Phonological Parsing and Lexical Retrieval

Kenneth W. Church

Abstract

Speech is a natural error-correcting code. The speech signal is full of rich sources of contextual redundancy at many levels of representation including allophonic variation, phonotactics, syllable structure, stress domains, morphology, syntax, semantics and pragmatics. The psycholinguistic literature has tended to concentrate heavily on high level constraints such as semantics and pragmatics and has generally overlooked the usefulness of lower level constraints such as allophonic variation. It has even been said that allophonic variation is a source of confusion or a kind of statistical noise that makes speech recognition that much harder than it already is. In contrast, I argue that aspiration, stop release, flapping, palatalization and other cues that vary systematically with syllabic context can be used to parse syllables and stress domains. These constituents can then constrain the lexical matching process, so that much less search will be required in order to retrieve the correct lexical entry. In this way, syllable structure and stress domains will be proposed as an intermediate level of representation between the phonetic description and the lexicon.

My argument is primarily a computational one and will include a discussion of a prototype phonetic parser which has been implemented using simple well-understood parsing mechanisms. No experimental results will be presented.

[Note to the editor: A number of the reviewers have complained quite correctly about my IPA notation. I had intended to design some IPA text macros like what I had at MIT, but so far, I simply haven't had the time. Please replace my IPA characters with your standard. Here is a partial definition of my IPA notation:

x	schwa
|	front schwa (often written 'I' with bar)
J	'judge'
C	'church'
&	flap
?	glottal stop as 'atlas'
H	aspiration
>	unreleased stop

Hopefully the rest of my notation should be self explanatory.]

1. Allophonic Variation is ''Noise''?

The lack of phonemic invariance has always been considered one of the classic hard problems for speech research*. It is well known that phonemes have very different acoustic phonetic properties in different contexts. For instance, the phoneme /t/ in ''Tom'' will almost certainly be released and aspirated unlike the /t/s in ''cat'', ''butter'', ''atlas'', etc. Moreover, it is also well known that different phonemes can be realized with almost the same allophone (phonetic variant). For example, /t/ and /d/ conflate in flapping contexts (e.g., ''writer''/''rider''), vowels often neutralize to schwa when unstressed, /dy/ and /J/ merge under palatization (''did you''), and so forth. Thus the mapping between phonemes and their allophonic realization is highly variable and extremely ambiguous; there are many ways that a speaker might choose to realize a given phoneme and there are many ways that a listener might choose to interpret a given allophone. This uncertainty is generally considered to be problematic for practical applications of speech technology (e.g., speech synthesis and recognition) as well as scientific investigations of human production

* This research was supported (in part) by the National Institutes of Health Grant No. 1 P01 LM 03374-01 and 03374-02 from the National Library of Medicine.

and perception.

The view of allophonic variation as problematic was commonly held during the time of the ARPA speech project; one can find support for this position from both the psychological [Cole and Jakimik] (CMU) and computational literatures [Jelinek] (IBM). This position was most clearly presented by Klatt (MIT), who argued that even if we had an ideal front-end that could produce an error-free phonetic transcription, it would still be very difficult to build a speech recognition device that could decode a phonetic transcription such as [dIJxhI&It|tam] into the sentence ‘‘Did you hit it to Tom,’’ because the decoding device would have to account for at least five allophonic rules:

- Palatalization of /d/ before /y/ in ‘‘did you’’
- Reduction of unstressed /u/ to schwa in ‘‘you’’
- Flapping of intervocalic /t/ in ‘‘hit it’’
- Reduction of schwa and devoicing of /u/ in ‘‘to Tom’’
- Reduction of geminate t in ‘‘it to’’

Klatt concluded that

> ‘‘In most systems for sentence recognition, such modifications must be viewed as a kind of ‘noise’ that makes it more difficult to hypothesize lexical candidates given an input phonetic transcription. To see that this must be the case, we note that each phonological rule results in irreversible ambiguity -- the phonological rule does not have a unique inverse that could be used to recover the underlying phonemic representation for a lexical item. For example, in a word like ‘about’ or the surface realization of almost any English vowel appearing in a sufficiently destressed word. The tongue flap [&] could have come from a /t/ or a /d/.’’ [Klatt80, pp. 548-549]

2. Allophonic Variation is Useful

In contrast, I argue that allophonic variation is useful. When I find a flap, I know that it is foot internal (not initial in a stressed syllable); when I find a reduced vowel, I know that is has zero stress; and when I find an aspirated stop, I know it probably starts a syllable. These examples demonstrate the well-known fact that allophonic distribution determines, in part, the location of syllables and stress. Thus, *in principle*, there ought to be important information in the distribution of allophones, though admittedly, *in practice*, distributional information is often confusing to existing speech recognizers, as Klatt points out, because the existing technology doesn’t know how to take advantage of it. We need a principled mechanism for extracting distributional information from the speech signal. I will propose that well-understood parsing technology is ideally suited to fill this gap. The next-generation recognizer ought to be able to parse a phonetic description of an utterance into a limited set of well-formed syllable structures and stress domains. These constituents will then be used to constrain the lexical matching process, so that much less search will be required in order to retrieve the correct lexical entry.

Let me illustrate the approach with Klatt’s example (enhanced with allophonic diacritics to show aspiration and glottalization): [dIJxhI&I?tH|tHam]. Using phonotactic and allophonic constraints on syllable structure such as:*

- /h/ is always syllable initial,
- [&] is always syllable final,

---

* This formulation of the constraints is oversimplified for expository convenience; see [Fujimura and Lovins, Kahn, Kiparsky] and references therein for discussion of the more subtle issues.

- [?] is always syllable final, and
- [tH] is always syllable initial.

the parser can insert the following syllable boundaries:

[dIJx # hI& # I? # tH| # tHam]

With most of the syllables correctly parsed, it is now relatively easy to match the constituents with words in the lexicon to find the sentence: ‘‘Did you hit it to Tom?’’.

Parsed Transcription		Decoding
[dIJx]	→	did you
[hI&]	→	hit
[I?]	→	it
[tH|]	→	to
[tHam]	→	Tom

This example shows that allophonic variation should not be viewed as noise; even though allophonic rules are extremely difficult to invert, they can be very powerful cues for a syllable parser.

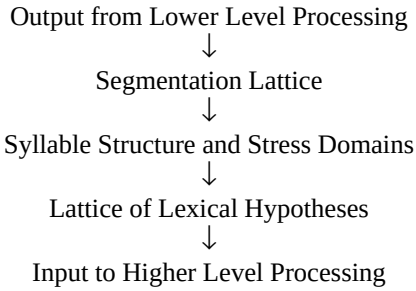
Thus far, the discussion has been assuming perfect input transcriptions, an idealization that is obviously unrealistic in practice. This idealization is convenient for exposition, and permits me to demonstrate the usefulness of allophonic constraints as parsing cues without having to discuss control and representation strategies (e.g., island driving) for dealing with errorful input. On the other hand, being an engineer, I really would like to expand upon these issues, and I have already begun to apply parsing techniques in limited practical problems such as the recognition of connected digit sequences. The parsing mechanisms have been extended to accept extremely ambiguous and errorful input, produced, in fact, by a statistical training procedure that generates a probability score for each allophone every 2 msec. Parsing these scores requires considerable engineering knowhow since straightforward methods would be too slow to deal with the extremely long input sentences of approximately 1000 speech frames. Let me resist the temptation to discuss these complex and interesting questions here, and simply continue to assume ideal (though unrealistic) input, for the remainder of this paper.

3. Syllable Structure as an Intermediate Level of Representation

The proposed solution to Klatt’s puzzle demonstrated the role of syllables and other linguistically motivated constituents as an intermediate level of representation between the input transcription and the output word lattice. In this way, I have factored the lexical retrieval problem into two subtasks:

- *Parse* the segment lattice into the intermediate representation, and
- *Match* the intermediate constituent structure against the lexicon.

This parsing and matching approach differs from many psychological/computational models (e.g., the Cohort model, IBM’s Markov model) which attempt a direct match between acoustic/phonetic segments and the lexicon, and do not assume an intermediate constituent structure.



A few psycholinguists have noticed the usefulness of allophonic rules as parsing cues. Two examples will be cited here:

> ‘‘In evaluating these data, linguists will not be surprised that aspiration on a voiceless stop in English serves as a cue that that stop is in syllable initial position. Such a result would be expected to follow from the fact that in English aspirated stops appear only in initial position in a stressed syllable. One would like to suggest that allophones are more than excess baggage through which a listener must work his way, and that they actually can carry information about the structure of the message. Such a suggestion was made and discussed in Lehiste (1960). But... By contrast, in the present experiment... Lehiste’s suggestion that the allophones can serve to carry information about the structure of the message is thus confirmed.’’ [Christie, p. 820]

Nakatani has also noted that so-called word boundary rules can be used as very powerful parsing cues:

> ‘‘We see the fission and fusion processes fitting into a comprehensive model of speech perception in the following way. The prosodic stress and rhythm cues indicate when a word boundary is likely to occur, but they do not indicate precisely where the word boundary is. For example, a word boundary must occur somewhere between the syllabic nuclei of ‘gray twine’ since two consecutive stressed syllables cannot belong to a single English word. The precise location of the word boundary is indicated by the allophonic cues. In this example, the word boundary must fall before the /t/ because its strong aspiration is characteristic of word-initial allophones of voiceless stops.’’ [In contrast, in ‘great wine’, the word boundary must fall after the /t/ because its realization is characteristic of word-final allophones.] [Nakatani, p. 3]

4. Review of Phonotactic Constraints

In addition to allophonic constraints, the parsing approach is also well suited to take advantage of phonotactics, the constraints on ordering (tactics) of phonemes. Most models of speech recognition and perception operate with an inadequate model of phonotactic constraints of the form:

word → syllable*
syllable → initial-cluster vowel final-cluster

This model assumes a list of permissible initial and final clusters. How can I estimate the set of permissible clusters? it is widely assumed that the set of *syllable* initial clusters are identical to the set of *word* initial clusters, and that the set of *syllable* final clusters are identical to the set of *word* final clusters. But this is inadequate for two reasons. First, it will be necessary to add some more constraints (e.g., the maximal onset principle and stress resyllabification) so that words like ‘‘extra’’ are assigned just one syllabification /ek-stra/, not three: /ek-stra/, */eks-tra/ and */ekst-ra/. Secondly though, the set of syllable final clusters is far more constrained that the set of word final clusters. Note, for example, if any word final cluster could end a syllable then we might expect to find a word final cluster like /rnt/ to appear in word medial position followed by another syllable. But sequences of consonants like /rntstr/ are not found in word medial position. It will be argued that sequences of six consonants can be excluded for principled reasons. In practice, one rarely finds sequences with as many as four consonants within the same morpheme. Long sequences of consonants almost always cross a morpheme boundary (e.g., ‘‘iN-STRructive’’, ‘‘eX-

PLanation'', ''thaNKSGiving''). Phonotactic constraints such as this might improve the performance of a speech recognition device in many ways. In particular, these constraints could lead to more compact representation of lexical entries, and more efficient search procedures for matching portions of the utterance against the lexicon.

According to Kiparsky's review paper of developments in suprasegmental phonology [Kiparsky], there are three types of phonotactic constraints on syllable structure (excluding morpheme final syllables for the moment):

- *Length*: There are at most three consonants before the vowel and two afterward
- *Sonority*: The relative prominence (vowel-likeness) must decrease from the nucleus towards the margins. More precisely, the sonority order (stops < fricatives < nasals < liquids < glides < vowels) must strictly decrease from the nucleus.
- *Idiosyncratic systematic gaps*: English happens not to allow certain possibilities, e.g., voiced stops after nasals and [u], even though these meet the previous two conditions.

In addition, there are also some well-known voicing and place assimilation and dissimilation facts.

4.1 The Affix Position

All of these constraints can be violated in morpheme final syllables.

- *Length*: wild, world, paint, exempt, lounge, texts, estrangedst
- *Sonority*: apse, axe, bets, width, depth, paths, laughs
- *Idiosyncratic systematic gaps*: bind, sand, mend, wood, hood, could, should
- *Place Assimilation*: seemed, seems, sings, hanged

In order to account for these exceptions, many analyses introduce a morpheme final *affix* position. This position was originally intended to hold regular inflectional affixes (e.g., plural /s, z/ and past tense /t, d/), though it has been generalized to apply whenever there is an apparent violation of one of the phonotactic constraints mentioned above. Thus, for example, the /d/ is analyzed as an affix in both ''fined'' and ''find,''* even though the latter /d/ is not a regular infectional affix. The consonants in the affix position must be coronal and they must agree in voicing.

It also turns out that a number of phonotactic principles can be violated in certain other positions. For example, /s/ doesn't seem to to fit into the sonority hierarchy; it to be less sonorous than a stop in ''stress'' but more sonorous in ''most''. One solution is to consider /st/, /sp/, /sk/ and other s-clusters to be a single ''unit,'' and thereby exempt from the constraints. In addition, this move would permit the length constraint to be tightened so that onsets, as well as codas, would be limited to at most two ''units''.

4.2 The Maximal Onset Principle and Stress Resyllabification

Phonotactic constraints are not sufficiently strong to produce a unique syllabification in all cases. For example, in words like ''wesTern'', ''winTer'', ''waTer'', ''veTo'', ''diviniTy'', and so forth, phonotactic constraints do not say which way the /t/ will attach. Most theories of syllable structure invoke ''tie-breaking principles'' for words such as these. Two very common ''tie-breaking principles'' are the *Maximize Onset Principle* and *Stress Resyllabification*. By the first principle, there is a preference toward assigning consonants to the onset of the following syllable rather than the coda of the previous syllable. Thus, for example, the /t/ in ''reTire'' will be assigned to the onset of the second syllable (i.e., ''re-tire''), and not to the coda of the first syllable (i.e., ''*ret-ire''). By the second principle, there is a preference

---

* Note that ''find'' violates the length constraint because it has three post-vocalic consonants /ynd/. It also violates the prohibition in English against voiced obstruents after nasals.

toward assigning consonants to a stressed syllable over an unstressed one. Thus, for example, the /k/ in ‘‘record’’ will be assigned to the first syllable when it is stressed (i.e., ‘‘REC-ord’’), and to the second syllable when it is stressed (i.e., ‘‘re-CORD’’).

- *Maximize Onsets*: When phonotactic and morphological constraints permit, maximize the number of consonants in onset position (e.g., ‘‘re-tire’’).
- *Stress Resyllabification*: When phonotactic and morphological constraints permit, maximize the number of consonants in stressed syllables (e.g., ‘‘re-CORD’’ vs. ‘‘REC-ord’’).

The maximal onset principle is relatively uncontroversial. Stress resyllabification, on other hand, has been the subject of considerable debate. Stress resyllabification is intended to adjust syllable boundaries so that it will be possible to account for stress dependent phonetic and phonemic facts at the level of syllable structure without direct reference to stress. Thus, in order to account for the phonetic and phonemic differences in a pair like ‘‘reCORD’’/‘‘RECord’’, the stress resyllabification principle will syllabify the /k/ into the stressed syllable, and then define the phonetic and phonemic rules to predict the observed phonetic and phonemic differences from the different syllable structures. In a very similar fashion, Kahn and other proponents of ambi-syllabicity, would assign two different syllable structures to ‘‘reCORD’’/‘‘RECord’’ in order to account for the different phonetic and phonemic possibilities and then define the phonetic rules to run off each of these structures appropriately. Similarly, metrical phonology would account for the difference by assigning two different foot structures: re[cord] and [record], and then account for the phonetic difference by defining the phonetic rules to work differently when the /k/ is in foot initial (pre-stress) position than when it is in foot internal (post-stress) position. Other solutions are also possible.

In summary, although there is still room for considerable debate over many of the details, there is a fairly well established general consensus. Each of the elements of the syllable (i.e., onset, peak, coda, affix) are highly restricted in a number of ways (e.g., length, sonority, place and voicing assimilation). The boundary between syllables is somewhat less well understood, though it too is governed by well-studied principles (e.g., maximal onset and stress resyllabification). All of these constraints can be encoded into a phrase structure grammar so that the parser can take advantage of phonotactic constraints in order to restrict the set of possible syllable assignments.

5. Parser Implementation

The parsing approach depends upon the hypothesis that the intermediate level of representation captures a large number of significant linguistic generalizations that are manifested by a large number of multiple (often redundant) low level cues: phonotactics, allophonics, suprasegmentals (e.g., duration, pitch, intensity), etc. This observation could be viewed as a form of a ‘‘constituency hypothesis.’’ Just as syntacticians have argued for the constituent-hood of noun phrases, verb phrases and sentences on the grounds that these units seem to capture crucial linguistic generalizations (e.g., question formation, wh-movement), so too, I might argue (along with certain phonologists such as Kahn [Kahn]) that syllables, onsets, and rhymes are constituents because they also capture important allophonic and phonotactic generalizations. If this constituency hypothesis for phonology is correct, then it seems natural to propose a syllable parser for processing speech, by analogy with sentence parsers that have become standard practice in the natural language community for processing text.

A program has been implemented [Church] which parses a lattice of phonetic segments into a lattice of syllables and other phonological constituents. In many ways it is very much like a standard chart parser. Let me provide a quick review of context-free chart parsing. Recall that a conext-free parser takes an input sentence such as *They are flying planes* and a grammar such as

N → they	V → are	N → flying
A → flying	V → flying	N → planes
S → NP VP	VP → V NP	VP → V VP
NP → N NP	NP → AP NP	NP → VP
AP → A		

and produces as output a set of parse trees such as: [Note to the editor: perhaps you might typeset these

trees graphically if you can.]

- [S [NP [N They]] [VP [V are] [NP [AP [A flying]] [NP [N planes]]]]]
- [S [NP [N They]] [VP [V are] [NP [VP [V flying] [NP [N planes]]]]]]
- [S [NP [N They]] [VP [V are] [VP [V flying] [NP [N planes]]]]]

For efficiency purposes, the set of parse trees are usually represented as a *chart* of *labeled phrases*. A labeled phrase is a sequence of words from the input sentence labeled with a part of speech, e.g., [NP *flying planes*]. Formally, a labeled phrase is a triple <i, j, c> where i and j indicate where the phrase starts and ends and c is a category label (part of speech). Thus, [NP *flying planes*] would be represented as the triple <2, 4, NP>, indicating that the phrase spans from position 2 to position 4 and is labeled NP. Positions are numbered as following:

0 They 1 are 2 flying 3 planes 4

A chart is simply set of triples. For the sentence *They are flying planes* and the above grammar, the parser would compute the following triples:

<0, 1, they>
<1, 2, are>
<2, 3, flying>
<3, 4, planes>

<0, 1, N>
<1, 2, V>
<2, 3, N>
<2, 3, V>
<2, 3, A>
<3, 4, N>

<0, 1, NP>
<1, 2, VP>
<2, 3, NP>
<2, 3, VP>
<2, 3, AP>
<3, 4, NP>

<1, 3, VP>
<1, 4, VP>
<2, 4, VP>
<2, 4, NP>

<0, 2, S>
<0, 3, S>
<0, 4, S>

Various shorthand notations have been adopted for representing these triples. In the AI literature (e.g., [Kay] and [Kaplan]), the chart is often presented as a graph where each position is represented as a node and each triple <i, j, c> is represented as a labeled arc spanning from node i to node j with the labeled c. I prefer the to present the chart as an i by j table:

	0	1	2	3	4

0	{}	{NP,N,they}	{S}	{S}	{S}
1	{}	{}	{VP,V,are}	{VP}	{VP}
2	{}	{}	{}	{NP,VP,AP,N,V,A,flying}	{NP,VP}
3	{}	{}	{}	{}	{NP,N,planes}
4	{}	{}	{}	{}	{}

where phrases with common i and j are factored together. This tabular notation is popular in the algorithms literature (e.g., [Aho, p. 316]).

There are many well-known parsing algorithms that produce a chart like that above in time $O(n^3)$ (proportional to the cube of the number of words). One such algorithm is outlined below. It finds a phrase between positions i and j by picking a position k in between and testing whether there are two phrases, one form i up to k and another from k to j, that can combine according to the grammar. For example, the algorithm will determine the triple <0, 4, S> because there is an NP from 0 to 1 and there is a VP from 1 to 4 and the two phrases can combine according to the grammar rule S → NP VP. The general entry in the chart is:*

$$chart(i,j) = \bigcup_{i<k<j} chart(i,k) \cdot chart(k,j)$$

where multiplication combines sets of phrases according to the rules of the grammar (e.g., {NP} · {VP} = {S}) and union combines alternative sets of analyses (e.g., {NP} $\cup$ {VP} = {NP, VP}).

for j := 1 to n do
    chart(j-1, j) := {A | A derives the j^{th} word}
    for i := j-2 downto 0 do
        $chart(i,j) := \bigcup_{i<k<j} chart(i,k) \cdot chart(k,j)$

This algorithm can be performed in $O(n^3)$ time by choosing all combinations of i, j and k, each of which have n possible values. (The multiplication step requires constant time, independent of the actual input words. It depends only on the grammar and hence it is often known as the *grammar constant*.)

6. Chart Parsing of Phonetic Transcriptions

The same chart parsing methods can be used to find syllable structure from an input transcription. Suppose that I was given an ideal error-free transcription [t|tam] of the phrase ''to Tom'' and I was asked to recover the syllable structure. Recall that a parser takes a grammar and an input sentence. In this case, let's assume a highly oversimplified grammar:

onset → [t] | [m]	peak → [|] | [a]
coda → [t] | [m]	syl → (onset) peak (coda)

The input sentence is the transcription (shown here with position numbers for the reader's convenience):

Input Sentence: 0 [t] 1 [|] 2 [t] 3 [a] 4 [m] 5 (= to Tom)

Using the algorithm outlined above, the resulting chart is:

---

* This formulation assumes the phrases have one or two daughters, or more formally, that the grammar is in Chomsky Normal Form. Let me not complicate the issues here by showing how this restriction can be relaxed.

	0	1	2	3	4	5
0	{}	{[t],onset,coda}	{syl}	{syl}	{}	{}
1	{}	{}	{[|],peak,syl}	{syl}	{}	{}
2	{}	{}	{}	{[t],onset,coda}	{syl}	{syl}
3	{}	{}	{}	{}	{[a],peak,syl}	{syl}
4	{}	{}	{}	{}	{}	{[m],onset,coda}
5	{}	{}	{}	{}	{}	{}

From this chart, it is possible to determine that the input sentence is ambiguous and that there are two alternative parses: [t|][tam] and [t|t][am]. But as suggested above, the parser ought to be able to rule out the second analysis by exploiting allophonic constraints. Recall that I argued that there must be a syllable boundary before the aspirated /t/ in ‘‘Tom’’ because (at least as a first approximation in American English), aspirated /t/s are constrained to be in syllable initial position. This reasoning can be captured within the parsing framework by refining the grammar to express the aspiration constraint discussed above. The rules below account for the aspiration constraint by permitting aspiration in syllable initial position (under the onset node), but not in syllable final position (under the coda). And consequently, with this refined grammar, the parser no longer finds the spurious syllables from 0 to 3 and from 1 to 3.

Input Sentence: 0 [tH] 1 [|] 2 [tH] 3 [a] 4 [m] 5 (= to Tom)

Grammar:

syllable → (onset) peak (coda)
onset → tH | kH | pH | ...
coda → t> | k> | p> | ...

Chart:

	0	1	2	3	4	5
0	{}	{[tH],onset,coda}	{syl}	{}	{}	{}
1	{}	{}	{[|],peak,syl}	{}	{}	{}
2	{}	{}	{}	{[tH],onset,coda}	{syl}	{syl}
3	{}	{}	{}	{}	{[a],peak,syl}	{syl}
4	{}	{}	{}	{}	{}	{[m],onset,coda}
5	{}	{}	{}	{}	{}	{}

Stating allophonic and phonotactic processes in this way allows the speech recognition device to exploit allophonic constraints on syllable structure and stress with very powerful and well-understood parsing techniques.

7. Matching

Once the syllable structure has been established, the structure is then matched against the dictionary to find word hypotheses. Matching ought to be a subject of future research; I was primarily concerned in this work with the parser, and simply implemented enough of a matcher so that I could demonstrate a working system which inputs linguistic transcriptions and outputs lexical entries.

7.1 Dictionary Lookup

The dictionary is stored in two levels along the lines suggested in [Smith]; the first indexes words in terms of syllables, and the second indexes syllables in terms of its sub-constituents (sylparts). The matcher looks up every sylpart in the syllable dictionary, and then every syllable in the word dictionary. I should have added a third level of indexing at the level of foot structure. Thus, syllables would be indexed in terms of feet, and feet in terms of words. This was not implemented.

> ‘‘Noah uses two levels between the input level of segment-label hypotheses and the output level of word hypotheses, where POMOW [an earlier system based on Markov processes] uses one. There are 1) the *sylpart* level, consisting of parts of syllables -- onsets (the initial non-nucleus part of syllables), vowels, and codas (the final non-nucleus part of syllables) -- and 2) the syllable level, consisting of complete syllables (not syltypes as found in POMOW). Knowledge is stored in a *hierarchy-tree* representation. That is, between each pair of adjacent levels (segment-sylpart, sylpart-sylpart, and syllable-word level pairs) is a tree structure storing a sequence of lower level units to define a higher level unit. The last node of the sequence of lower level units points to the defined higher level unit. For example, the syllable-word tree stores sequences of syllables defining each word in the vocabulary. The tree between each pair of levels permits merging common initial parts of sequences to reduce storage costs and recognition time. Thus, the words ‘‘confide’’ and ‘‘confuse’’ share the first syllable node ‘‘con’’, in the tree, which then points to subnodes ‘‘fide’’, ‘‘fuse’’, etc.’’ [Smith, p. 13]

Smith achieved two bottom-line engineering advantages by adopting a hierarchical representation: both storage cost and recognition time are reduced due to sharing. For example, the tree stores the syllable ‘‘con’’ just once, not once for each word containing the prefix ‘‘con’’ (e.g., ‘‘confide’’, ‘‘confuse’’, and so forth). Furthermore, the hierarchical system will recognize the prefix ‘‘con’’ just once, not once for each word containing the prefix. Smith argues that his representation will reduce recognition costs by 41% over the HWIM design for the same 1000 word dictionary, at a 22% increase in storage due to pointer overhead [Smith, p. 40]. In addition, Smith found that savings increase approximately logarithmically with larger lexicons, and that accuracy improves with larger lexicons.

Smith rested most of his argument for hierarchical representation on the force of these engineering advantages. There are, though, some equally important linguistic advantages. A hierarchical representation allows the lexicon to be organized along lines that reflect natural linguistic generalizations. Thus, for example, the prefix /ri-/ ought to be shared in words like ‘‘retry’’ and ‘‘recycle’’ where it is a linguistically motivated constituent, but it should not be shared in words like ‘‘read’’ and ‘‘real’’ where /ri/ is merely a common subsequence of segments. In this respect, Smith’s proposal fairs favorably to the Cohort model which would lump all words beginning with /ri/ into a single cohort, whether /ri/ forms a constituent or not.

If Smith had employed a phonetic parser, he might also have found that the /ri/ constituent will be realized with different allophones from the /ri/ which is not a constituent, and these differences could (often) be detected with the phonetic parser. Thus, representing the lexicon along linguistically motivated lines has the additional advantage that the matching process can take advantage of the constituent structure provided to it by the parsing process.

7.2 Canonicalization

Before constituents are looked up in the dictionary, a canonicalization process is applied which undoes postlexical (allophonic) rules. Canonicalization strips a phonetic segment of its so-called ‘‘non-distinctive’’ features (e.g., aspiration, flapping, glottalization) and leaves the segment with just its ‘‘distinctive’’ features (e.g., place, voice, manner). For example, canonicalization replaces both aspirated and glottalized variants of /t/ with the phoneme /t/. In some cases, canonicalization introduces nondeterminism by mapping a single allophone into two or more phonemes. Flaps, for instance, are mapped into both /t/ and /d/ (and also into /n/ in certain contexts). Canonicalization removes all variant phonetic detail, but does not sacrifice information, I hypothesize, because the information content of the allophonic constraints has already been milked by the parser. Thus, whatever information was encoded in the distribution of allophones has already been recoded into the syllable (and metrical) constituent structures.

In suggesting the canonicalization process, I am assuming along with linguists such as Kiparsky that the lexicon is represented solely in terms of phonological features and does not contain allophones. Kiparsky and other advocates of lexical phonology assume that allophones are generated solely by postlexical rules such as flapping and aspiration and that these rules apply outside the lexicon. Lexical phonology distinguishes postlexical rules from lexical rules such as trisyllabic laxing (e.g., *divIne* → *divInity*) and /t/ → /s/ (e.g., *president* → *presidency*) which do not generate allophones. Advocates of lexical phonology point out that lexical rules must apply within the lexicon because they are subject to lexically marked exceptions and because they are sensitive to the internal morphological structure of words. In contrast,

postlexical rules share neither of these properties; they apply without lexical exceptions and without regard for the internal morphological structure of words. Moreover, the fact that postlexical rules cross word boundaries (unlike lexical rules) suggests that postlexical rules do not operate on words, but rather on some other level of representation. For these reasons, I accept Kiparsky's conclusion that the lexicon is free of allophones and reject the hypothesis that allophonic rules are compiled into the lexicon, an alternative that was popular during the ARPA speech project (e.g., [Oshika et al]).

8. Conclusions

I have argued that lexical retrieval should be decomposed into two subproblems: parsing and matching. This model elegantly makes use of both contextually varying cues as well as contextually invariant cues. The parser is designed to take advantage of features that vary systematically with context, whereas the matcher performs optimally with invariant cues. This is no accident; it is hard to imagine how invariant cues could be used to parse the context or how variant cues could be used to match constituents.

Most previous models (e.g., the Cohort Model, Harpy) were unable to make much use of variant cues because they lacked a mechanism such as a parser for reasoning about context. These pure matching models were forced to depend entirely on invariant features because they had no principled way to take advantage of the rich contextual constraints imposed by allophonic variation and phonotactics.

9. Acknowledgements

I have received a considerable amount of help and support over the course of this project. Let me mention just a few of the people that I should thank: Jon Allen, Glenn Burke, Francine Chen, Scott Cyphers, Sarah Ferguson, Margaret Fleck, Osamu Fujimura, Dan Huttenlocher, Jay Keyser, Lori Lamel, Mark Liberman, Mitch Marcus, Ramesh Patil, Janet Pierrehumbert, Dave Shipman, Pete Szolovits, Meg Withgott and Victor Zue.

REFERENCES

[1] Aho, A., ''The Theory of Parsing, Translations, and Compiling,'' volume I, Prentice-Hall, Englewood Cliffs, N. J., 1972.

[2] Chomsky, N. and Halle, M., ''The Sound Pattern of English,'' Harper & Row, 1968.

[3] Christie, W., ''Some Cues for Syllable Juncture Perception in English,'' the Journal of the Acoustical Society of America, pp. 819-821, 55:4, 1974.

[4] Church, K., ''Phrase-Structure Parsing: A Method for Taking Advantage of Allophonic Constraints,'' unpublished doctoral dissertation, department of Electrical Engineering and Computer Science, MIT, 1983 (also available from Indiana University Linguistics Club).

[5] Cole, R., and Jakimik, J., ''A Model of Speech Perception,'' in R. Lawrence Erlbaum, Hillsdale, N.J., 1980.

[6] Earley, J., ''An Efficient Context-Free Parsing Algorithm,'' CACM, 13:2, February, 1970.

[7] Fry, D., ''Duration and Intensity as Physical Correlates of Linguistic Stress,'' JASA 17:4, 1955, (reprinted in Lehiste (ed.), ''Readings in Acoustic Phonetics,'' MIT Press, 1967.)

[8] Fujimura, O., and Lovins, J., ''Syllables as Concatenative Phonetic Units,'' Indiana University Linguistics Club, 1982.

[9] Jelinek, F., course notes, MIT, 1982.

[10] Kahn, D., ''Syllable-Based Generalizations in English Phonology,'' Indiana University Linguistics Club, 1976.

[11] Kay, M., ''The MIND System,'' in R. Rustin, ''Natural Language Processing,'' Algorithmics Press, New York, 1973.

[12] Kaplan, R., ''A General Syntactic Processor,'' in R. Rustin, ''Natural Language Processing,'' Algorithmics Press, New York, 1973.

[13] Kiparsky, P., ‘‘Remarks on the Metrical Structure of the Syllable,’’ in W. Dressler (ed.) Phonologica 1980. Proceedings of the Fourth International Phonology Meeting, 1981.

[14] Klatt, D., ‘‘Word Verification in a Speech Understanding System,’’ in R. Reddy (ed.), ‘‘Speech Recognition,’’ Invited Papers Presented at the 1974 IEEE Symposium, Academic Press, pp. 321-344, 1974.

[15] Klatt, D., ‘‘Review of the ARPA Speech Understanding Project,’’ JASA, 62:6, December 1977.

[16] Klatt, D., ‘‘Scriber and Lafs: Two New Approaches to Speech Analysis,’’ chapter 25 in W. Lea, ‘‘Trends in Speech Recognition,’’ Prentice-Hall, 1980.

[17] Nakatani, L., ‘‘Fission and Fusion Processes in Speech Perception,’’ unpublished ms., Bell Laboratories, Murray Hill, New Jersey.

[18] Oshika, B., Zue, V., Weeks, R., Neu, H., and Aurbach, J., ‘‘The Role of Phonological Rules in Speech Understanding Research,’’ IEEE Trans. Acoust. Speech and Signal Processing, Vol. AASP-23, 1, pp. 104-112.

[19] Smith, A., ‘‘Word Hypothesization in the Hearsay-II Speech System,’’ Proc. IEEE Int. Conf. ASSP, pp. 549-552, 1976.