derivation that can take place in one layer only. This is achieved by virtue of the regularity of the

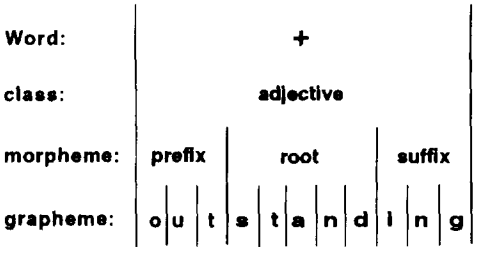

Fig. 6. A portion of the Speech Maker grid representing the word "outstanding".

hierarchical structure and the use of probabilities. The hierarchical structure, in layered bigrams format, is regular in that each column advances uniformly from left to right, and thus synchronization marks are not required. The set of probabilities used in the parser is conditioned upon a very elaborate left context, which includes the entire left history, and thus they provide tight linguistic constraints for generation. Consequently, complex rules may not be necessary

It is also important to note that the hierarchical framework offers the possibility of a uniform approach to all levels of analysis in spoken language systems. The dimensions of the hierarchy, along with the layered bigram formalism, could be further extended upwards to encompass natural language constraints (Seneff, 1992), prosody, discourse and even perhaps dialogue constraints, and augmented downwards to include layers capturing phonetics and acoustics. As such the hierarchy should be a promising means of representation for speech synthesis, recognition and understanding.

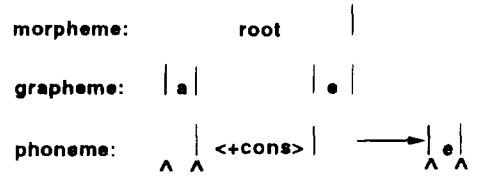

Fig. 7. An example of a two-dimensional rule in Speech Maker.

[11] Perplexity is the average branching factor, or the geometric mean of the number of possible choices for the next character.

Appendix A

The complete set of phonemes used in the system include: /w/, /l/, /l̩/, /h/, /r/, /y/, /ɪ/, /ɛ/, /æ/, /a/, /ʌ/, /ɔ/, /ʊ/, /ɝ/, /z/, /s/, /š/, /ž/ /f/, /θ/, /ð/, /v/, /p/, /t/, /k/, /d/, /b/, /g/, /ǰ/, /č/, /m/, /m̩/, /n/, /ŋ/, /oʷ/, /u/, /ɑʷ/, /e/, /ɑʸ/, /i/, /ɔʸ/, /yu/, /ær/, /ɪr/, /or/, /al/, /ol/, /ar/, unstressed /i/, unstressed /u/, unstressed /o/ and /ə/. (The pseudo diphthongs are /yu/, /ær/, /ɪr/, /or/, /al/, /ol/ and /ar/.)

References

J. Allen, S. Hunnicutt and D. Klatt (1987), *From Text to Speech: The MITalk system*, Cambridge Studies in Speech and Science Communication (Cambridge Univ. Press, Cambridge).

N. Chomsky and M. Halle (1968), *The Sound Pattern of English* (Harper & Row, New York).

C. Coker, K. Church and M. Liberman (1990), "Morphology and rhyming: Two powerful alternatives to letter-to-sound rules for speech synthesis", *Proc. Conf. on Speech Synthesis* (European Speech Communication Association).

D. Conroy, T. Vitale and D.H. Klatt (1986), DECtalk DTC03 text-to-speech system owner's manual, Educational Services of Digital Equipment Corporation, P.O. Box CS2008, Nashua, NH 03061, Document Number EK-DTC03-OM-001.

R. Damper (Forthcoming), "Self-learning and connectionist approaches to text-phoneme conversion", *Proc. 2nd Neural Computation and Psychology Workshop*, edited by J. Levy.

M.J. Dedina and H.C. Nusbaum (1991), "PRONOUNCE: A program for pronunciation by analogy", *Computer Speech and Language*, January, pp. 55–65.

D. Goddeau (1994), Personal communication.

A. Golding (1991), Pronouncing names by a combination of case-based and rule-based reasoning, Ph.D. Thesis, Stanford University.

S.R. Hertz, J. Kadin and K. Karplus (1973), "The Delta rule development system for speech synthesis from text", *Proc. IEEE*, Vol. 11, pp. 1589–1601.

L. Hirschman et al. (1993), "Multi-site data collection and evaluation in spoken language understanding", *Proc. Darpa Human Language Technology Workshop, Princeton, USA*, pp. 19–24.

J. Hochberg, S.M. Mniszewski, T. Calleja and G.J. Papcun (1991), "A default hierarchy for pronouncing English", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No. 9, pp. 957–964.

S. Hunnicutt (1976), Phonological rules for a text-to-speech system, AJCL Microfiche 57.

S. Hunnicutt, H. Meng, S. Seneff and V.W. Zue (1993), "Reversible letter-to-sound sound-to-letter generation based on parsing world morphology", *Proc. Eurospeech, Berlin, Germany*, pp. 763–766.

D. Klatt (1987), "Review of text-to-speech conversion for English", *J. Acoust. Soc. Amer.*, Vol. 82, No. 3, pp. 737–793.

D. Klatt and D. Shipman (1982), "Letter-to-phoneme rules: A semi-automatic discovery procedure", *J. Acoust. Soc. Amer.*, Vol. 72, Fall, Supplement 1, pp. S46, CC4.

H. Kucera and W.N. Francis (1967), *Computational Analysis of Present-Day American English* (Brown University Press).

M. Lehnert (1987), "Case-based problem solving with a Large Knowledge Based of Learned Cases", *Proc. AAAI-87, Seattle, USA*, pp. 301–306.

S.M. Lucas and R.I. Damper (1992), "Syntactic neural networks for bi-directional text-phonetics translation," in *Talking Machines: Theories, Models and Designs*, edited by G. Bailly and C. Benoit (North-Holland, Amsterdam), pp. 127–142.

J. Lucassen and R. Mercer (1984), "An information theoretic approach to the automatic determination of phonemic baseforms", *Proc. Internat. Conf. Acoust Speech Signal Process., San Diego, USA*, pp. 42.5-1–42.5-4.

R. Luk and R. Damper (1993), "Inference of letter-phoneme correspondences with pre-defined consonant and vowel patterns", *Proc. Internat. Conf. Acoust. Speech Signal Process., Minneapolis, USA*, pp. 203–206.

H. Meng, S. Seneff and V.W. Zue (1994a), "Phonological parsing for reversible letter-to-sound/sound-to-letter generation", *Proc. Internat. Conf. Acoust. Speech Signal Process., Adelaide, Australia*, Vol. 2, pp. 1–4.

H. Meng, S. Seneff and V.W. Zue (1994b), "The use of higher level linguistic knowledge for letter-to-sound generation", *Proc. International Symposium on Speech, Image Processing and Neural Networks, Hong Kong*, pp. 670–673.

H. Meng, S. Seneff and V.W. Zue (1994c), "Phonological parsing for bi-directional letter-to-sound/sound-to-letter generation", *Proc. ARPA Human Language Technologies Workshop, Princeton, USA*, pp. 289–294.

S. Oakey and R. Cawthorne (1981), "Inductive learning of pronunciation rules by hypothesis testing and correction", *Proc. IJCAI, Vancouver, Canada*, pp. 109–114.

S. Parfitt and R. Sharman (1991), "A bi-directional model of English pronunciation", *Proc. Eurospeech*, pp. 801–804.

A. Segre, B. Sherwood and W. Dickerson (1983), "An expert system for the production of phoneme strings from unmarked English text using machine-induced rules", *Proc. First European ACL, Pisa, Italy*, pp. 35–42.

T.J. Sejnowski and C.R. Rosenberg (1987), "NETtalk: A parallel networks that learn to pronounce English text", *Complex Systems*, Vol. 1., pp. 145–168.

S. Seneff (1992), "TINA: A natural language system for spoken language applications", *Computational Linguistics*, Vol. 18, No. 1, pp. 61–86.

S. Seneff, H. Meng and V.W. Zue (1992), "Language mod-

elling using layered bigrams", *Proc. ICSLP, Banff, Canada*, pp. 397–320.

D.W. Shipman and V.W. Zue (1982), "Properties of large lexicons: Implications for advanced isolated word recognition systems", *Proc. Internat. Conf. Acoust. Speech Signal Process., Paris, France*, pp. 546–549.

C. Stanfill (1987), "Memory-based reasoning applied to English Pronunciation", *Proc. AAAI*, pp. 577–581.

K. Sullivan and R. Damper (1992), "Novel-word pronunciation within a text-to-speech system", in *Talking Machines, Theories, Models and Designs*, ed. by G. Bailly and C. Benoit (North Holland, Amsterdam), pp. 183–195.

B. Van Coile, S. Lyes and L. Mortier (1992), "On the development of a name pronunciation system", *Proc. ICSLP, Banff, Canada*, pp. 487–490.

A. Van den Bosch and W. Daelemans (1993), "Data-oriented methods for grapheme-to-phoneme conversion", *Proc. Sixth European ACL*, pp. 45–53.

H.C. Van Leeuwen (1993), "Speech maker formalism: A rule formalism operating on a multi-level, syncronized data structure", *Computer Speech and Language*, October, pp. 369–390.

Websters Ninth New Collegiate Dictionary (1984) (Merriam-Webster, Springfield, MA).

V. Zue, J.R. Glass, G. Flammia, D. Goddeau, D. Goodine, C. Pao, M. Phillips, J. Polifroni and S. Seneff (1993), "Multilingual human–computer interactions: A case study in the VOYAGER domain", *Proc. ATR Internat. Worshop on Speech Translation, Kyoto, Japan*, pp. 7.1–7.16.