USING AN ON-LINE DICTIONARY TO FIND RHYMING WORDS AND PRONUNCIATIONS FOR UNKNOWN WORDS

Roy J. Byrd

I.B.M. Thomas J. Watson Research Center
Yorktown Heights, New York 10598

Martin S. Chodorow

Department of Psychology, Hunter College of CUNY
and
I.B.M. Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT

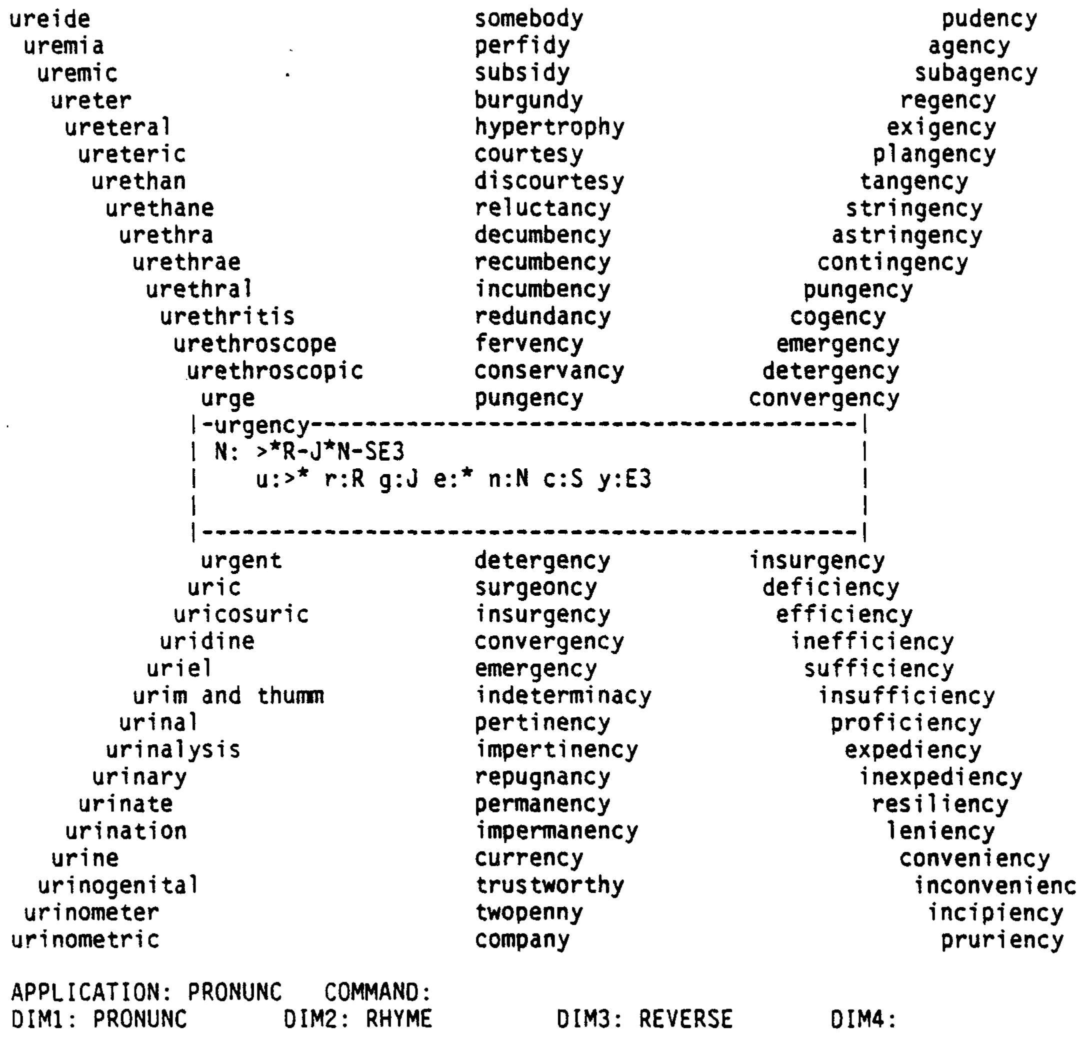
Humans know a great deal about relationships among words. This paper discusses relationships among word pronunciations. We describe a computer system which models human judgement of rhyme by assigning specific roles to the location of primary stress, the similarity of phonetic segments, and other factors. By using the model as an experimental tool, we expect to improve our understanding of rhyme. A related computer model will attempt to generate pronunciations for unknown words by analogy with those for known words. The analogical processes involve techniques for segmenting and matching word spellings, and for mapping spelling to sound in known words. As in the case of rhyme, the computer model will be an important tool for improving our understanding of these processes. Both models serve as the basis for functions in the WordSmith automated dictionary system.

1. Introduction

This paper describes work undertaken in the development of WordSmith, an automated dictionary system being built by the Lexical Systems group at the IBM T. J. Watson Research Center. WordSmith allows the user to explore a multidimensional space of information about words. The system permits interaction with lexical databases through a set of programs that carry out functions such as displaying formatted entries from a standard dictionary and generating pronunciations for a word not found in the dictionary. WordSmith also shows the user words that are "close" to a given word along dimensions such as spelling (as in published dictionaries), meaning (as in thesauruses), and sound (as in rhyming dictionaries).

Figure 1 shows a sample of the WordSmith user interface. The current word, *urgency*, labels the text box at the center of the screen. The box contains the output of the PRONUNC application applied to the current word; it shows the pronunciation of *urgency* and the mapping between the word's spelling and pronunciation. PRONUNC represents pronunciations in an alphabet derived from *Webster's Seventh Collegiate Dictionary*. In the pronunciation shown "*" represents the vowel *schwa*, and ">" marks the vowel in the syllable bearing primary stress. Spelling-to-pronunciation mappings will be described in Section 3.

Three dimensions, displaying words that are neighbors of *urgency*, pass through the text box. Dimension one, extending from *uriede* to *urinometric*, contains words from the PRONUNC data base which are close to *urgency* in alphabetical order. The second dimension (from *somebody* to *company*) shows words which are likely to rhyme with *urgency*. Dimension three (from *pudency* to *pruriency*) is based on a reverse alphabetical ordering of words, and displays words whose spellings end similarly to *urgency*. The RHYME and REVERSE dimensions are discussed below.



```
ureide                            somebody                           pudency
 uremia                           perfidy                           agency
  uremic                          subsidy                          subagency
   ureter                         burgundy                        regency
    ureteral                      hypertrophy                    exigency
     ureteric                     courtesy                      plangency
      urethan                     discourtesy                  tangency
       urethane                   reluctancy                  stringency
        urethra                   decumbency                 astringency
         urethrae                 recumbency                contingency
          urethral                incumbency               pungency
           urethritis             redundancy              cogency
            urethroscope          fervency               emergency
             urethroscopic        conservancy           detergency
              urge                pungency             convergency
             |-urgency-----------------------------------------|
             | N: >*R-J*N-SE3                                  |
             |    u:>* r:R g:J e:* n:N c:S y:E3                |
             |                                                 |
             |-------------------------------------------------|
              urgent              detergency           insurgency
             uric                 surgeoncy             deficiency
            uricosuric            insurgency             efficiency
           uridine                convergency             inefficiency
          uriel                   emergency                sufficiency
         urim and thumm           indeterminacy             insufficiency
        urinal                    pertinency                 proficiency
       urinalysis                 impertinency                expediency
      urinary                     repugnancy                   inexpediency
     urinate                      permanency                    resiliency
    urination                     impermanency                   leniency
   urine                          currency                        conveniency
  urinogenital                    trustworthy                      inconvenienc
 urinometer                       twopenny                          incipiency
urinometric                       company                            pruriency

APPLICATION: PRONUNC   COMMAND:
DIM1: PRONUNC       DIM2: RHYME         DIM3: REVERSE        DIM4:
```

Figure 1. WordSmith User Interface.

Section 2 describes the construction of the WordSmith rhyming dimension, which is based on an encoding procedure for representing pronunciations. The encoding procedure is quite flexible, and we believe it can be used as a research tool to investigate the linguistic and psycholinguistic structure of syllables and words. Section 3 outlines a program for generating a pronunciation of an unknown word based on pronunciations of known words. There is evidence (Rosson, 1985) that readers sometimes generate a pronunciation for an unfamiliar letter string based on analogy to stored lexical "neighbors" of the string, i.e. actual words that differ only slightly in spelling from the unfamiliar string. A program which generates pronunciations by analogy might serve as a supplement to programs that use spelling-to-sound rules in applications such as speech synthesis (Thomas, *et al.*, 1984), or it might be used to find rhyming words, in WordSmith's rhyming dimension, for an unknown word.

2. Rhyme

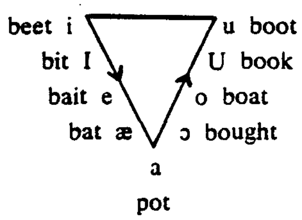
The WordSmith rhyme dimension is based on two files. The first is a main file keyed on the spelling of words arranged in alphabetical order and containing the words' pronunciations organized according to part of speech. This same file serves as the data base for the PRONUNC application and dimension shown in Figure 1. The second file is an index to the first. It is keyed on encoded pronunciations and contains pointers to words in the main file that have the indicated pronunciations. If a single pronunciation corresponds to multiple spellings in the main file, then there will be multiple pointers, one for each spelling. Thus, this index file also serves as a list of homophones. The order of the encoded pronunciations in the index file defines the rhyming dimension so that words which are close to one another in this file are more likely to rhyme than words which are far apart.

The original motivation for the encoding used to obtain the rhyme dimension comes from published reverse dictionaries, some of which (e.g., Walker, 1924) even call themselves "rhyming dictionaries". Such reverse dictionaries are obtained from a word list by (a) writing the words right-to-left, instead of left-to-right, (b) doing a normal alphabetic sort on the reversed spellings, and (c) restoring the original left-to-right orientation of the words in the resulting sorted list. This procedure was used to derive the REVERSE dimension shown in Figure 1.

There are several problems with using reverse dictionaries as the basis for determining rhymes. First, since English spelling allows multiple ways of writing the same sounds, words that in fact do rhyme may be located far apart in the dictionary. Second, since English allows a given spelling to be pronounced in multiple ways, words that are close to one another in the dictionary will not necessarily rhyme with each other. Third, the location of primary stress is a crucial factor in determining if two words rhyme (Rickert, 1978). Primary stress is not encoded in the spelling of words. As an extreme example of this failure of reverse dictionaries, note that the verb *record* does not rhyme with the noun *record*. Fourth, basing rhyme on the reverse linear arrangement of letters in words gives monotonically decreasing weight to the vowels and consonants as one moves from right to left in the word. This procedure does not capture the intuition that the vowel in the syllable bearing primary stress and the vowels following this syllable are more significant determiners of rhyme than are the consonants. For example, we feel that as a rhyme for *urgency, fervency* would be better than *agency*. A reverse dictionary, however, would choose the latter. More specifically, even if the difficulties associated with spelling differences were overcome, a reverse dictionary would still accord more weight to the /g/ consonant sound of *agency* than to the /ər/ vowel sound of *fervency*, contrary to our intuitions.

As already indicated, our procedure uses word pronunciations rather than spellings as the basis for the rhyme dimension. A total of more than 120,000 pronunciations from Webster's Seventh Collegiate Dictionary have been submitted to the encoding process. The first step in encoding replaces the symbols in the pronunciation representations with single-byte codes representing phonetic segments. The procedure which maps segments to byte codes also allows different segments to be mapped into a single code, in effect defining equivalence classes of segments. For example, the French *u* sound in *brut* is mapped onto the same segment as the English long *u* sound in *boot*. This is the same mapping that most English speakers would make.

In the mapping currently in use, all vowels are organized linearly according to the vowel triangle. At one end of the spectrum is the long *e* sound in *beet* (/i/). At the other end is the long *u* sound in *boot* (/u/).



The diphthongs are organized into two subseries, one for rising diphthongs and the other for falling ones. As with the vowels, each subseries is a linear arrangement of the diphthongs according to the position of the initial sound on the vowel triangle. The consonants are similarly organized into several subseries. There are voiced and voiceless stops, voiced and voiceless fricatives and affricates, nasals, and liquids.

An important point about this mapping from pronunciation patterns to phonetic segments is that it is flexible. Both the phonetic equivalence classes and the collating sequence can be easily changed. The system can thus serve as the basis for experimentation aimed at finding the precise set of phonetic encodings that yield the most convincing set of rhymes.

The second encoding step arranges the segments for a pronunciation in the order representing their importance for determining rhyme. This ordering is also the subject of continuing experimentation. The current arrangement is as follows:

(1) All segments preceding the syllable bearing primary stress are recorded in the order that they occur in the pronunciation string.

(2) All consonantal segments in and following the syllable bearing primary stress are added to the encoding in the order in which they occur.

(3) All vocalic segments (vowels and diphthongs) in and following the syllable bearing primary stress are placed before any segments for trailing consonants in the final syllable. If there are no trailing consonants in the final syllable, then these vocalic segments are placed at the end of the encoding.

Note that this scheme preserves the order of the segments preceding the point of primary stress, as well as those in the final syllable. For words where primary stress occurs before the final syllable, the vowels are raised in importance (with respect to rhyming) over all consonants except final ones. This procedure allows us to capture the intuition that *fervency* is a better rhyme for *urgency* than *agency*.

The final step in the encoding procedure reverses the phonetic segment strings right-for-left, groups them according to the position of the syllable bearing primary stress (i.e., the distance of that syllable from the end of the word) and sorts the groups just as in the production of reverse dictionaries. The difference is that now neighbors in the resulting sorted list have a better chance of rhyming because of the use of pronunciations and the application of our intuitions about rhymes.

We note that the resulting lists of rhymes are not perfect. This is so first because we have not completed the experiments which will result in an "optimal" set of intuitions about the encoding process. One planned experiment will clarify the position of the *schwa* vowel in the vowel triangle. Another will study intervocalic consonant clusters which, especially when they contain nasals or liquids, result in less successful rhymes. A third study will allow us to identify "discontinuity" in the rhyme list, across which rhyming words are very unlikely to be found. In Figure 1., a discontinuity seems to occur between *currency* and *trustworthy*.

The second reason that our rhyme lists are not perfect is that it is unlikely that any single dimension will be sufficient to guarantee that all and only good rhymes for a given word will appear adjacent to that word in the dimension's order, if only because different people disagree on what constitutes "good" rhyme.

Examples

We give two sequences of words selected from the WordSmith RHYME dimension.

antiphonary
dictionary
seditionary
expeditionary
missionary

These five words have their primary stress in the forth syllable from the right, and they also have the same four vowel sounds from that point onwards. Notice that the spelling of *antiphonary* would place it quite far from the others in a standard reverse dictionary. In addition, the extra syllables at the beginning of *antiphonary*, *seditionary*, and *expeditionary* are irrelevant for determining rhyme.

write
wright
rite
right

These four words, each a homonym of the others, share a single record in the rhyming index and are therefore adjacent in the WordSmith RHYME dimension.

3. Pronunciation of Unknown Words

Reading aloud is a complex psycholinguistic process in which letter strings are mapped onto phonetic representations which, in turn, are converted into articulatory movements. Psycholinguists have generally assumed (Forster and Chambers, 1973) that the mapping from letters to phonemes is mediated by two processes, one based on rules and the other based on retrieval of stored pronunciations. For example, the rule *ea* -> /i/ converts the *ea* into the long e sound of *leaf*. The other process, looking up the stored pronunciation of a word, is responsible for the reader's rendering of *deaf* as /dɛf/, despite the existence of the *ea* -> /i/ rule. Both processes are believed to operate in the pronunciation of known words (Rosson, 1985).

Until recently, it was generally assumed that novel words or pseudowords (letter strings which are not real words of English but which conform to English spelling patterns, e.g., *heaf*) are pronounced solely by means of the rule process because such strings do not have stored representations in the mental lexicon. Hcwever, Glushko (1979) has demonstrated that the pronunciation of a pseudoword is influenced by the existence of lexical "neighbors," i.e., real words that strongly resemble the pseudoword. Pseudowords such as *heaf*, whose closest neighbors (*leaf* and *deaf*) have quite different pronunciations, take longer to read than pseudowords such as *hean*, all of whose close neighbors have similar pronunciations (*dean*, *lean*, *mean*, etc.). (It has been assumed that words which differ only in initial consonants are "closer" neighbors than those which differ in other segments.) Glushko has also demonstrated an effect of lexical neighbors on the pronunciation of familiar words of English.

The picture that emerges from this psychological work depicts the retrieval process as selecting all stored words which are similar to a given input. If the input is not found in this set (i.e., the input is a novel word or pseudoword), its pronunciation is generated by analogy from the pronunciations that are found. Analogical processing must take note of the substring common to the input and its neighbors (*ean* in the case of *hean*), use only this part of the pronunciation, and make provision for pronouncing the substring which is different (*h*). When the pronunciations of the lexical neighbors are consistent, the pronunciation of the pseudoword can be generated by the reader more quickly than when the pronunciations are inconsistent.

There are of course many unanswered questions about how readers actually generate pronunciations by analogy. One approach to answering the questions is to build a computational system that can use various strategies for finding lexical neighbors, combining partial pronunciations, etc., and then compare the output of the system to the pronunciations produced by human readers. The following is an outline of such a computational system.

Two WordSmith files will be used to support a proposed program that generates pronunciations for unknown words based on stored pronunciations of known words. The first is a main file which is keyed on the spelling of words and which contains pronunciations organized according to part of speech. This is the file which supported the PRONUNC and RHYME WordSmith functions described earlier. In this file, each pronunciation of a word has stored with it a mapping from its phonetic segments onto the letters of the spelling of the word. These mappings were generated by a PROLOG program that uses 148 spelling-to-pronunciation rules for English (e.g., ph -> /f/). The second file is an index to the main file keyed on reverse spelling. This file is equivalent to the one which supports the REVERSE WordSmith dimension shown in Figure 1.

The strategy for generating a pronunciation for an unknown word is to find its lexical neighbors and produce a pronunciation "by analogy" to their pronunciations. The procedure is as follows: (a) Segment the spelling of the unknown word into substrings. (b) Match each substring to part of the spelling of a known word (or words). (c) Consult the spelling-to-pronunciation map to find the pronunciation of the substring. (d) Combine the pronunciations of the substrings into a pronunciation for the unknown word.

These steps are illustrated below for the unknown word *brange*.

(a) Segmentation

```
brange
<-->    initial substring
 <---> final substring
```

Strategies for segmentation will be discussed later.

(b) Matching

bran is the longest initial substring in *brange* that matches a word-initial substring in the dictionary. The word *bran* is a dictionary entry, and 20 other words begin with this string.

range is the longest final substring in *brange* that matches a word-final substring in the dictionary. The match is to the word *range*. In the reverse spelling file, 22 other words end in *ange*.

(c) Pronunciation of substrings

All 21 words that have the initial string match for *bran* have the mapping

```
b  r  a  n
|  |  |  |
b  r  æ  n
```

In 20 of the 23 words that match word-final *ange*, the mapping is

```
a  n  ge
|  |  |
e  n  j   as in range (/renj/)
```

The other three words are *flange* (/ænj/), *orange* (/Inj/), and *melange* (/anj/).

(d) Combining pronunciations

From the substring matches, the pronunciations of /b/, /r/, /n/, /g/, and /e/ are obtained in a straightforward manner, but pronunciation of the vowel *a* is not the same in the *bran* and *ange* substrings. Thus, two different pronunciations emerge as the most likely renderings of *brange*. (i) below is modelled after *range* or *change*, and (ii) is modelled after *bran* or *branch*.

```
(i) b  r  a  n  ge
    |  |  |  |  |
    b  r  e  n  j
(ii)b  r  a  n  ge
    |  |  |  |  |
    b  r  æ  n  j
```

Here, pronunciation by analogy yields two conflicting outcomes depending upon the word model selected as the lexical neighbor. If people use similar analogical strategies in reading, then we might expect comparable disagreements in pronunciation when they are asked to read unfamiliar words. A very informal survey we conducted suggests that there is considerable disagreement over the pronunciation of *brange*. About half of those we asked preferred pronunciation (i), while the others chose (ii).

In the example shown above, segmentation is driven by the matching process, i.e. the substrings chosen are the longest which can be matched in the main file and the reverse spelling file. There are, of course, other possible strategies of segmentation, including division at syllable boundaries and division based on the onset-rhyme structure within the syllable (for *brange*, *br* + *ange*). Evaluation of these alternative methods must await further experimentation.

There are other outstanding questions related to the Matching and Combining steps. If matches cannot be found for initial and final substrings that overlap (as in the example) or at least abut, then information about the pronunciation of an internal substring will be missing. Finding a match for an internal substring requires either a massive indexing of the dictionary by letter position, a time consuming search of the standard indexes, or the development of a clever algorithm. With regard to combining substring pronunciations, the problem of primary stress assignment arises when primary stress is absent from all of the substrings or is present at different locations in two or more of them. Finally, there is a question of the weight that should be assigned to alternative pronunciations generated by this procedure. Should a match to a high frequency word be preferred over a match to a low frequency word? Is word frequency more important than the number of matching substrings which have the same pronunciation? These are empirical psycholinguistic questions, and the answers will no doubt help us generate pronunciations that more closely mirror those of native English speakers.

4. Conclusion

The two applications described here, finding rhyming words and generating pronunciations for unknown words, represent some ways in which the tools of computational linguistics can be used to address interesting psycholinguistic questions about the representation of words. They also show how answers to these psycholinguistic questions can, in turn, contribute to work in computational linguistics, in this case to development of the WordSmith on-line dictionary.

Acknowledgements

We are grateful to Barbara Kipfer for her preliminary work on the syllabification of unknown words, and to Yael Ravin and Mary Neff for comments on earlier versions of this report.

References

Forster, K. and Chambers, S. (1973), Lexical access and naming time. *Journal of Verbal Learning and Verbal Behavior*, 12, 627-635.

Glushko, R. (1979), The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology*, 5, 674-691.

Rickert, W.E. (1978), Rhyme terms. *Style*, 12(1), 35-46.

Rosson, M.B. (1985), The interaction of pronunciation rules and lexical representations in reading aloud. *Memory and Cognition*, in press.

Thomas, J., Klavans, J., Nartey, J., Pickover, C., Reich, D., and Rosson, M. (1984), WALRUS: A development system for speech synthesis. IBM Research Report RC-10626.

Walker, J. (1924), *The Rhyming Dictionary*, Routledge and Kegan Paul, London.

Webster's Seventh Collegiate Dictionary (1967), Merriam, Springfield, Massachusetts.