Algorithms for Grapheme-Phoneme Translation for English and French: Applications for Database Searches and Speech Synthesis

Michel Divay*
Université de Rennes

Anthony J. Vitale†
Digital Equipment Corporation

Letter-to-sound rules, also known as grapheme-to-phoneme rules, are important computational tools and have been used for a variety of purposes including word or name lookups for database searches and speech synthesis.

These rules are especially useful when integrated into database searches on names and addresses, since they can complement orthographic search algorithms that make use of permutation, deletion, and insertion by allowing for a comparison with the phonetic equivalent. In databases, phonetics can help retrieve a word or a proper name without the user needing to know the correct spelling. A phonetic index is built with the vocabulary of the application. This could be an entire dictionary, or a list of proper names. The searched word is then converted into phonetics and retrieved with its information, if the word is in the phonetic index. This phonetic lookup can be used to retrieve a misspelled word in a dictionary or a database, or in a text editor to suggest corrections.

Such rules are also necessary to formalize grapheme-phoneme correspondences in speech synthesis architecture. In text-to-speech systems, these rules are typically used to create phonemes from computer text. These phonemic symbols, in turn, are used to feed lower-level phonetic modules (such as timing, intonation, vowel formant trajectories, etc.) which, in turn, feed a vocal tract model and finally output a waveform and, via a digital-analogue converter, synthesized speech. Such rules are a necessary and integral part of a text-to-speech system since a database lookup (dictionary search) is not sufficient to handle derived forms, new words, nonce forms, proper nouns, low-frequency technical jargon, and the like; such forms typically are not included in the database. And while the use of a dictionary is more important now that denser and faster memory is available to smaller systems, letter-to-sound still plays a crucial and central role in speech synthesis technology.

Grapheme-to-phoneme technology is also useful in speech recognition, as a way of generating pronunciations for new words that may be available in grapheme form, or for naive users to add new words more easily. In that case, the system must generate the multiple variations of the word.

While there are different problems in languages that use non-alphabetic writing systems (syllabaries, as in Japanese, or logographic systems, as in Chinese) (DeFrancis 1984), all alphabetic systems have a structured set of correspondences. These range from the trivial in languages like Spanish or Swahili, to extremely complex in languages such as English and French. This paper

* Université de Rennes, Institut Universitaire de Technologie, B.P. 150, 22302 Lannion, France. E-mail: divay@iut-lannion.fr
† Digital Equipment Corporation, 200 Forest St. (MRO1-1/L31), Marlborough, MA 01752-3011. E-mail: vitale@dectlk.enet.dec.com

will outline some of the previous attempts to construct such rule sets and will describe new and successful approaches to the construction of letter-to-sound rules for English and French.

1. Introduction and Historical Background

The interest in letter-to-sound rules goes back centuries and can be found (in relatively unsystematic descriptions) in many of the older descriptive grammars of languages such as English and French. The paucity of literature in grapheme-to-phoneme translation is partially due to the fact that the field of linguistics, and in particular, descriptive linguistics, has traditionally shied away from the writing system (except as a study in its own right) since the phonological system was considered of primary importance. Papers on the subject are rarely found in linguistics journals. Nevertheless, there have been some important studies done on grapheme-phoneme correspondences in past years; for English: Ainsworth (1973), Bakiri and Dietterich (1991), Bernstein and Nessly (1981), Elovitz et al. (1976), Hertz (1979, 1981, 1982, 1983, 1985), Hunnicutt (1976, 1980), Levin (1963), McCormick and Hertz (1989), McIlroy (1974), O'Malley (1990), Venezky (1962, 1967a, 1967b, 1967c, 1970), Venezky and Weir (1966), Vitale (1991), Weir (1964); for French: Aubergé (1991), Béchet, Spriet, and El-Bèze (1996), Catach (1989), Catach and Catach (1992), Cotto (1992), Divay (1984, 1985, 1990a, 1990b, 1991, 1994), Laporte (1988), Prouts (1980), Yvon (1996).

Some of these studies (Weir 1964; Venezky 1966, 1970) were more descriptive in nature and represented a solid base of data from which a rule set could be built. These works consisted of tables of correspondences and examples of words containing these correspondences. These studies made use of phonetic, phonemic, or even morphophonemic form such as palatalization (*credulity, cuticle,* etc.), morphophonemic alternation (*symmetry* vs. *symmetric*) and even morphology (*singer* vs. *finger*). Other studies included pause (McIlroy 1974) and even syntactic information (Divay 1984, 1985).

More recent studies have attempted to use learning algorithms to incorporate pronunciation by analogy (Dedina and Nusbaum 1991), a neural network or connectionist approach to the problem (Sejnowski and Rosenberg 1986; Bakiri and Dietterich 1991; Gonzalez and Tubach 1982; Lucas and Damper 1992), automatic alignment by an induction method (Hochberg et al. 1991); a computational approach (Klatt and Shipman 1982; Klatt 1987), an information theoretic approach (Lucassen and Mercer 1984), hidden Markov models (Parfitt and Sharman 1991), and a case-based approach (Golding 1991). Some have even developed a bidirectional approach of letter-to-sound as well as sound-to-letter (Meng 1995), which is a hybrid of data-based and rule-driven approaches and is also useful for automatic speech recognition. This paper will focus on a rule-based approach, as for example in Allen (1979). Divay (1984, 1985, 1990a, 1990b, 1991, 1994), and others, all of which are essentially knowledge-rich expert systems.

The various attempts at rule formulation were related to differences in the phonemic inventory, the number of rules, the type and format of rules, and even the direction of parse of the rules (whether they were scanned from left to right or from right to left). Different approaches were also taken in the size of the dictionary, the algorithm used to scan or rescan the dictionary (if one was used), the methods for determining lexical stress placement, the amount of morphological analysis used, and the difficulties in the prediction of the correct phonemic form of homographs.

Part of the educational process for a child is learning to read, and educational literature is filled with disparate pedagogical approaches to this problem. Developing a letter-to-sound rule set in software is essentially teaching the computer how to read (pronounce) a language. The difficulty in developing an accurate algorithm to perform this task is directly proportional to the fit between graphemes and corresponding phonemes as well as the allophonic complexity of the language in question.

2. Dictionary versus Letter-to-Sound Rules

Any procedure to convert text into phonemes would necessarily make use of a lexical database or dictionary to provide for lookup of words prior to letter-to-sound conversion. Such a database typically consists of words that exhibit unusual stress patterns (for languages such as English), and of unassimilated or partially assimilated loanwords including place names and personal names that do not fit into the canonical phonological or phonotactic form of the language.

Memory is increasingly less expensive and we now have the capability to store in memory a large number of words (along with their phonetic equivalent, grammatical class, and meaning). Why not then store all words (or certainly all of the words that would be commonly encountered in text) in memory? First, if we include derived forms and technical jargon, there are well over three-quarters of a million words in the English or French language. It would be an extremely difficult task to create such a list. More importantly, new words come into the language every day and from these are generated many derived forms. Lastly, when we factor in items that may not even be found in a dictionary, such as proper nouns (first names, surnames, place names, names of corporations, etc.), the necessity of a rule-governed approach quickly becomes apparent. For example, there are roughly 1.5 million different surnames in the US alone (Spiegel 1985; Spiegel and Machi 1990; Vitale 1991); moreover, one-third of these surnames are unique in that they are singletons. In fact, at this stage in the technology, it is still the rule set and not the dictionary that is the more dominant, although this is beginning to change, primarily due to the need for more complex lexical entry containing information on syntax, semantics, and even pragmatics for more natural prosodics in text-to-speech tasks.

It is difficult and time consuming to place all derived forms in the dictionary, including singular and plural forms and all verb affixes, especially for a language like French where a verb can expand, depending on the conjugation, into about fifty strings consisting of the root plus suffixes. Code could, of course, be added in the dictionary modules providing information on how to form the plurals or conjugations. The lookup procedure could then strip some of the affixes to retrieve the root in the dictionary.

There do exist letter-to-sound systems based on very large dictionaries (for French, see Laporte [1988]) but they require a great deal of memory, especially if the lexical entries contain graphemic, phonetic, syntactic, and semantic information. The main advantage is that this dictionary can then be used to drive a sentence tagger and parser necessary for improving intonation and naturalness for speech synthesis. This universal electronic dictionary could also be used for speech recognition and machine translation. Today, most speech synthesizers do not include such a large dictionary, which, in any case, must be complemented by a set of rules just in case the word or the proper name is not in the dictionary.

3. Grapheme-to-Phoneme Conversion Problems for Both English and French

In this section, we describe the problems encountered when converting from graphemes to phonemes for English and French. Some problems are similar in both languages, others are specific to one language or the other.

3.1 History Results in an Incoherent Letter-to-Sound System for Both English and French

English and French are difficult languages to construct letter-to-sound rules for. Grapheme-phoneme correspondences in English are complex, primarily due to nonlinguistic factors related to the aftermath of the Norman invasion (1066 A.D.) as well as various waves of immigration into anglophone countries such as the UK and the US. The sound system has undergone many shifts, including the addition of the /f/-/v/ phonemic distinction under the influence of massive borrowing from French, the Great Vowel Shift (1400–1500 A.D.) (Wells 1982), and others. It has been estimated that external borrowing has been so extensive that English has retained only about 25% of its original Germanic lexicon (Ben Crane 1981). A quick scan of a text in Old English is enough to convince the reader of this. Since the late Middle Ages, due to the availability of printed materials, dictionaries and grammars, the orthographic system has remained virtually unchanged.

Letter-to-sound transcription for French is complex because a gap has slowly appeared between the orthographic and phonetic forms at the inception of what we would call French (about the 12th Century), and the orthographic and phonetic forms at the current time. Prior to the Roman invasion in 51 B.C., inhabitants of northern France (called at that time *Gallia*) spoke a Celtic language. The Roman occupation (which lasted until 476 A.D.) resulted in well-educated people speaking Latin; for others, a language derived from Latin. After the Roman pullback, the language continued to evolve in different ways, especially since few people knew how to read. The first written texts in the new language, called *Roman* ([romã]), appear only in the 9th Century. Then official papers written in Latin began to be written in *Roman*. Up until the 17th Century, standardization in spelling had been vague until dictionaries, schools, and laws enforced a standard spelling (Burney 1955; Catach 1978; Thimonnier 1978). Spelling reform is currently an extremely controversial subject in France, which can cause social and business problems when it is under discussion. Only minor corrections have recently been approved. Consequently, computational applications have to deal with this problem in the same way as do young students or foreigners learning the language.[1]

There are many classic examples of the problem in English. Among those often cited are the different phonetic realizations of the grapheme sequence *ough* as in *rough* [ʌf], *through* [uː], *bough* [aʊ], *thought* [ɔː], *dough* [əʊ], *cough* [ɒf], and *hiccough* [ʌp].[2]

A single grapheme or sequence of graphemes in various orthographic pairs may show considerable phonetic difference even within nearly identical environments: the *a* in *swan-swam* [swɒn–swæm] or the *wh* in *whorl-who* [wɜːl–huː]. When morpheme boundaries enter the equation, we find a similar situation, *uni* in *uniformed-uninformed*, *th* in *pothole-matthew*, *ph* in *flophouse-sphere*, and so on. Finally, lexical (external) borrowing results in exceptions to normal Germanic letter-to-sound rules such that many additional rules (or an exception table) are needed. For example, words like *cello* and *concerto* keep their Italian [tʃ] phoneme for the letter *c* followed by *e*. Similar problems are encountered with other partially assimilated loan-words from a variety of other languages, e.g., *entente* (one of the few forms: *entrée*, *entourage*, and a few oth-

[1] Many of the more sophisticated language pedagogical approaches now used to teach a conversational form of a language such as French or English will intentionally *not* expose the student to the orthographic system in the first lessons, since to do so causes interference with the phonetic form.

[2] Conversely, one phoneme may have a number of orthographic manifestations: the phone [iː] may be realized as *ey* (*key*), *ee* (*lee*), *ei* (*receive*), *ie* (*achieve*), *i* (*machine*), *e* (*she*), *ea* (*peace*), *eo* (*people*), *oe* (*amoeba*), *ay* (*quay*), *ae* (*aegis*), and *y* (*lovely*).

ers, in which orthographic *e* is realized phonetically as [ɒ]); and finally *bourgeois, hors d'oeuvres, Arkansas, Illinois,* and a variety of personal names that come from a disparate group of language families. When borrowed into English, these words are pronounced neither according to the rules of general English nor according to the rules of the source language.

In French, there are also different phonetic realizations of a single grapheme sequence: *x* is pronounced [ks] in *axiome,* [gz] in *exemple,* [s] in *soixante,* [z] in *sixième,* and [ ] (not pronounced) in *auxquels.*[3] There is sometimes a linking phoneme added between two words. A latent *e* is deleted in certain cases (elision).[4] For some phonemes (mostly vowels and semivowels), variations are sometimes acceptable and depend on the area, the speaker, or even the speech rate. Some words have a different pronunciation depending on the grammatical category of the word (when the form is a member of a pair of homographs).

Spelling and pronunciation are for both languages the result of history. With the Great Vowel Shift for English, pronunciation has changed, but not spelling (*face* for instance was pronounced with an [a:]). In French, *oi,* as in *roi* has been pronounced successively [ɔi], [ue], [we], [wa]. French has a lot of words that have kept their original pronunciation: Celtic words like *dolmen, menhir;* Latin words like *posteriori, in fine.* The spelling is, for many words, still dependent on their Latin or Greek origin.

3.2 Normalization

Normalization consists for both English and French in replacing graphemes by other graphemes or phonemes to expand numbers (12, 12.57, 12E+02), dates (12-03-97, 12-Mar-97), fractions (1/3, 1/4), telephone numbers, abbreviations (*kg, km*), acronyms (spelled, such as *U.F.O.* in English, *S.N.C.F.* in French, or pronounced as a word such as *NATO* in English, *OTAN* in French), and the like. For instance, 24 is replaced by *twenty-four* for English, and *vingt quatre* for French.

For French, the replacement is sometime context-dependent as in:

- 1 *enfant* '1 child' replaced by *un enfant* where there is a linking: a phoneme [n] added between *un* and *enfant,*
- or in 3 *enfants* '3 children' where a phoneme [z] is added between *trois* and *enfants.*
- 1 *fille* '1 girl' has to be replaced by *une fille* and not *un fille* due to the feminine gender of *fille.*

If the pronunciation of the word to normalize is not context-dependent, the word can be replaced by the equivalent phoneme string rather than another grapheme string.

3.3 Morphology

In some cases, depending on the language, words have to be decomposed into morphemes for letter-to-sound purposes. In English, *hothead, hothouse* are the concatenation of two morphemes, which keep their own pronunciation. The *th* in these words ([t+h] where + is a morpheme boundary) is different from the *th* in *this* [ð] or *thin* [θ]. Similarly, for French, forms like *tournesol, entresol, télésiège* are formed from two morphemes,

[3] As in English, one phoneme may have a number of orthographic manifestations: the phone [s] may be produced from *s, ss, c, c',* and so on as in *son, brosse, cet, c'est;* [f] is produced from *f* or *ph* as in *feu, flamme, amphibie, alpha.*

[4] Called an *e muet* 'mute *e*' in French linguistics.

each of which retains its pronunciation. Usually, in French, *s* between two vowels is pronounced [z], otherwise [s]. The *s* in *tournesol, entresol, télésiège, vraisemblable, contresens, antisocial* must be considered the beginning of a morpheme, and although it occurs between two vowels, is pronounced [s]. This morpheme decomposition is difficult and is sometimes based on a large dictionary of morphs. Some implementations have had as many as 12,000 for English (Allen et al., 1979). For English, and French, the number of words having this problem is relatively small, and can be dealt with by a dictionary or rules. In the English implementation, for example, many such morphemes can be incorporated directly into the letter-to-sound rule set itself. For certain other languages, such as German, where word compounding is quite common, morpheme decomposition algorithms tend to be much more complex.

3.4 Homographs

Homographs are pairs of words that are orthographically identical but phonetically different. In English, this difference is often simply a difference in stress depending on the grammatical category of the word: *permit* ([ˈpɜːmɪt] noun vs. [pəˈmɪt] verb), *baton* ([ˈbætən] noun vs. [bəˈtɔn] verb), *arithmetic* ([əˈrɪθmətɪk] noun vs. [ærɪθˈmetɪk] adjective) and so on. However, it can also be a difference of one or more segments: *deliberate* ([dɪˈlɪbərɪt] adjective vs. [dɪˈlɪbəreɪt] verb), *use* ([juːs] noun vs. [juːz] verb) differ in terms of only one segment. Further, it is not always possible to resolve the ambiguity from part of speech: in *I read books*, the pronunciation of *read* ([riːd] or [red]) is ambiguous. A less-frequently examined category, but one that is crucial to more natural speech synthesis, is what we will refer to as **functor homographs**. These are more subtle variations found in pairs such as *can*, which could be a verb ([kæn]) or a model auxiliary ([kN] or [kɨn] ~ [kæn]); *just*, which could be an adjective ([dʒʌst]) or an adverb ([dʒɨst]~[dʒʌst]), etc., where there is partial overlapping in careful speech. See Yarowsky (1994) on homograph disambiguation.

In French, the situation is similar. The same spelling can produce different phonemic forms: *fils* ([fis] 'son' vs. [fil] 'thread'); président ([prezidã] 'president' vs. [prezid] 'they preside'), etc. The pronunciation typically depends on the grammatical category of the word: *fier* ('proud' or 'to trust'), *est* ('is' or 'East'), *couvent* ('convent' or 'they brood'), *notions* ('we were noting' or 'the notions'), as ('an ace' or 'you have'), are all ambiguous in terms of their pronunciation. The word *six* can be pronounced [sis] (*j'en veux six*), [siz] (*six enfants*), [si] (*six filles*). First-order context can sometimes solve the problem (*nous notions* vs. *des notions; un as* vs. *tu as*), but, generally, a parsing of the entire sentence is required. The ambiguity is often between a conjugated verb and another grammatical category. The entire sentence can be ambiguous as in *"les fils sont jolis"* where *fils* is pronounced differently depending on the meaning (sons or threads).

3.5 Stress

For English, due to the interaction of stress and vowel reduction, knowing the stressed syllable is often crucial in determining the correct phoneme sequence (Halle and Keyser 1971). For instance, a word like *aggravation* has three tokens of the vowel grapheme a, but all are phonetically different. The vowel nucleus of the first syllable is [æ]; the stressed syllable *va* is manifested by [eɪ]; and vowel nucleus of the unstressed syllable *gra* (in this case) undergoes automatic vowel reduction and is realized as [ə]. The stress pattern for English is difficult to predict and has to be learned. Nevertheless, some basic rules exist. We have seen the verb/noun homographs in the previous section. In words of two syllables, the verb has stress on the second syllable, the noun on the first.

Adding one of a set of suffixes, S_i, to a word can keep the stress on the original syllable, such as for *-ful* in *beautiful*, *-less* in *defenceless*, *-able* in *changeable*, *-ness* in *shallowness*. A different set, S_j, modifies the stressed syllable. For instance, *-ity* moves the stress to the syllable preceding *-ity*, as in *fatal* [ˈfeɪtl]~*fatality* [fəˈtælɪtɪ]; *probable* [ˈprɒbəbl]~probability [prɒbəˈbɪlɪtɪ]; *-ation* moves the stress to the penultimate syllable of words ending in *-ation*: *aggravation* [ˌægrəˈveɪʃən].

Another difficulty for English concerns the stress in noun compounds such as *coffee cup*, *Thermos bottle*, *tape recorder*, etc., which exhibit a stress pattern of [1 2] or [1 0] due to the semantic unity of the two words in spite of the white space between them. Because of this white space, a program without special rules would assign primary stress to both words: [ˈkɒfɪ#ˈkʌp] instead of [ˈkɒfɪ#ˌkʌp] and [ˈθɜːməs#ˈbɒtl] instead of [ˈθɜːməs#ˌbɒtl]. This problem has been known some time (Liberman and Prince 1977).

Phonemes for French, on the other hand, are stress independent, since while lexical stress is nonfunctional except in a small number of ambiguous words and phrases, it plays a subordinate role to phrasal stress, which is invariant and phrase-final. Even for function words where the stress of the word is reduced in terms of duration and intonation, the phonemes stay the same.[5] The stress could be different in *belle fille* [ˈbɛl#fij] vs. *belle-fille* [bɛl#ˈfij], *grand père* [ˈgrã#pɛr] vs. *grand-père* [grã#ˈpɛr], stressing *belle* or *grand* if it is not a compound word. Even a word like *pomme de terre* could have two meanings: 'potato', or (the unlikely) 'apple made of earth'.

3.6 Morphophonemics

The conversion can also depend on the preceding and following words. For English, when *the* precedes a word beginning with a vowel it is pronounced differently from when it precedes a word beginning with a consonant: *the* [ðiː] *apple* versus *the* [ðə] *boy*. For English, except for grammatical category considerations, the conversion can be done simply by examining the phonetic status of the first segment of the following word.

In French, there are many cases where a phoneme is added at the beginning of a word depending on the last grapheme of the preceding word. This link between words is always between a word ending with a consonant (*d*, *t*, *n*, *s*, *z*), usually latent, and a word beginning with a vowel or an aspirated *h*. There are linkings that are mandatory, and others that are speaker or style dependent. These links occur between words in the same syntactic group: a personal pronoun and a verb (*nous avons* [nuzavɔ̃]), an article and a noun (*un enfant* [œ̃nãfã]), an adjective and a noun (*trois enfants* [trwazãfã], *grand homme* [grãtɔm]). In natural speech, these words are pronounced together, within the same breath group and without a pause between them. In these cases, a word-by-word conversion, or a dictionary lookup is insufficient to convert graphemes into their correct phonemic equivalents. Instead, the preceding word needs to be analyzed. For a few words, such as *nord-ouest* 'North-West', the link is done with the *r* of *nord*, not with the final consonant *d*. In words like *bon enfant*, *non aligné*, the link is done with *n*, but the preceding vowel of *bon* or *non* is also modified from [ɔ̃] to [ɔ].

3.7 Elision or Epenthesis of Schwa (Mute e)

Cases of elision are specific to French. For example, the grapheme *e* is sometimes realized phonetically as [ ] (the empty phoneme, meaning *e* is eliminated). This occurs in words ending in *e*, like *poule* 'hen', *genre* 'kind', except for monosyllabic forms,

[5] This is in marked contrast to vowel reduction in English, where most unstressed vowels become [ə].

such as *le* 'the' and *de* 'of'. If the last syllable is a consonant cluster ending in *e*, and the next word begins with a consonant, a short [ə] is heard as in *les chèvres de* [lɛ ʃɛvrədə] 'the goats of'; otherwise, more than two consonants would be in the same consonant cluster, and this presents articulatory difficulty in French and violates the constraints on syllable structure. Elision can be done in the first syllable of a word, but is considered familiar (vs. normal) style: *petit* [pti] 'small', *recommencer* [rkɔmãse] 'to begin again'.

In the middle of a word, elision is done for words such as *tellement* [tɛlmã] 'so much' but not for *justement* [ʒystəmã] 'precisely', which is additional support for the three-consonant cluster (CCC) constraint.[6] This elision sometimes does not occur, as in poetry reading, for example.

The rule does not provide for words like *batelier* [batəlje] 'boatman', or *bachelier* [baʃəlje] 'bachelor' where elision is not done. The semivowel [j] can be considered a consonant, and the three-consonant cluster constraint applies.

Sometimes, an [ə] phoneme is added between two words. For instance, in the newspaper name *Ouest-France* [wɛst#frãs], an epenthetic [ə] vowel is often inserted [wɛstə #frãs]. This happens between two words, in the context CC#C, and is again the result of the difficulty of pronouncing more than two consecutive consonants.

3.8 Segmental Phonology and Speech Rate

These rules are generic rules and sometimes may not apply in unusual cases, such as in very slow speech where each word is pronounced or in poetry (which often has its own set of rules different from normal speech). Thus far, in the area of speech synthesis, at least, not much has been done to modify segmental phonology according to speech rate.

In English, when the speech rate exceeds a certain threshold, in natural speech, pauses disappear and segmental durations become shortened. In the future, in text-to-speech systems, some segments and even syllables will disappear entirely and certain functors will be greatly attenuated. See Dirksen and Coleman (1994) for more on speech rate.

In French, in words containing a semivowel followed by a vowel, if the speech rate is slow enough (or sometimes in poetic contexts), a semivowel could be produced as a vowel: *lui* 'him' ([lɥi] vs. [lyi]), *nuage* 'cloud' ([nɥaʒ] vs. [nyaʒ]), *lier* 'to bind' ([lje] vs. [lie]). A common phrase such as *parce que* 'because', which is typically two syllables in normal speech ([parskə]) becomes three syllables in very slow or emphatic speech ([parsəkə]). In fast speech, the phrase *je te le dirai* [ʒətələdirɛ] 'I will tell you' is pronounced *je t'le dirai* [ʒətlədirɛ] or *j'tel dirai* [ʒtəldirɛ] eliding one or two [ə].

3.9 Proper Names

For proper names, the correspondence between written names and their pronunciation is even more difficult to specify due to their disparate origins. In English (whether British or American), there are many different ethnic groups represented in a telephone book or database of names. In a typical American telephone book, for example, are names that originate from hundreds of languages. In France, when a person is asked to provide a proper name, he or she is also often asked to spell it. For cities like *Caen* ([kã]), *Rennes* ([rɛn]), *Reims* ([rɛ̃s]), etc., the pronunciation differs substantially from the spelling. In proper names like *Lesage, Després, Bourgneuf, Montrouge, Lesventes,* it is important to recognize the morphemes *Le, Des, Bourg, Mont* to correctly transcribe. In

[6] In French, *la règle des 3 consonnes.*

both anglophone and francophone countries, these patterns of immigration have been sufficient to make this a serious problem for any automatic phoneticization algorithm.

The rules for proper names can generally be derived from the rules for words. Nevertheless, a large superset of rules has to be added to obtain very high accuracy since the phonotactics change from language to language. Moreover, to compound the problem, the pronunciation of proper names outside of the foreign speech community is often different from their original pronunciation. For example, in the US, *e* ending Italian names (pronounced [e] in Italian) is typically pronounced [iː] or even [ ] (not pronounced). The proper name *Falcone* is pronounced in anglophone countries as either [fælkɒniː] or even [fælkɒn], *Bach* as either [bax] or [bak]. In French, we observe a similar situation where the name *Smith* is pronounced [smis] and *Thatcher* as [satʃər] as French does not have a [θ] phoneme.

There have been successful attempts to automatically detect the ethnic group of a proper name for use in anglophone countries like the United States, and to apply a different set of rules depending on that group (Church 1985, Vitale 1991). Trigram frequencies are computed from a large set of proper names whose ethnic group is known, and used to classify a new proper name in terms of some language, language group, or language family (the linguistic etymology of the name). Depending on that classification, different subsets of language-specific rules can be activated.

4. Expert Systems

Expert systems are used to facilitate the transfer of the knowledge of a specific domain from an expert to a computer. They traditionally distinguish between the system, which is as independent as possible from the application, and the expert rules, which are application dependent. The system requires a computer specialist, the rules require an expert in the domain to be processed, in this case, a linguist. Everybody is an "expert" in reading his or her own language, and the average educated individual does not hesitate in front of a word like *monsieur* or *second* in French, or *hiccough* or *Edinburgh* in English, even though the pronunciation may be quite different from the spelling. In any case, we apply, albeit unconsciously, rules to read text aloud.

Considering the complexity of the problems presented above, it was quickly understood that letter-to-sound rules had to be treated like an expert system with a rule set developed by an expert (a linguist) and an interpreter to interpret the rules. This is a pragmatic approach based on failures of systems that use hard-coded rules that the linguist would be forced to program or the programmer would be forced to articulate.

5. The English Rule Set

5.1 The Rule Formalism for English

Essentially, a letter-to-sound rule can be viewed as similar to a phonological rule in classical phonology except that it converts a grapheme string to a phoneme string. These rules may be context-sensitive or context-free. A lexical entry in a dictionary (without syntactic and semantic information) is, in essence, a context-free letter-to-sound rule.

An efficient rule set had to be developed. This rule set had to be:

- rigorous (have a minimum of ordering constraints, such that new rules could be added at random with a minimum of liability);
- complete, with a large number of rules covering large sequences

including morphs both free and bound;

- optimally parsed in order to make use of morphological information relevant to allophonic variation as well as to stress.

Using these criteria as a working basis, we developed a set of highly accurate letter-to-sound rules.

In English, the scan is done right to left to strip the suffixes of a word in sequence as shown in Example 4 below. The input is a string of graphemes, the output a string of phonemes (and occasionally the allophones themselves). There is only one scan. The rules themselves are stated in terms a linguist would be familiar with such as the following:

$$X \rightarrow [y]; \quad \text{(context-free)}$$

or

$$X \rightarrow [y] \ /W - Z; \quad \text{(context-sensitive)}$$

where X, W, and Z are grapheme sequences and [y] is a phoneme (or phone) sequence.

A two-tiered architecture (compiler and interpreter) has been designed to easily define and modify the rule set in our implementation of grapheme-to-phoneme rules.

The rule compiler transforms the external form of the rules into an internal form that can be easily used by the rule interpreter. The grapheme pattern is encoded as a simple text string. The left and right context patterns are encoded as strings of operators and parameters for a pattern-matching procedure, and the replacement phoneme string is encoded using the system's internal phoneme codes.[7] The grapheme pattern and the left context pattern are reversed by the rule compiler (that is, stored in right-to-left order) so that they are stored in the direction that they are actually used. The rule compiler does not perform any sophisticated checking of the rules; it does not check that the rule set is complete, nor does it check that long rules are always presented before short rules.

The rule interpreter begins processing a word by setting its current position to the rightmost grapheme. It then searches linearly through the rules, in the order they were written, until it finds a rule that matches at that current position. A rule matches if the grapheme string matches, the left context pattern matches (if present), and the right context string matches (if present). The grapheme string is matched using a simple right-to-left text compare, and the context strings are matched by a recursive procedure that interprets the pattern string built by the rule compiler. The phonemes for the rule are then placed in the output, the current position is advanced over the matched graphemes, and the process is repeated until a rule consumes the leftmost grapheme. Since the rule set contains an unconstrained rule for each grapheme, the matcher will always find a rule, and will always make progress. Matched graphemes are not deleted; the word is left intact, since "consumed" graphemes could be part of the right-hand context of some future rule. The phoneme string generated by the letter-to-sound rule interpreter is represented as a double linked list. This representation was chosen

[7] The right-to-left match has already been described. It should be pointed out that the use of "text" in "text string" was not ASCII, but an encoded alphabet in which some grapheme pairs, like *qu*, *gu* and certain others were encoded as single letters, because doing so made it unnecessary to have a large number of (unnecessary) blocking rules in the rules for the grapheme *u*.

because subsequent processing (syllable marking, stress analysis, and final allophone adjustment) needs to be able to scan the phoneme string in both directions, and needs to be able to add and delete phonemes at arbitrary places.[8]

It would be, of course, possible to use more elaborate string-matching techniques to increase the speed of rule selection, but this was not done in our system because letter-to-sound processing never uses a significant fraction of the total processing time.

5.2 Examples of Rules for English

Example 1

The following is an example of a set of two letter-to-sound rules for the letter c in English. The first is context-sensitive and the second context-free:

c → [k] / – {a, o};
c → [s];

This set reads as follows: The grapheme *c* is realized phonemically as [k] if occurring immediately before the grapheme *a* or *o* as in *cab, cake, decaf*; it is realized as [s] elsewhere: *cease, cigar*.[9]

Example 2

Such rules, of course, handle only those forms that constitute the set of assimilated or partially assimilated loanwords. In the case of the English rules above, words such as *call, cell, cilia, cool* would be handled, as well as *cure, cute*, (assuming that palatalization issues are handled by another rule). It does not account for words such as *cello* [ˈtʃeləʊ] or *concerto* [kənˈtʃɛətəʊ], because these are unassimilated borrowings that still show the original Italian palatalization rule of:

c → [tʃ] / – {i, e};

When we have a rule that handles *n* words, where *n* is between 1 and some small number, say fewer than 7, we generally put these forms in a dictionary instead of using up computation to process such a small number of words. Similarly, even if a rule to convert *e* to [ɒ] (to handle words such as *entrée* [ˈɒntreI], *entente*, or *entourage*) could be written, it would be much easier and more efficient to put the words it affects in a dictionary, because there are so few of them.

Example 3

ation → [1][eI] = [0][ʃ][ə][n] / – +;

indicates that the string *ation* at the end of the word (morpheme boundary) is replaced by the phoneme string:

- [eIʃən]
- plus a mark [1] of primary stress for [eI],

[8] Syllabification, stress, and final allophone adjustment are done after the first output of a phoneme string.

[9] We used square brackets for the segmental output of the rules. We have adopted this convention because the output could be either phonemic or phonetic. That is, if allophonic rules can be done in one pass here, we include them along with rules that output phonemes.

- a syllable boundary =,
- and a mark [0] of unstressed syllable for [ʃən], as, for instance, in *aggravation.*

Example 4
This example shows the decomposition of words into their constituents morphs in such a way as to "undo" the mutations caused by suffixes. In some cases, the input string is modified to add a morpheme boundary, or replace the suffix. With the word *finishing,* a context-sensitive rule in *ing* would, for instance, produce the phonemes for *ing* plus a mark [0] indicating that the syllable is unstressed, add a morpheme boundary mark (+) in the input string, which is then *finish+ing,* and continue the conversion from right to left starting on *h* of *finish.*

ing > + → [0][ɪ][ŋ] / – +;

With the word *riding,* a context-sensitive rule in *ing* would produce the phonemes for *ing* plus a mark indicating that the syllable is unstressed, replace the suffix *ing* by *e+* in the input string, which is then *ride+,* and continue the conversion from right to left starting on *e.*

ing > e+ → ... / – +;

With the word *relationship,* the rule decomposes the word into *relation* + *ship*:

ship > + → ... / – +;

scandalousness is decomposed into *scandal* + *ous* + *ness* by the following rules:

ness > + → ... / – +;
ous > + → ... / – +;

This suffix stripping is the main reason for a right-to-left scan for English (Allen 1976).

Example 5
o → [ɒ|əʊ] / micr –;

means *o* will be translated as [ɒ] if the syllable is stressed (*micrometer*), and as [əʊ] otherwise (*microgram*). (See Section 5.7 for stress assignment)

5.3 Normalization for English

Text normalization, i.e., replacing numbers, abbreviations and acronyms, by their full text equivalents is done in a preprocessing section. In English, the choice between expansion to the full graphemic equivalence or expansion to a full phonetic equivalence was made in favor of the latter.

English contains a separate preprocessing section for numbers (24 in *twenty-four*), acronyms (IBM, FBI), or abbreviations (*Pr.* for *Professor,* $ for *dollar(s)*). Some of these examples can become quite complex: $50 is retranscribed (or phoneticized) as *fifty dollars;* $50.60 as *fifty dollars and sixty cents;* $50 *million* as *fifty million dollars;* $50.2 *million* as *fifty point two million dollars;* and so on.

Some characters may or may not be pronounced depending on the application (punctuation spelling for instance): 1 *kg* is a singular *one kilogram* but 5 *kg* is plural *five kilograms*. Similarly, *Dr.* may be *doctor* or *drive* and *St.* may be *street* or *saint*, depending upon the context. We disambiguate and expand all such abbreviations in a separate module that by-passes letter-to-sound. There are switches that can be set, for example to turn all punctuation off, to turn it all on, or to normal pronunciation, where very few punctuation marks need to be pronounced. Any of these approaches works. The advantage of a separate text preprocessing module is that it does not clutter up the letter-to-sound rules. It can be optional, removed or replaced as necessary depending on the application.

5.4 Homographs for English

In English, homographs represent a common problem that cannot be solved entirely by letter-to-sound rules. There has traditionally been an avoidance of the problem by defaulting to one member of the pair based on blind form class selection (default to the noun), which, of course, is less than adequate. For example, in grapheme strings such as *refuse* and *produce*, the default to noun would be to [ˈrefjuːs] and [ˈprɒdjuːs], which, in unrestricted text, are less frequent than the verb forms.

Later solutions in our system involved a default to the member with the higher frequency of occurrence. For example, using the same words, the default would be to [rɪˈfjuːz] and [prəˈdjuːs] rather than to [ˈrefjuːs] and [ˈprɒdjuːs].

5.5 Morphophonemics

There are several rules for phonemic tuning, especially to account for morphonemic alternations, which are extremely important. For example, there are a number of essential morphonemic rules in English that perform various tasks, such as plural and past tense formation. These rules are very well known among linguists and need to be formalized in the same way as were the grapheme-to-phoneme rules. This time, however, we are always going from a morphophonemic to a phonetic realization as in:

$$\{x\} \quad \rightarrow \quad [y] \quad / \quad [w] \quad - \quad [z];$$

where {x} is an archiphoneme or abstract morphophoneme, [y] is some phonetic sequence, and [w] and [z] are some environment *E*, where *E* is either phonemic or phonetic. For example, the following are two well-known rules that implement the phonetic realizations for [plural] and [past]:[10]

After conversion, we have for *roses*, the following phoneme string: [r][əʊ][z]+[z]

$$\{z\} \quad \rightarrow \quad [\text{ɨ}][z] \quad / \quad [+\text{Cons}, +\text{Sib}] + - \#;$$

applies for the second [z], which is preceded by + (morpheme boundary), and by a sibilant consonant ([z]).

After conversion, we have for *cats*, the following phoneme string: [k][æ][t]+[z]

$$\{z\} \quad \rightarrow \quad [s] \quad / \quad [+\text{Cons}, -\text{Voice}] + - \#;$$

[10] {z} and {d} are abstract base forms that are replaced by appropriate phones.

applies for [z], which is preceded by + and an unvoiced consonant ([t]).

After conversion, we have for *spotted*, the following phoneme string: [s][p][ɒ][t][t]+[d]

{d} → [ɨ][d] / {[t], [d]} + − #;

applies for [d], which is preceded by + and by [t].

After conversion, we have for *walked*, the following phoneme string: [w][ɔː][k]+[d]

{d} → [t] / [+Cons, −Voice] + − #;

applies for [d], which is preceded by + and by an unvoiced consonant.

5.6 Syllabification

A phone scanning, from right to left, marks the positions of the syllables according to consonant clusters, vowels, and morph boundaries.

For instance, *scandalousness*, which has been processed by the previous steps as:

[s][k][æ][n][d][ə][l] + [ə][s] + [n][ɨ][s]

is decomposed into syllables as follows:

[s][k][æ][n] − [d][ə][l] + [ə][s] + [n][ɨ][s]

chevron would result in:

[ʃ][e][v][r][ə][n]

and would be decomposed as:

[ʃ][e][v] − [r][ə][n]

Although there are several different theories of syllabification, any standard linguistics book will have a reference to these valid clusters and an accurate definition of the syllable for a language *L* (Clemens and Keyser 1983). It is beyond the scope of this paper to discuss the merits of one theory of the English syllable over another. Whatever theory is chosen, syllabification should serve as an accurate input into the module that handles stress.[11]

5.7 Stress

The letter-to-sound rule set described above sets lexical stress in a wide variety of cases, especially where the word is monosyllabic or the suffixal information is sufficient to place primary or secondary stress.

These routines contain special rules, which contain a number of different options:

(a) assign primary stress,

(b) place primary stress *n* syllables to the left or right,

(c) place secondary stress,

[11] Syllabification can be applied in the user interface as a useful addition to spell mode (i.e., "say letter"), and word mode ("speak word by word"). Such an interface can then be used in applications ranging from language pedagogy to the teaching of reading to individuals with learning disabilities.

(d) place secondary stress *n* syllables to the left or right,

(e) assign [-stress] (not stressed) to a syllable,

(f) refuse stress.

Example of letter-to-sound (morph) rules that would have already assigned primary stress:

```
ation   →   [1][eɪ] = [0][ʃ][ɨ][n]   /   – +;
```

- [eɪ] has primary stressed (marked [1]) as in *transformation;*
- [ʃ][ɨ][n] is unstressed (marked [0])
- = is a syllable boundary.

Example of letter-to-sound rule that would have assigned primary stress one syllable on the left:

```
graphy  >+  →  [S1left] [g][r][ə]=[0][f][ɪ]  /  - + ;
geo         →  [dʒ][ɪ][ɒ|ə]                  /  - + ;
```

The primary stress is one syllable [S1left] to the left of *graphy,* so [ɒ|ə] is stressed and the phoneme is [ˈɒ]

There are certain affixes in English that refuse to be assigned stress. For example the prefix *in-* normally does not take [1 stress] except under contrastive stress, e.g., I said *in*clude, not *pre*clude. A word is scanned left to right and on syllables that fall under the category of stress-refusers, a flag is set. It is possible that more that one contiguous syllable will refuse to take stress.

Generic stress rules in this module assign primary stress if and only if [1 stress] has not yet been assigned. In this block, the word is scanned left to right, the number of syllables is counted, and pointers are stored in syllable-initial position in an array *A*. The number of syllables in the root form is counted and the syllable that forces the primary stress is marked as [1 stress].

Primary stress ([1 stress]) is a requisite for all words except certain words already marked otherwise in the dictionary and noun compounds. If at the end of these rules, [1 stress] still has not been placed on a word, a set of generic rules applies. First the number of syllables in the root is noted and a flag is set on that syllable with the most likely default for the placement of [1 stress].

Examples of default rules are as follows where $ is a syllable:

```
$  →      $
      [1 stress]
```

For instance: *smart*

```
$$  →       $          $
        [1 stress] [0 stress]
```

For instance: *baby* (stressed on the first syllable *ba*)

5.8 Allophonics

The allophonic pass performs some allophonic rules well known to those familiar with phonemic variation.

The phoneme string is scanned left to right, performing such tasks as vowel reductions. This is done in a prepass, to ensure that each [ə] or [ɨ] (reduced) vowel is accurately adjusted before the main body of the allophonic rules are run.

The following are examples (a small subset) of (ordered) rules of the final allophonic pass:

[n] → [ŋ] / – {[k], [g]};

pancake, previously transcribed [ˈpænkeɪk], becomes [ˈpæŋkeɪk].

[s][s] → [ʃ] / – [uː]+;

issue, previously transcribed [ˈɪssuː], becomes [ˈɪʃuː].

Finally, one member of geminate pairs is deleted. There are some special pairs like [l] and [ʟ] (syllabic [l]) that get deleted even if there is a morpheme boundary between them. Nevertheless, often these rules are blocked if they cross a morpheme boundary.

[d][d] → [d];

This rule applies for *adder*, which is *add+er* but does not apply for *midday*, which is *mid+day*

6. The French Rule Set

For French, an ad hoc programming language has been designed to easily define and modify the rule set. Text normalization, i.e., replacing numbers, abbreviations, and acronyms, by their full text equivalents, and grapheme-to-phoneme transcription can be achieved using this formalism.

6.1 The Rule Formalism for French

6.1.1 Input and Output Characters. The external codes of the units to be processed must be declared, i.e., the grapheme codes (upper and lower case letters, numbers, punctuation, diacritics) and the phoneme codes. These codes may be composed of one or more characters. In this way, users can define their own code, and the formalism can be used for different languages. These basic input and output units, or *elements*, are expressed as e*i*, where *i* is some number.

6.1.2 Strings and Classes. A string consists of the concatenation of the predeclared external characters: e1e2e3 is a string. A class is a set of strings having a common property. 'C1' and 'C2' are classes.

'C1' : e1, e2, e3/

'C2' : e1e2, e5e1, e2e3/

6.1.3 Blocks of Rules. The set of rules consists of one or several blocks of rules. Each block describes a process, taking the input text, processing it, and replacing it by the result of the processing. The different blocks can be activated sequentially, or directly (execute block 5 for example).

```
begin  {Block i}
       rule 1
       ......
       rule n
end
```

6.1.4 Rules. The syntax of a rule is:

⟨number⟩ : ⟨ls⟩ → ⟨rs⟩/⟨lc⟩ – ⟨rc⟩;

where *number* is the rule label; *ls* (left string) is the string to be replaced; *rs* (right string) is the string replacing *ls*; *lc* (left context) represents the strings to be found on the left side of *ls*; *rc* (right context) represents the strings to be found on the right of *ls*. *lc* and *rc* are formed with operands (characters, strings, classes) and operators (concatenation, logical or, negation).

6.1.5 Using One or Two Buffers. If the contexts match, the *rs* string replaces the *ls* string. This process can be achieved using either one or two buffers. With one buffer, the *rs* string replaces the *ls* string, so the left context of a rule must be written according to the rules previously used. With two buffers, the writing of the left context of a rule is easier because the input string is only modified at the end of the block of rules. In effect, three contexts are usable: the left context and right context in the input buffer, and the left context in the output buffer. This left output context is written between angled brackets.

6.1.6 Formal Examples of Rules.

E = A, B, C, D, E, F, G, H/; input and output characters

'C1', 'C2', 'C3' classes formed with strings of E.

'C1' : AB, CD/

'C2' : CFG, DE, AAH/

'C3' : CCC, CBA/

```
1 : EF   → H      /E.'C2'–;
```

EF is replaced by H if, on the left side of EF, an element of the class 'C2' is found preceded by another E.

```
2 : AB, CD → FC      / 'C1'.H - 'C2'.G, 'C3';
```

The string AB or CD is replaced by FC, in the following contexts:

- on the left of AB or CD, a H preceded by an element of the class 'C1',

- on the right of AB or CD,

  either an element of 'C2' followed by a G,
  or an element of 'C3'.

```
3 : HH   → A    /Non(C, E, H)−;
```

HH is replaced by A if the left context is not a C, an E, or an H.

```
4 : CA   → GE   /⟨G.'C3'⟩H−;
```

CA is replaced by GE if:

- the left context of the output buffer (between angle brackets) is an element of 'C3' preceded by G,
- and the left context of the input buffer is H.

6.1.7 Interpreting the Rules. The rule having the longest match between the set of all the *ls* strings of the block, and the string beginning with the next character to be processed in the input text, is searched first. If both contexts are true, the rule applies, otherwise another rule is searched for, first any other rule with the same *ls*, and then in decreasing length of *ls* matches.

Let us consider the following rules:

```
begin
        50 : AB       → . . . ;
        51 : A        → . . . ;
        52 : ABC      → . . . ;
        53 : AB       → . . . ;
        54 : BC       → . . . ;
        55 : ABCG     → . . . ;
end
```

and the input string, "ABCE" to be processed.

The longest match between the left string (*ls*) of the rules in the block, and the input string to be processed is searched. In this case the longest match is "ABC". So, rule 52 is tested. If the contexts are true, the rule is applied, and the next character to process is "E" in the input string. If the context is false, the rules are tested in decreasing order of the longest match. Rules with "AB" as *ls* are tested in the order in which they are written (50, 53). Then if no rule has yet been applied, rule 51 is tested. If no rule is true, the first character *A* to process is copied into the output buffer, and the procedure starts again with the next character *B*. The order in which rules are tried is: 52, 50, 53, 51. The order in which the rules are written is significant only for those having the same *ls*.

Using the formalism of the expert system, the expert is in charge of defining a set of rules to simulate his or her expertise.

6.2 Examples of Rules for French

As this paper is in English dealing with the French language, and in the event that the reader might be not familiar with the idiosyncrasies of French, only a few examples will be given to explain the mechanism of the letter-to-sound rules for French.

Example 6

o	is pronounced [o]	in *moto, loto, solo.*
oi	is pronounced [w][a]	in *moi, pois, fois.*
on	is pronounced [ɔ̃]	in *bon,* but not in *abandonner, bonheur,* or *bonne,* where the rule for *o* applies.
oin	is pronounced [w][ɛ̃]	in *loin, poing* but not in *avoine* where the rule for *oi* applies.

The rules could be written as shown below.

'CexceptN': B, C, . . . grapheme consonants except N /

5 : oin → [w][ɛ̃] / – 'CexceptN', _;

oin gives [w][ɛ̃] if *oin* is followed by an element of 'CexceptN' or a space. Otherwise,

6 : oi → [w][a]; context-free rule

7 : on → [ɔ̃] / – 'CexceptN', _;

on gives the phoneme [ɔ̃] if *on* is followed by a consonant except N, or a space. Otherwise,

8 : o → [o];

Independently of the order of the rules, the rules having the longest match will be first tested. Here, the order of the rules is irrelevant.

Example 7
er at the end of words is pronounced [e] as in *chanter, danser* but [ɛr] in *super, joker, fer,* or *hier.*

The rules could be formulated as:

'Wer': sup, jok, f, hi/

9 : er → [ɛ][r] /_.'Wer' – _;

er is pronounced [ɛ][r] if *er* if preceded by an element of 'Wer' (words ending in *er*) preceded by a space and followed by a space.

10 : er → [e];

Otherwise *er* is rewritten as [e].

Example 8
The *ai* string in French words like *bienfaisant, contrefaisait, faisait, faisan, satisfaisant,* etc., is pronounced [ə] but not in *faisceau, chauffais* where the corresponding phoneme is an [ɛ].

The rule can be written as:

'Vowels': a, e, i, o, u, y/

11 : fais → [f][ə][z] / – 'Vowels';

fais is pronounced [fəz] if *fais* is followed by an element of the class 'Vowels'.

Example 9

In order to eliminate geminates, one possibility is to analyze the last character sent to the output buffer.

12 : *b* → /⟨[*b*]⟩–;

b is eliminated if the left context in the output buffer is already a phoneme [b]. (See Section 6.1.5 on using one or two buffers.)

6.3 Normalization for French: from Graphemes to Graphemes

The first step, done by a block of rules, is to normalize the text, replacing numbers, abbreviations, and acronyms by their full text equivalents. Both input and output are graphemes. Normalization is handled in the letter-to-sound rule set and in a preprocessing module. By rules, the contexts indicate if the replacement is required.

Numbers. 123 is rewritten as *cent vingt trois* by a set of rules checking the left and right context for each digit.

'Digit' : 0,1,2,3,...,9/ is the class for digits

13 : 1 → cent_ / – 'Digit'.'Digit'._;
14 : 2 → vingt_ / – 'Digit'._;
15 : 3 → trois / – _;

1 is rewritten *cent_* if followed (right context) by two digits and a space, etc.

Abbreviations. kg for *kilo, Dr* for *Docteur, Pr* for *Professeur, bd* for *Boulevard*, etc.

16 : kg → kilos /_,'Digit' – _;

kg is replaced by *kilos* in 5*kg* or *trois kg*.

Acronyms. Similar rules are used to spell acronyms (*I.B.M.* gives [IbeƐm]):

17 : B. → bé /_,.–;

B followed by a point is replaced by *bé* (spelled) if *B* is preceded by another point or a space. In *I.B.M.*, or *vitamine B., B* is spelled correctly.

Preprocessing procedures are also used in cases like $50, which gives: *cinquante dollars* and where you have to permute $ and 50.

6.4 Morphology

The problem mentioned in Section 3.3 is solved most of the time using rules for French. For words like those in Section 3.3 (*homosexuel, hétérosexuel, télésiège, entresol, tournesol*), a class is defined with the prefixes ending with a vowel.

For instance,

Prefix: homo, hétéro, télé, entre, tourne /

18 : s → [s] / 'Prefix' - 'V' ;

s is pronounced [s] if preceded by an element of 'Prefix', and followed by an element of 'V' (a vowel) as in télésiège.

19 : s → [z];

as in *base, bise, anglaise, opposition*

6.5 Homograph Problem

A limited parsing has been done using the same formalism as letter-to-sound. A dictionary lookup gives one or several grammatical categories for the most common words. By examining the left and right words, it is possible in most of the cases to get an idea of the grammatical categories of the unmarked words or to reduce (to one if possible) the set of potential grammatical categories for each word of a sentence. The same formalism allows the processing of grammatical categories (verb, adverb, preposition, etc.) instead of characters for transcription. A class is a set of grammatical categories (Divay 1984, 1985).

If the grammatical category is known (where V = Verb), it can be used in the rules:

20 : ent(V) → / – _;

ent is eliminated at the end of a word (right context is a space) if the word is a verb (*ils chantent*).

6.6 Linking

In some cases, a new phoneme is added between two words of a same-breath group. For instance, a [z] phoneme is added between the two words of *les enfants*. The second word has to begin with a vowel or aspirated *h*, and the first one to end with *n, s, d, t, x*, or *z*. It depends also on the grammatical category of both words. This problem also is solved by rules, such as the following:

21 : _ → _[z] /_les, _des, _ses, _nous – 'Vowels';

The space between two words is replaced by a space and a phoneme [z] if the space is preceded by *les* or *des*, etc., and followed by a vowel, as in *les enfants*.

If the left context is very large, a new class can be created, and used as left context.

6.7 Elision: From Phonemes to Phonemes

Some rules, mostly rules dealing with mute *e* and semivowels, can be more easily expressed on the phonemes strings. This is a new block of rules run after the grapheme-phoneme conversion.

The following is an example of elision with mute *e*:

```
'VP' : [a], [i], . . . /       the vowel phonemes
'CP' : [b], [d], . . . /       the consonant phonemes
22 : [ə]   →      /'VP'      – 'CP'.'VP';
```

Mute *e* is eliminated before a vowel phoneme and after a consonant phoneme followed by a vowel phoneme as in *emploiera* [ãplwaəra], which becomes [ãplwara].

Elision often occurs at the end of words (*petite*), or in the middle of words (*emploiera, tellement*). It can be done in the first syllable (*pesanteur, retard, teneur*) except if there are two consonants as in *premier*. It is never done if suppressing [ə] would result in three or more consecutive consonants.

7. Testing

No standardized tests exist for evaluating letter-to-sound systems, although some researchers are beginning to look at the problem in order to determine whether one approach has merit over another (Golding and Rosenbloom 1993). For example, the Oregon Graduate Institute is currently investigating letter-to-sound rules done in more traditional ways and comparing them to neural network learning.

Tests can be done:

1. with or without an exception dictionary lookup running before the rules,
2. on text extracted from papers, books, magazines. In that case, the same words is counted as many times as it appears in the text. This is especially true for linking words (*one, a, the, is,* etc.), which are then counted many times. A more systematic test can be carried out using an electronic dictionary, having for each entry (grapheme string) the corresponding phoneme string. In that case, every word is tested and counted one time, even though its occurrence frequency might be very low,
3. in terms of percentage of phonemes or of words correctly transcribed. Percentage of phonemes is obviously higher than percentage of words.

7.1 English Analysis

The rule set for English consists of about 1,500 rules containing morphs as well as nonsemantic grapheme strings. An exception dictionary has been defined for words not correctly translated by these rules. These consist mostly of functors, abbreviations, homographs, and unassimilated loanwords such as *adobe, bayou, cello, coyote,* and the like. In addition, the lexical entry need not contain phonetics, especially if the entry in question is adequately handled by rule. It may, however, still be used to convey both syntactic and semantic information that would then serve as input to a parser for more accurate prosodic rules.

In this study, we took two different corpora: (1) a 1,676-word corpus originally used by Bill Huggins (BB&N) and eventually by Dennis Klatt (MIT). This corpus was chosen because it consists of complex polysyllabic forms; (2) a sample taken from the Brown corpus (19,837 words), which we felt to be sizable enough and representative enough to use to examine letter-to-sound accuracy.

On the Huggins corpus, without the use of the exceptions dictionary, our rule set scored 94.9% of words. The 5.1% errors consisted mainly of incorrect morphological analysis and consequent inaccuracies in lexical stress placement.

On the Brown corpus, we had a large number of dictionary hits, which was not unexpected since the corpus contains many high-frequency forms. Out of a total word count of 19,837 words, the dictionary hit count was 7,337 (36.99%); the rules matched 5,432 words (27.38%) for a total word match of 12,769 or 64.37%. Of the words missed, 3,905 (19.69%) missed by only one segmental phoneme or phone and 3,636 (18.33%) had incorrect stress placement. We consider incorrect stress placement to be a more serious error than one incorrect segmental phoneme.

The latest version is used in different products, from text-to-speech synthesizers both hardware and software, assistive devices, and games, and will soon be used in proper name retrieval, both on computer systems and over the telephone. Using the same formalism, a different set of rules has been defined for proper names found in a typical telephone book in the US and could be extended to other languages.

7.2 French Analysis

The set of rules for French consists of about 600 rules and 100 classes. Some of these classes contain 100 or more elements. The French letter-to-sound rule set was tested on the 55,000 unique word *Le Petit Robert* dictionary, and the 100,000 word *Le Grand Robert de la Langue Française* dictionary. An exception dictionary is automatically defined for words not correctly translated by these 600 rules.

The execution of the set of rules on the 55,000 unique word dictionary gives 4.4% of words whose pronunciation is different from the dictionary *Le Petit Robert* but is acceptable from the authors' point of view. These differences are only due to a mismatch between open or closed phonemes for phonemes [a], [e] and [o].

The distinction between the open [a] and closed [ɑ] has almost disappeared in France in favor of the open [a]. The proposed pronunciation varies even from one dictionary to another. Words like *accablant, phase, câble, vase,* and *trois* have different pronunciations depending on the dictionary used. Sometimes, both are mentioned. They are even differences between *Le Petit Robert* and *Le Grand Robert* dictionaries.

Both open [ɔ] and closed [o] are also acceptable in many words e.g., *automobile, aérodrome, augmenter, autonome, austral, ozone.* Nevertheless for some words, the distinction has to be made (*bol* [bɔl] vs. *rose* [roz]). The closed phoneme is used for instance before a phoneme [s]: *pose, chose, oser* or at the end of a word: *abricot, escargot.*

The closed [e] and open [ɛ] are also very much interchangeable in many words (*les, baisser, adolescent, essai, agressif, blessant, intéressant, aigri, biennal, accession*).

Of the 55,000 words, 2.8% are incorrectly processed (1,500 out of 55,000), and have to be added to an exception dictionary. Some words have several acceptable pronunciations (*août* [aut, ut], *ananas* [anana, ananas], *dompter* [dɔ̃mpte, dɔ̃te]), *bat, babil, blet, chenil, exact, but, as,* but only one is stored in the electronic dictionary. Some problems result also from a different but acceptable elision of mute *e*, as in *chemin de fer, briqueterie, petit-neveu, amenuiser, point de vue, porte-bébé, redevenir.* But, most of the errors come from foreign words such as: *accelerando, adagio, allegro, artefact, posteriori, mea culpa, beluga, placebo, torero, baby, girl, shirt, blue-jeans, base-ball, steward, business, building, copyright, bonsaï.*

The number of applications of each of the 600 rules has been calculated on the 55,000 words to give an indication of its weight.

This program is currently in use in different laboratories in France, Canada (O'Shaugnessy et al. 1981) and the United States (DEC) as the first level in speech synthesis for French. It has been used by various companies producing electronic board speech synthesizers for French.

This transcription program has also been used to create a phonetic index and retrieve a word without knowing how to write it. The word is converted to phonetics and searched for in the phonetic dictionary index (used in both CD-ROM dictionaries *Le Grand Robert* and *Le Petit Robert*) (Rey et al. 1989). For information retrieval, open and closed phonemes are always considered identical. The same mechanism (using phonetics) is used to retrieve a proper name (without knowing how to spell it) through the 30,000 proper names of the phone book of the city of Dakar (Senegal). The system is also used in the Taurus multimedia database software (from DCI: Data Concept Informatique) to create an index on one field of a structure defined by the user of the database, and to retrieve the corresponding information even if it is misspelled. Other similar uses are under investigation for the pronunciation of names from on-line telephone books in particular and telecommunications applications in general (Alcatel TITN Answare).

8. Final Remarks

It is beyond the scope of this paper to discuss letter-to-sound procedures in languages other than English and French. However, the disparate nature of different languages argues for a brief mention of our experience in developing letter-to-sound rule sets in other languages.[12]

8.1 Simple Systems

In certain languages, as diverse as Spanish and Swahili, letter-to-sound rule sets are extremely easy to produce, due to the extremely close fit between orthography and its phonemic/phonetic equivalent. First, there are many languages that developed a writing system only recently. Swahili, for example, was written in Arabic script until 1850 when Krapf, a German missionary, introduced the Roman alphabet to the Bantu-speaking peoples of the East African coast. Consequently, in the time span of less that 150 years, the phonological and phonetic systems of the language have not had time to change to any significant extent. Secondly, many languages have undergone some spelling reform. Czech, for example, underwent spelling reform fairly recently and the orthographic system was brought into line with the phonological and phonetic system. Third, there are some languages in which the orthography had a close fit with the phonemic system. Spanish, for example, is a simple system in that there is an almost iconic relationship between graphemes and their phonemic equivalent. In fact, even lexical stress is marked in many forms and, where it is not, it is almost always predictable.

8.2 Mid-Level Systems

Many languages are somewhat more complex and fit into a second category of languages of mid-level difficulty. German, for example, has a large morphological system yet it is surprisingly simple in terms of letter-to-sound rules. If one lists a large number of common morphemes, it becomes a simple task to state an accurate set of letter-to-

[12] All languages of the world are of an equal degree of complexity. Primitive languages are a myth perpetrated by early anthropologists, missionaries, and adventurers. However, when we compare different subsytems of any two languages, it quickly becomes clear that subsystems are vastly different in complexity. This is true of the phonological, phonetic, morphological, syntactic, semantic and letter-to-sound subsystems of two different languages; some are an order of magnitude more complex than others.

sound rules. Many languages with a high synthetic index (Greenberg 1990) fall into this category.[13]

8.3 Complex Systems

Certain languages, such as English and French, are among the most complex languages to construct letter-to-sound rules for. These are not the only languages in this last category. Any language with an old writing system that has not undergone a modicum of spelling reform but has undergone dramatic phonological, morphonemic, and morphological changes will probably fall into this category.

9. Conclusions

We have presented the difficulties of grapheme-to-phoneme conversion for English and French. Both languages have evolved from different origins, and are the results of the historical influence of other languages from which words have been borrowed and assimilated, sometimes only partially. English and French have interacted and continue to interact with each other. For both languages, the spelling has been enforced by dictionaries and laws, but the pronunciation has continued to evolve, widening the gap between the written and spoken components of the language.

Both the English and French translation systems presented in this paper are based on rewriting rules. Nevertheless, some differences exist in the syntax and the interpretation of these rules. For a more theoretical approach to rewriting rules, see Kaplan and Kay (1994).

English is scanned once from right to left to better take into account the suffixes of the word, which in certain cases determine the stressed syllable. The rule transforms the grapheme into phonemes and stress marks used by the stress module. In some cases, the input string is modified to add a morpheme boundary, or to replace the suffix by another suffix to continue the conversion. Syllabification, stress, and allophonic rules are achieved by programs.

French uses the concept of a class that allows for the grouping of strings having a common property, thus reducing the number of rules. Several blocks of rules can be defined corresponding to different scans from left to right of the string (the output string replacing the input text at the end of a block of rules). The input string is not modified. Rules can check the left and right contexts of the input string, and the left context of the output string. French is not a stressed language, so there is no need for a syllabification module or a stress module.

Many problems persist for phonemicization. In English, suffix stripping, compound decomposition, and primary stressed syllable are very important to get the proper phoneme string, and are carried out mostly by rules without an explicit morph dictionary, contrary to as in Allen (1976), who uses a morph dictionary with 12,000 morphs, or Coker (1985), whose dictionary has 43,000 morphs. In French, the word-by-word conversion is probably simpler due to the absence of stressed syllables. Affixes do not alter the pronunciation of the root; compare, for instance, *photo, photograph,* and *photography* in English with *photo, photographe* and *photographie* in French. But in French, there are more interactions between words due to the linking problem (*nous avons*) and mute *e* (*chemin de fer*). These interactions are also dependent on speech rate.

Sometimes the homograph problem can be solved by looking at the left and right context of the word, but the general case requires a better understanding of the overall

[13] The synthetic index is $I_s = w \div m$, where w is a word and m is a morpheme.

structure of the sentence. This is also required to get a more natural prosodics in text-to-speech synthesizers.

The same formalism could be used for both English and French with a slight modification, for instance, of the French formalism. Blocks of rules should indicate if the scan is to be done from left to right or from right to left. In the right-to-left scan, the right context of the output buffer would be usable (instead of the left context if the scan is done from left to right). The word *scandalousness* could be decomposed by the following rules:

```
begin RL          RL for Right to Left
      10 : ness   →   +ness / ous   -  _ ;
      20 : ous    →   +ous  / al    -  ⟨+⟩, _ ;   + in the output buffer
end
```

resulting in *scandal+ous+ness*. Translating the root *scandal* could be done either from left to right or from right to left in one or more blocks of rules.

The required output phoneme string depends on the application. For speech synthesis, one output string is needed for a word. If several pronunciations are possible, the software has to produce only one for the synthesizer.

Speech recognition algorithms must know all the phonetic variations of the words in the vocabulary to be recognized, so the output should be a set of phonetic strings corresponding to the input word. Some rules must be declared optional, and the interpreter modified to take them into account.

For database searches, a set of equivalences can be devised where two (or more) phonemes or allophones could be considered correct. For example, in many cases [ə] and [ɨ] can be considered equivalents. Similarly, [ə][l] and [ʟ] (syllabic [l]) can also be considered equivalents. For French open [a] and close [ɑ] could be equivalent, as would be [o] and [ɔ], or [e] and [ɛ]. The search could even be done only on phoneme consonants (for proper name searches, for instance).

To our knowledge, learning algorithms, although promising, have not (yet) reached the level of rule sets developed by humans. The automatic discovery of the underlying structure of a language is not easy, nor is the developing of a universal rewriting rule formalism for the different languages.

Dictionaries and sets of rules will have to continue to coexist either as a dictionary of exceptions and a large set of rules, or as a large dictionary and a set of rules to deal with exceptions.

References

Ainsworth, W. A. 1973. A system for converting English text into speech. In *IEEE Transactions of Audio and Electroacoustics*, pages 288–290.

Allen, J. 1976. Synthesis of speech from unrestricted text. In *IEEE 64* (4), April.

Allen, J., R. Carlson, B. Granström, S. Hunnicutt, D. H. Klatt, and D. B. Pisoni. 1979. Conversion of unrestricted text-to-speech. Unpublished Monograph, Massachusetts Institute of Technology, Cambridge, MA.

Aubergé, V. 1991. La synthèse de la parole: "des règles au lexique". Thèse, Université Stendhal, Grenoble.

Bakiri, G., and T. G. Dietterich. 1991. *Converting English Text to Speech: A Machine Learning Approach*. Ph.D. thesis. Rep. No. 91-30-1. Department of Computer Science, Oregon State University.

Béchet F., T. Spriet, and M. El-Bèze. 1996. Traitement spécifique des noms propres dans un système de transcription graphème-phonème. JST Avignon.

Ben Crane, L., E. Yeager, and R. Whitman. 1981. *An Introduction to Linguistics*, Chapter 4: History of English. Little, Brown and Company.

Bernstein, J. and L. Nessly. 1981. Performance comparison of component algorithms for the phonemicization of orthography. In *Proceedings of the 19th*

Annual Meeting, Stanford University. Association for Computational Linguistics.

Burney, P. 1955. Que sais-je? *L'orthographe.* Collections.

Catach, N. 1978. Que sais-je? *L'orthographe.* Collections.

Catach, N. 1989. Informatique: Traitement automatique du Langage. *Bulletin Liaisons-Heso,* September.

Catach, N. and L. Catach 1992. Présentation du logiciel VOISINETTE, "Un correcteur à entrée phonétique". CNRS-INFOS.

Church, K. W. 1985. Stress assignment in letter to sound rules for speech synthesis. In *Proceedings of the 23rd Annual Meeting,* pages 246–253, University of Chicago. Association for Computional Linguistics.

Clemens, G. N. and S. J. Keyser. 1983. *CV Phonology: A Generative Theory of the Syllable. Linguistic Inquiry* Monograph Nine. MIT Press, Cambridge, MA.

Coker, C. H. 1985. A dictionary-intensive letter-to-sound program. *Journal of the Acoustical Society of America* Supplement 1, Vol. 78, S7.

Cotto, D. 1992. *Traitement automatique des textes en vue de la synthèse vocale.* Thèse, Université Paul Sabatier, Toulouse III.

Dedina, M. J. and H. C. Nusbaum. 1991. PRONOUNCE: A program for pronunciation by analogy. *Computer Speech and Language* 5:55–64.

DeFrancis, J. 1984. *The Chinese Language: Fact and Fantasy.* University of Hawaii Press.

Dirksen, A. and Coleman, J. 1994. All-Prosodic speech synthesis. Second ESCA/IEEE Workshop on Speech Synthesis.

Divay, M. 1984. *De l'écrit vers l'oral ou contribution à l'étude des traitements des textes écrits en vue de leur prononciation sur synthétiseur de parole.* Thèse d'Etat, Université de Rennes, France.

Divay, M. 1985. A text-processing expert system. 5ème Congrès Reconnaissance des formes et Intelligence Artificielle, Novembre 1985, Grenoble, France.

Divay, M. 1990a. Traitement du langage naturel: la phonétisation ou comment apprendre à l'ordinateur à lire un texte Français. *MICRO-SYSTEMES,* March.

Divay, M. 1990b. A written processing expert system for text to phoneme conversion. In *Proceedings of the International Conference on Spoken Language (ICSLP 90),* Kobe, Japan.

Divay, M. 1991. *CD-ROM Electronic Dictionary,* November.

Divay, M. 1994. Indexation phonétique et recherche documentaire. In *Proceedings of the 9ème Congrès AFCET,* Paris, Volume 2: Intelligence Artificielle.

Elovitz, H. S., R. W. Johnson, A. McHugh, and J. E. Shore. 1976. Automatic translation of English text to phonetics by means of letter-to-sound rules. NRL Report 7948, Naval Research Laboratory, Washington, D.C.

Golding, A. R. 1991. *Pronouncing Names by a Combination of Case-based and Rule-based Reasoning.* Ph.D. Thesis, Stanford University.

Golding, A. R. and P. S. Rosenbloom. 1993. A comparison of anapron with seven other name-pronunciation systems. *Journal of the American Voice Input/Output Society* 14:1–21.

Gonzalez, S. and J. P. Tubach. 1982. Réseaux connexionistes pour la traduction orthographique phonétique: Applications à l'Espagnol et au Français. 13ème Journées d'Études sur la Parole, Montreal, Canada.

Greenberg, J. 1990. A quantitative approach to the morphological typology of language. In K. Denning and S. Kemmer, editors, *On Language: Selected Writings of Joseph H. Greenberg.* Stanford University Press, Stanford.

Halle, M. and S. J. Keyser. 1971. *English Stress: Its Form, its Growth and its Role in Verse.* Harper and Row, New York.

Hertz, S. R. 1979. Appropriateness of different rule types in speech synthesis. In J. J. Wolf and D. H. Klatt, editors, *Speech Communication Papers,* No. 50, pages 511–514. Acoustical Society of America.

Hertz, S. R. 1981. SRS text-to-phoneme rules: A three-level rule strategy. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* pages 102–105.

Hertz, S. R. 1982. From text to speech with SRS. *Journal of the Acoustical Society of America* 72: 1155–1170.

Hertz, S. R. 1983. The "morphology" of English spelling: A look at the SRS text-modification rules for English. *Working Papers of the Cornell Phonetics Laboratory* 1: 17–28.

Hertz, S. R. 1985. A versatile dictionary for speech synthesis by rule. *Journal of the Acoustical Society of America,* Supplement 1:77, S11.

Hochberg, J., S. M. Mniszewski, T. Calleja, and G. J. Papcun. 1990. What's in a name?: Last names as a computational problem. Unpublished manuscript, Los Alamos National Laboratory, Los Alamos, NM.

Hochberg, J., S. M. Mniszewski, T. Calleja, and G. J. Papcun. 1991. A default hierarchy for pronouncing English. *IEEE Transactions on Pattern Matching and Machine Intelligence* 13(9): 957–964.

Hunnicut, S. 1976. Phonological rules for a text-to-speech system. *American Journal of Computational Linguistics*, Microfiche 57.

Hunnicut, S. 1980. *Grapheme to Phoneme Rules: A Review*. STL-QPSR 2-3.

Kaplan, R. M. and M. Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20(3).

Klatt, D. H. 1987. Review of text to speech conversion for English. *Journal of the Acoustical Society of America* 82(3): 737–793.

Klatt, D. H. and D. W. Shipman. 1982. Letter-to-phoneme rules: A semi-automatic discovery procedure. *Journal of the Acoustical Society of America* 82: 737–793.

Laporte, E. 1988. Méthodes algorithmiques et lexicales de phonétisation de textes. Thèse, Université Paris 7, May.

Levin, H. 1963. A basic research program on reading. Final Report, Cooperative Research Project No. 639, Cornell University.

Liberman, M. Y. and A. Prince. 1977. On stress and linguistic rhythm. *Linguistic Inquiry* 8(2): 249–336.

Lucas, S. M. and R. I. Damper. 1992. Syntactic neural networks for bi-directional text-phonetics translation. In G. Bailly and C. Benoit, editors, *Talking Machines, Theories, Models and Designs*. North-Holland Publishers.

Lucassen, J. M. and R. L. Mercer. 1984. An information theoretic approach to the automatic determination of phonemic baseforms. In *Proceedings of ICASSP-84*, pages 42.5.1–42.5.3, San Diego.

McCormick, S. and S. R. Hertz. 1989. A new approach to English text-to-phoneme conversion using delta, Version 2. 117th Meeting. *Journal of the Acoustical Society of America*, Supplement 1, Vol. 85, S124.

McIlroy, M. D. 1974. Synthetic English speech by rules. Bell Telephone Laboratories Memo.

Meng, H. M. 1995. *Phonological Parsing for Bi-Directional Letter-to-Sound and Sound-to-Letter Generation*. Ph.D. Thesis, MIT, Cambridge, MA.

O'Malley, M. H. 1990. Text-to-speech conversion technology. *Computer IEEE*, page 17.

O'Shaughnessy, D., M. Lennig, P. Mermelstein, and M. Divay. 1981. Simulation d'un lecteur automatique du Français. 12èmes Journées d'Études sur la Parole, Montreal, Canada.

Parfitt, S. and R. Sharman. 1991. A bi-directional model of English pronunciation. In *Proceedings of Eurospeech*, volume 2, pages 801–804.

Prouts, B. 1980. *Contribution à la synthèse de la parole à partir de texte; transcription graphémo-phonétique en temps réel sur microprocesseur*. Thèse de Docteur-Ingénieur, Université de Paris Sud, Orsay.

Rey, A., A. Duval, B. Vienne, B. Struyf, M. Divay, T. Lootens and S. Zimmermann. 1989. *Le Robert Electronique*. Ensemble d'Outils d'Aide à la Rédaction de Textes Français sur Disque Optique Compact (CD-ROM), November.

Sejnowski, T. J. and C. R. Rosenberg. 1987. NETtalk: Parallel networks that learn to pronounce English text. *Complex Systems* 1:145–168.

Spiegel, M. F. 1985. Pronouncing surnames automatically. In *Proceedings of AVIOS*.

Spiegel, M. F. and M. J. Machi. 1990. Synthesis of names by a demi-syllable-based speech synthesizer (Orator). *Journal of the American Voice Input/Output Society* 7:1–10.

Thimonnier, R. 1978. *Code Orthographique et Grammatical*. Collections Marabout.

Venezky, R. L. 1962. *A Computer Program for Deriving Spelling to Sound Correlations*. MA thesis, Cornell University. Published in part in *A Basic Research Program on Reading*. *See* Levin 1963.

Venezky, R. L. 1966. Automatic spelling-to-sound conversion. *Computation in Linguistics: A Case Book*. Indiana University Press, Bloomington, IN.

Venezky, R. L. 1967a. English orthography: Its graphical structure and its relation to sound. *Reading Research Quarterly*, II.

Venezky, R. L. 1967b. Reading: Grapheme-phoneme relationships. *Education* 87: 519–524.

Venezky, R. L. 1967c. The basis of English orthography. *Acta Linguistica* 10: 145–159.

Venezky, R. L. 1970. *The Structure of English Orthography*. Mouton, The Hague.

Venezky, R. L. and R. Weir. 1966. A study of selected spelling-to-sound correspondence patterns. Final Report, Cooperative Research project No. 3090, Stanford University.

Vitale, A. J. 1991. An algorithm for high accuracy name pronunciation by parametric speech synthesizer. *Computational Linguistics* 17(3).

Yarowsky, D. 1994. Homograph disambiguation in text-to-speech synthesis. Second ESCA/IEEE Workshop on Speech Synthesis.

Yvon, F. 1996. *Prononcer par analogie: motivation, formalisation et évaluation.* Thèse, Ecole nationale des Télécommunications, Paris.

Weir, R. 1964. Formulation of grapheme-phoneme correspondence rules to aid the teaching of reading. Final Report, Cooperative Research project No. S-039 Stanford University.

Wells, J. C. 1982. *Accents of English, An Introduction, Chapter 3.* Cambridge University Press.