The PRONALSYL Letter-to-Phoneme Challenge

Bob Damper* and Yannick Marchand†
*University of Southampton, UK
†Institute for Biodiagnostics (Atlantic), Canada

PASCAL Workshop, Venice, Italy
11 April 2006

Structure

Divided into three parts:

1. The problem of letter-to-sound conversion (20 min; Bob Damper)

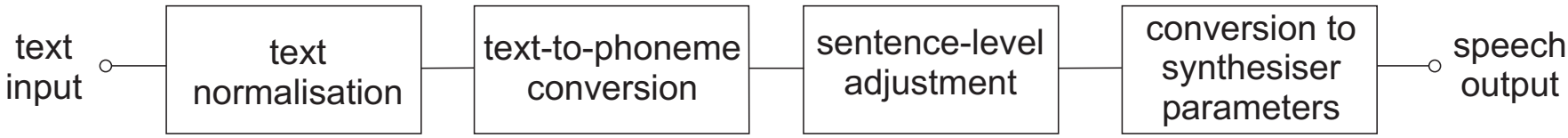
2. The PRONALSYL Challenge (20 min; Yannick Marchand)

3. Discussion of issues (20 min, all)

Goals for Part 1

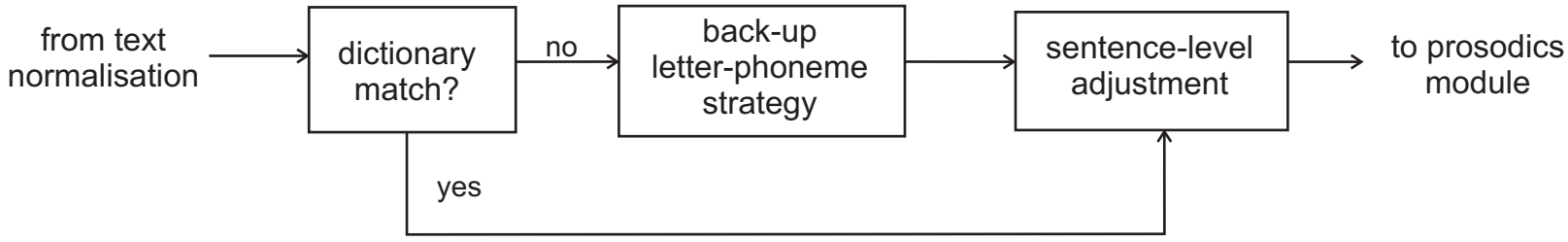
- Scene-setting: Introduce and motivate the problem of letter-to-phoneme conversion.
- Convince you that it is both *hard* and *important*, e.g., in speech technologies like text-to-speech (TTS) synthesis.
- Outline history of approaches to solution, both:
  1. traditional, based on experts' rules; and
  2. data-driven, based on learning from example pronunciations.
- Encourage participation.

Basic Scheme for TTS Synthesis



- The purpose of *text normalisation* is to convert logographs (e.g., &), abbreviations (Mr., Mrs., etc.), numerals and so on to normal text.
- We then convert the text into some intermediate description—almost always **phonemic**—more closely related to the sound system of the language being synthesised (*linguistic mapping*).
- Sentence-level adjustment is concerned with issues such as word and sentence stress, and vowel reduction as a result of sentential context (not always treated as a separate process).
- The final stage produces parameters to drive the speech synthesis hardware—the *parametric mapping*.

Letter-Phoneme Conversion



- Dictionary look-up is the method of choice.
- It is simply not possible to list *all* the words of a language, because language is highly generative; new words are being created all the time.
- So we must have a back-up strategy, i.e., a way of transcribing ‘unknown’ words not in the dictionary (aka ‘lexicon’).
- The problem of deriving a pronunciation automatically from its spelling turns out to be extraordinarily hard for English.

What's a Phoneme?

- Phonemes are abstract units of sound, defined by their ability to distinguish between 'words' (lexemes) such as <pit> and <bit>, which are minimally distinctive.
- Letter-to-phoneme (L2P) conversion is sometimes (actually, more frequently) called *grapheme*-to-phoneme conversion.
- So what's a grapheme? It is a group of letters which is pronounced as a single phoneme; e.g., <ough> $\rightarrow$ /ɔ/ as in <ought>, <ph> $\rightarrow$ /f/ as in <phase>.
- I don't like this term as *grapheme* has at least one other meaning (namely, an abstract unit of the writing system, similar to the concept of 'phoneme').

Why is L2P Conversion so Hard?

- We use 26 letters in English orthography yet about 45-50 phonemes in specifying pronunciation $\Rightarrow$ PROBLEMS!
- For instance, the letter <c> is pronounced /s/ in <cider> but /k/ in <cat>. Yet, the /k/ sound of <kitten> is written with a letter <k>.
- The combination <ough> is pronounced /ɔ/ in <bought> but /ʌf/ in <enough>.
- Usually, there are fewer phonemes than letters but there are exceptions, e.g. (<six>, /sɪks/).
- English has non-contiguous *markings* as when letter <e> is added to (<mad>, /mad/) to make (<made>, /meɪd/). The final <e> is not sounded, but indicates that the vowel is lengthened or dipthongised.

Rule-Based Conversion

- Given these problems, how is it possible to perform automatic translation of text to phonemes at all?
- It is generally believed that the problem is largely soluble provided sufficient *context* is available.
- The traditional back-up strategy employs a set of phonological or context-dependent translation (CDT) rules written by an expert.
- The form of the rules (Chomsky and Halle, 1968) is:

$$A[B]C \rightarrow D$$

  i.e, the letter substring B with left-context A and right-context C receives the pronunciation (i.e. phoneme substring) D.
- Note that there is no numerical indication of rule 'probability' or 'certainty'.

Applying CDT Rules

- Rules typically applied left-to-right, starting with first letter of word.
- More than one rule generally applies at each stage of transcription.
- Conflicts are resolved by maintaining the rules in a set of sublists, grouped by (initial) letter and with each sublist ordered by specificity.
- Typically, the most specific rule is at the top and most general (a default) at the bottom.
- For the particular target letter (i.e., initial letter of the B substring), the appropriate sublist is searched from top-to-bottom until a match is found.
- Matching rule is then fired (i.e., corresponding D substring is right-concatenated to the evolving output string), the linear search terminated, and the next untranscribed letter taken as target.

CDT Rules . . . Continued

- Typical rules might be:

$$\begin{aligned} R_i: &\quad C[\mathrm{o\,o}]C &\rightarrow &\quad /\mathrm{u\textrm{ː}}/ \\ R_k: &\quad \#[\mathrm{o\,f}]\# &\rightarrow &\quad /\mathrm{\upsilon v}/ \end{aligned}$$

- Rule R_i states that <oo> preceded and followed by a consonant is pronounced /uː/ as in <root> or <food>.
- (But note that the vowel of <good> would receive a wrong pronunciation.)
- Rule R_k states that the word <of> is pronounced /ʊv/ (here, # is a symbol for word-delimiting space).
- Well known rule sets are those of Ainsworth (1973), Elovitz et al. (1976) and Divay and Vitale (1997).

Machine Learning Approaches

- The task of manually writing a set of CDT rules is very considerable and requires an expert depth of knowledge of the specific language.
- The expert has to decide on the particular rules, how many rules are sufficient, rule order so as to resolve conflicts appropriately, how to test for completeness, what to do as mispronunciations are discovered during rule development etc.
- These problems can be avoided by using automatic, machine learning techniques based on extracting spelling-to-sound regularities from large sets of example data.
- How do such “data-driven” techniques compare to traditional rules?

Some Quotes

"The performance *[of NETtalk]* is not nearly as accurate as that of a good set of letter-to-sound rules" (Klatt 1987)

"To our knowledge, learning algorithms, although promising, have not yet reached the level of rule sets developed by humans" (Divay and Vitale 1997)

"...such training-based strategies are often assumed to exhibit much more intelligence than they do in practice, as revealed by their poor transcription scores" (Dutoit 1997)

Unfortunately, these quotes are simply and straightforwardly **WRONG**. Why did no one notice this before???!!!

Comparing L2P Methods

Elovitz rules	25.7% out of 16,280 words
NETspeak	46.0% out of 8,140 unseen 54.4% out of 16,280 seen
Exemplar-based	57.4% out of 8,140 unseen
Analogy	71.8% out of 16,280 unseen

(from Damper et al. 1999)

- NETspeak is a feed-forward neural network (McCulloch et al., 1987)—a variant of NETtalk (Sejnowski and Rosenberg, 1987)—trained on error back propagation.
- In all cases except the rules (where the process is unnecessary), letters and phonemes were pre-aligned manually (see next slide).
- Note the very poor performance of the manually-written rules! Are rules always this bad?

Letter-Phoneme Alignment

- ML approaches generally require letters and phonemes to have been aligned as a bijection . . . to convert the problem of translation between two alphabets to a problem of classification.
- A ubiquitous process in speech and language processing.
- A possible alignment for the word (<make>, /meɪk/) is:

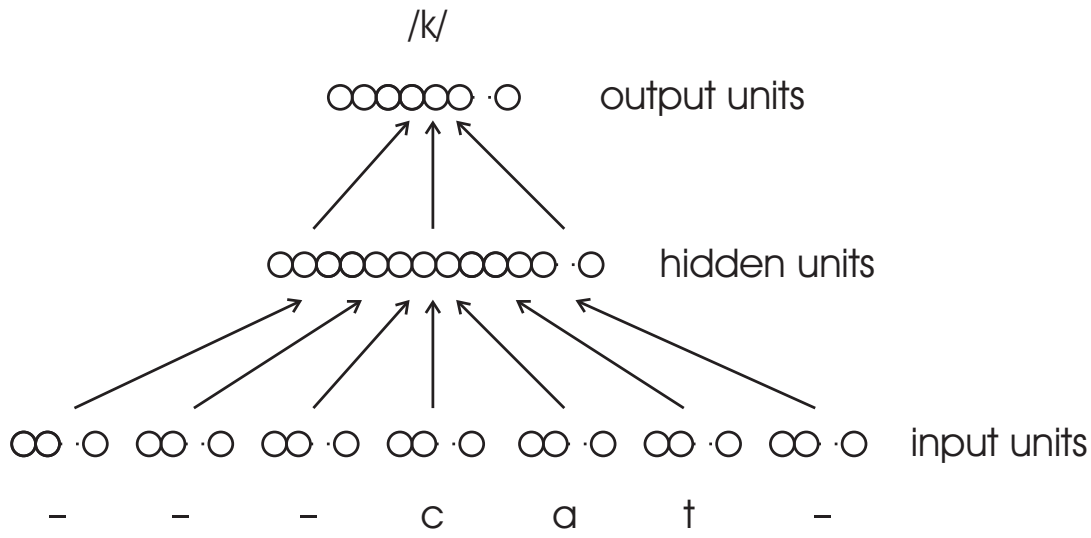
m	a	k	e
|	|	|	|
m	eɪ	k	–

- Note addition of a null phoneme '–' to make the number of letters and phonemes equal. Sometimes we need null letters!

Automatic Alignment

- Another hard problem!
- There is no real theoretical basis for alignment, hence no *gold standard* to use in a supervised learning approach or in evaluation of the result.
- Suppose we have a matrix $\mathbf{A}_k$ of letter-phoneme associations . . .
- Given $\mathbf{A}_k$, we can use dynamic programming to align all word spellings with their pronunciations.
- $\mathbf{A}_k$ can then be iteratively improved to $\mathbf{A}_{k+1}$ using the EM algorithm (Damper et al. 2005).
- Different initialisations ($\mathbf{A}_0$) possible . . . we generally use naïve initialisation.
- We are making our alignment algorithm available on the PRONALSYL website.

NETtalk



- A ‘window’ of 7 letters is fed to the (203) input units then via the hidden units (120) to the 26 output units.
- The central letter in the window is transformed into its equivalent phoneme.
- Here, the initial letter <c> of <cat> receives the pronunciation /k/.
- (Note the requirement for one-to-one alignment of training data.)

Input Coding

- The central letter is the ‘target’ and the letters to left and right provide the context.
- Each letter is binary coded as a 1-out-of-29 code (26 letters plus 3 special characters), so there are $i = 29$ input units.
- The input text is stepped through the window letter-by-letter.
- In the case where the orthographic representation is longer than the phonemic one, S&R added null phonemes to maintain a one-to-one alignment.
- In the (rarer) case when the phoneme string is the longer of the two, S&R invented new ‘phonemes’ (e.g., /K/ to correspond to the letter $<$x$>$ $\rightarrow$ /ks/ as in $<$sexual$>$).
- Alignment was done manually by Rosenberg.

Output Coding

- NETtalk uses 21 “articulatory features” to represent the (single) phoneme output, e.g., voicing, place of articulation (labial, velar, . . .) and tongue height.
- 5 additional output units represent stress and syllable boundaries. Hence, the number of output units o is $21 + 5 = 26$.
- Outputs obtained in actual use are continuously graded in the range [0, 1] but can be thresholded to give ‘hard’ values $\in \{0, 1\}$.
- Even so, very few correspond exactly to legal codings, since 2^{26} massively exceeds the cardinality of the phoneme set.
- Solution is to choose the “best guess” phoneme whose code has the smallest vector angle with the output code.

Hidden Units

- S&R studied various numbers of hidden layers and units (h).
- Performance increased with additional units and with an additional hidden layer.
- Most comprehensive results are presented for a single hidden layer of 120 units. So the number of connections (excluding variable thresholds) is:

$$(i \times h) + (h \times o) = (203 \times 120) + (120 \times 26) = 27{,}480$$

- To this must be added the small number ($h + o = 146$) of variable thresholds, so total number of adjustable parameters is approximately 28,000.
- (NETspeak uses a significantly smaller network of approximately 8,000 weights.)

Performance of NETtalk

- S&R trained and tested NETtalk on (transcribed) words in *Merriam-Webster's Pocket Dictionary* of American English.
- The 1,000 most common words were selected from the total of 20,012 in the dictionary. This was used as BOTH the training set and test set (a mistake!)
- Performance was asymptotic to 98% "best guess" phonemes correct after some tens of passes through the training set.
- Unfortunately, S&R never tested the generalisation ability of NETtalk on an entirely unseen dataset.
- Training to asymptote on the 1,000-word corpus and testing on the full 20,012-word dictionary, the result was 77% phonemes correct.
- But 77% *phonemes* correct is awful! (It's probably about 20-25% words correct.)

Discussion of NETtalk

NETtalk was highly influential in showing that data-driven techniques could be applied to the large, difficult problem of text-phoneme conversion with a degree of success BUT . . .

- The training data were manually prealigned, so simplifying the learning problem significantly.
- The generalisation power of NETtalk was never properly assessed on a totally unseen test set.
- Scoring was in terms of phonemes correct only, which gives an inflated and insensitive view of performance.
- As we have seen, approach does not work as well as exemplar-based or analogy . . .
- . . . and we think we know why (Daelemans et al. 1999, “Forgetting exceptions is harmful in language learning”).

Current State-of-the-Art

- Amazingly, manually-written expert rules remain the approach of choice in speech technology!
- This is a triumph of:
  1. tradition over evidence
  2. the reluctance of mainstream linguists to recognise the weakness of their discipline
  3. failure to read up to date literature
  4. the persistence of erroneous literature (claiming that automatic pronunciation is a solved problem, solved by rules)
- I get approximately 10 speech synthesis papers a year to review in which the authors have used rules and about 1 per decade in which they haven't.

What ML Method?

- An open problem . . . many candidates to be compared:
  - Neural nets
  - Decision trees
  - Hidden Markov models
  - Finite state transducers
  - Bayesian classifiers
  - kNN, instance-based, similarity-based, generalised table look-up
  - Analogy, latent analogy
  - SVMs
  - Logical rule-extraction
  - etc.
- Also, fusion of > 1 of the above.

On to Part 2

- Bob has introduced and motivated the problem.
- (Actually, several problems, not one!)
- Now let's get more specific to Pronalsyl.
- Goals for Part 2 are:
  - Housekeeping: organising committee, timetable, etc.
  - Introduce issues which need to be considered:
    - multilinguality, dictionaries
    - dictionary as a 'gold standard'
    - evaluation

    etc.

What's Pronalsyl?

- Stands for PRONunciation, ALignment, SYLlabification and anything else deemed relevant.
- As just discussed, the basic problem is *automatic PRONunciation* of text.
- Assuming an alphabetic writing system as for English, this can be viewed as a translation between parallel strings, or 'texts', of letters and phonemes.
- But this is not just a matter of one-to-one transliteration of symbols! So we (may) need ALignment . . .
- . . . the process of forcing one-to-one correspondence.
- Syllabification can also improve automatic pronunciation (Marchand and Damper, forthcoming). Pronunciation of a letter seems to depend upon where it is in the syllable. Why?

Organising Committee

Antal van den Bosch (Tilburg University, The Netherlands)

Stanley Chen (IBM T. J. Watson Research Center, USA)

Walter Daelemans (University of Antwerp, Belgium)

Bob Damper (University of Southampton, UK, Co-Chair)

Kjell Gustafson (Acapela Group and KTH, Sweden)

Yannick Marchand (National Research Council Canada, Co-Chair)

François Yvon (ENST, France)

Timetable

- Organising Committee in place and active.
- Challenge advertised on PASCAL and ELSNET at beginning of March 2006.
- Connectionist-list declined to advertise the Challenge as “off topic”!!!
- Web site is “semi-live” . . . some problems caused by THE FIRE and by some technical concerns.
- Challenge (provisionally) closes November 2006.
- We hope to organise a session at NIPS in December 2006.
- Journal special issue planned.

Issues

- Multilinguality:
  - The L2P problem looks different across different languages with different writing and phonological systems.
  - Are there really *hard* and *easy* languages?
  - If so, how can we quantify this? (entropy, perplexity, $|\Sigma_l|$ vs. $|\Sigma_p|$, . . .)
  - Different methods for different languages?
- Handling inconsistencies and multiple pronunciations.
- Using linguistic knowledge . . . Suppose, for example, part of speech information is available. Does it help? How should it be used?
- Evaluation . . . the BIG one.

Scientific Goals

- Understand better the L2P problem and the way it varies across languages.
- Benchmark and compare a range of approaches and methods . . .
- . . . not just on performance but on aspects like training requirements also (e.g., can we avoid problematic requirement for alignment?)
- All with a view to achieving:
  - Better capabilities in L2P . . .
  - Better capabilities in ML . . .
- Understand what makes a particular approach a good/bad one for this problem.

Evaluation

- A difficult one because of the different training needs of different approaches (e.g., neural networks vs. kNN, 'eager' vs. 'lazy' learning) . . .
- . . . and because some participants will use aligned data and/or syllabification, some will not.
- Majority opinion of Organising Committee is to impose 10-fold cross validation, with the 10 'folds' supplied.
- Train on each of the 10 folds, testing on the remaining 9; report mean word accuracy and standard error.
- Participants could still optimise parameters by repeated cross validations, which is 'unfair' but difficult to police.
- Decision on this influenced what we made available on the web site $\Rightarrow$ SOME UNAVOIDABLE DELAY.

Web Site and Dictionaries

- Web site will contain:
  - Wide range (we hope!) of multilingual resources
  - Some simple demos. of representative methods: analogy is there, DEC or naïve(ish) Bayes is planned
  - Our alignment algorithm
  - Possibly, script to divide into 10 folds
- At present, dictionaries for:
  - American English (Merriam Webster)
  - British English (BEEP)
  - French (Brulex)
  - Italian (Festival)
- These are aligned but undivided (into folds).
- They need to be divided and new resources will be added soon.

On to Part 3

- DISCUSSION