Enhancing Phonological Representations for Multilingual Speech Technology

Stephen Wilson, Julie Carson-Berndsen and Michael Walsh
Department of Computer Science, University College Dublin, Ireland
E-mail: {stephen.m.wilson, julie.berndsen, michael.j.walsh}@ucd.ie

ABSTRACT

This paper is concerned with the enhancement of phonological representations for the purposes of multilingual ubiquitous speech technology applications. Such applications would be multilingual, in the sense that the techniques they employ for acquiring and representing phonological information are generic in form and can therefore be used with any language, and ubiquitous meaning that the framework in which they are developed and applied is portable and readily accessible anywhere. Three XML based mechanisms are described with a view to facilitating the goal of multilingual ubiquitous speech technology: *Multilingual time maps,* which integrate linguistic data storage and finite state technology with XML's portability; *LeXMLicon,* an XML based syllable lexicon generator; and *PROMPT*, a phonological resource organiser for mapping data from one format to another.

1. INTRODUCTION

Ubiquitous language technology concerns the development of language technologies for different purposes on different platforms so that they can be made available to everybody at all times rather than to a select group for specific purposes. To this end, three integrated representation mechanisms are presented in this paper. The first, termed a *multilingual time map*, is defined in XML and interpreted as a multi-level finite state transducer [2]. It is based on the model of *Time Map Phonology* [3] and extends the notion of a *phonotactic automaton* as defined in [8].

The second mechanism, *LeXMLicon*, is a generic tool for the rapid design and creation of syllable lexicons. It builds on the idea of a generic lexicon model [4,7] and widens the scope of the system previously presented in [12].

The third mechanism presented is a Portable Resource Organiser for Managing Phonological Transformations (*PROMPT*). It allows users to define and store specific phoneme to feature attribute mappings as *feature profiles*. These profiles are then associated with phonemic representations in a variety of phonetic notations. *PROMPT* allows for both the generation of input data for either *multilingual time maps* or *LeXMLicon* as well as mapping data output from both mechanisms into different phonetic alphabets or phonological feature sets.

2. MULTILINGUAL TIME MAPS

The *Time Map* model [3], the underlying model upon which the concept of *multilingual time maps* is based, builds on the autosegmental approach to phonology [10] by allowing multilinear representations of autonomous features to be interpreted by an event-based computational linguistic model. It employs a finite-state representation of the permissible combinations of sounds in a language, a *phonotactic automaton*, along with axioms of event logic to interpret these multilinear representations. Although input to the model is in absolute signal time, in milliseconds, parsing takes place in the relative time domain, i.e. the temporal relation of features to one another, whether they overlap or precede. The multilinear representation of an utterance is gradually "windowed" through, and each window is examined for feature overlap constraints. These constraints are imposed top-down by the *phonotactic automaton*. On successful satisfaction of these constraints, a phoneme segment is recognized, a transition in the automaton is traversed and the window is moved on. A more detailed explanation of the parsing process can be found in [8].

An example of a *phonotactic automaton* as used in the *Time Map* speech recogniser is depicting the CC-onsets in English syllables is given in figure 1. Each arc of the phonotactic automaton depicted constraints on overlap relations between features which must be satisfied in a particular phonotactic context; the average duration of the sounds in this context is also given.

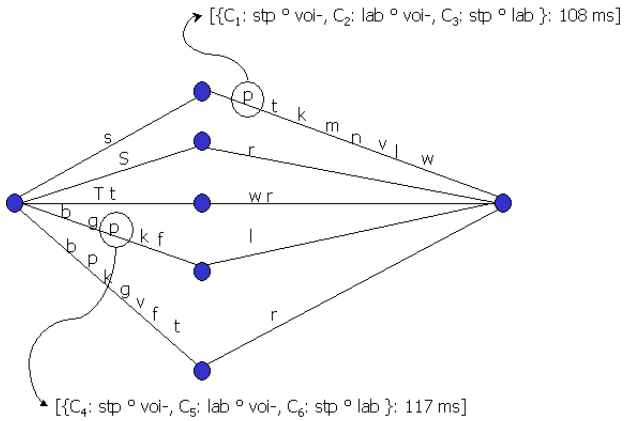

Figure 1: Phonotactic automaton for English CC-onsets

A *multilingual time map* takes this representation significantly further by integrating language-specific information of varying levels of granularity within a common structural context. Interpreted as a multilevel finite state transducer represented in XML, it can be

ISBN 1-876346-48-5

viewed as an extension of the *phonotactic automaton* which includes (at least) the following levels (or tapes): graphemes, phonemes, allophones, features, constraints on overlap relations, average duration, frequency and probability. An example of a one arc in a multilingual time map for German CC-onsets is shown in figure 2.

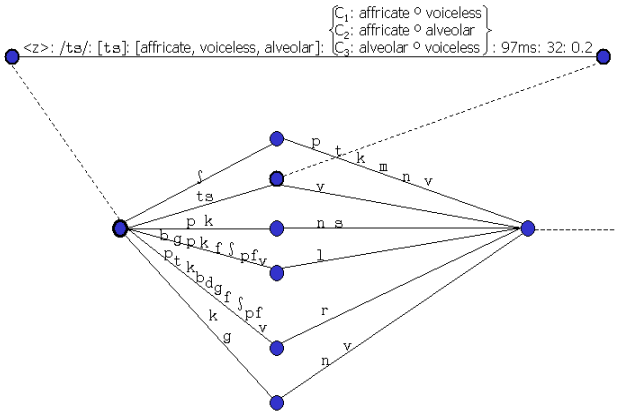

Figure 2: One arc of *mulilingual time map*

The XML representation of this arc of the *multilingual time map* (excluding the overlap constraints) is as follows:

```
<multilingual_timemap>
  <startState>s1</startState>
  <finalState>s2</finalState>
  <transition>
      <sourceState>s1</sourceState>
      <destinationState>s4</destinationState>
      <phonemeTape>ts</phonemeTape>
      <graphemeTape>z</graphemeTape>
      <allophoneTape>ts</allophoneTape>
      <featureTape>
        <featureDescription>voiceless</featureDescription>
        <featureDescription>affricate</featureDescription>
        <featureDescription>alveolar</featureDescription>
    </featureTape>
    ….
    <frequencyTape>32</frequencyTape>
    <durationTape>97ms</durationTape>
    <probabilityTape>0.2</probabilityTape>
  </transition>
</multilingual_timemap>
```

Figure 3: Multilingual time map arc in XML

The use of XML as the interface format guarantees portability and ensures that the information contained within the *multilingual time maps* is readily accessible by other applications [5].

Multilingual time maps offer a means of representing a rich depository of linguistic knowledge within the complete phonotactic context of a language. The generation of additional user-defined transducers from a single *multilingual time map*, through a series of XSLT transformations, broadens the functionality of the model beyond that of speech recognition or synthesis, to include any number of speech applications. An obvious example would be grapheme-phoneme conversion. Given that the representation mechanism provides no constraints on either the number or type of data tape per arc, it is clear that the scope for idiosyncratic user-required applications is large.

The challenge for the portability of the *Time Map* model lies in the efficient acquisition of *multilingual time maps.* We provide for three acquisition strategies: user-driven, data-driven and data-driven with user prompting. *Multilingual time maps* can either be produced manually by a trained linguist or can be learned from a data set either with or without human assistance. This involves the manual specification of separate phoneme and feature inventories by a language expert. *Multilingual time maps* can be learned automatically from a phonemically labeled dataset [6]. Since this initial *multilingual time map* specifies all the forms in the corpus, it automatically includes all the forms that should be included in the lexicon.

3. LeXMLicon

Within the context of ubiquitous language technology one of the main goals was to provide resources that can be used to build applications for use with any language. A primary motivation for this was to allow those languages that have traditionally received little attention from developers of speech applications access to such resources.

Specifically, within the context of such technologies and the lexicon, whereas related research concerned with broadening the scope of the lexicon across languages has focused on generalising over language families using hierarchical inheritance networks [1, 11], our aim was to focus on providing mechanisms for lexical generation. *LeXMLicon* is one such mechanism. It is a generic tool, that is language independent and that can rapidly create a syllable lexicon for any language, provided that language can be adequately represented using some phonetic notation.

LeXMLicon comprehensively describes the phonological features of a given language using DATR, an inheritance based lexical representation language [9]. Syllable templates can be defined manually, or by using the information contained within *multilingual time maps. LeXMLicon* then uses these templates to output an XML lexicon, containing phonological feature information for each segment, as well as information regarding the position of that segment within the syllable. Phonological information is extrapolated using the inference mechanisms of DATR. Such processing is completely hidden and users deal only with the output information in XML. This is accessed through a number of graphical interfaces, or can be readily transformed into a variety of different structures for additional processing. An entry for the syllable [So:n] is shown in figure 4. Each lexicon produced by *LeXMLicon* consists of a number of syllable entries. In the example given, we see that every syllable entry has a child *<lexeme>* which in this case is a SAMPA representation of the entire syllable, [So:n].

```
<syllable>
 <lexeme>So:n</lexeme>
 <onset type="first">
 <segment phonation="voiceless" manner="fricative"
   place="palato" duration="null">S</segment>
 </onset>
 <nucleus type="first">
 <segment phonation="voiced" manner="vowellike"
   place="back" height="mid" roundness="round"
   length="tense" duration="null">o:</segment>
 </nucleus>
 <coda type="first">
 <segment phonation="voiced" manner="nasal"
  place="apical" duration="null">n</segment>
 </coda>
</syllable>
```

Figure 4: A typical syllable entry in *LeXMLicon*

Each syllable entry also has a child for every onset, nucleus and coda within that syllable, each containing an attribute *type,* which indicates its syllabic position, i.e. first onset, second onset etc. Each onset, nucleus and coda element can have only one child, namely *,* which contains the phonemic information at that particular position. Segment elements have attributes denoting the phonological feature information associated with the segment. So, in the example given we see that the segment in the first onset position, [S], has attributes indicating that it is a voiceless fricative etc.

One of the applications of *LeXMLicon* is to distinguish between actual syllables of a language and those syllables that conform to the phonotactics of the language and are well formed but which are not part of its lexicon. Taking *multilingual time maps* to be an extension of a phonotactic automaton, it is clear that it could output several syllable hypotheses when used in speech applications. In cases such as this, these candidate syllables can then be passed to *LeXMLicon* for lexical verification. Those that are found to be in the lexicon are deemed to be actual syllables of the language. Those that are not found in the lexicon are simply phontactically well formed, but are not actual syllables. Those syllables that are not found to be in the lexicon can then be passed to a native speaker, to evaluate whether or not they should be added to the lexicon.

Furthermore, smaller sub-lexicons can be easily derived using *LeXMLicon*. This can prove useful where a lexicon is required for a particular or restricted domain.

4. PROMPT

The third system presented is *PROMPT*, a Portable Resource Organiser for Managing Phonological Transformations. Utilising the portable technologies of Java and XML, PROMPT acts as a mapping manager that allows users to define new phoneme to feature associations and store them as *feature profiles* represented in XML. Each feature profile has XML elements containing the phonemic representation of a segment in a number of phonetic alphabets. The phoneme to feature attribute information that has been newly input by the user is then stored in the profile. In this way, *PROMPT* allows users to define and associate a feature set with several different phonetic representations and switch between them easily.

One of the functions of *PROMPT* is to manage the generation of additional lexicons from those output by *LeXMLicon*, by taking the output lexicon containing IPA-like phonological information as the structured knowledge base for transformations. The user then selects a previously input feature profile and *PROMPT* generates a new lexicon containing all of the same syllabic forms but with the segmental phonological information being sourced from the information input by the user via the feature profile. An advantage of this functionality is to allow lexicographers and speech scientists to explore the merits of different feature sets over others within the context of specific speech applications as numerous lexicons can be rapidly generated from the default IPA-feature based lexicon output by *LeXMLicon*.

Using *PROMPT*, users can also map data stored in both *multilingual time maps* and *LeXMLicon* into a variety of phonetic notations. Users choose a phonetic alphabet and can generate either a completely new *multilingual time map* or lexicon using it, thus increasing the versatility and flexibility of the mechanisms described. *PROMPT* also allows for additional notation systems to be defined by users and associated with particular feature profiles or alphabets.

Figure 5: One of the PROMPT GUIs

ISBN 1-876346-48-5

Furthermore, *PROMPT* can be used as a source of information for the generation of additional data tapes for *multilingual time maps.* For example, should users define a new feature set and associate it with a certain phonetic notation, a data tape for the *multilingual time map* transducer containing these new features can be quickly generated and dispersed through the network. It is envisaged that *PROMPT*'s feature profiles will be expanded to include information on visual events associated with the phonological information already described. Thus *PROMPT* can play an important role in expanding, updating and maintaining *multilingual time maps.*

5. CONCLUSION

The goal of ubiquitous language technology is to provide tools and resources that are both truly language independent as well as generic, resulting in their use by anyone, anywhere and on any platform. The m*ultilingual time map,* which extends the *Time Map* model's notion of a phonotactic automaton (an already robust and reliable model of computational phonology), contributes significantly to this goal. It is a portable technology, using XML as the data exchange format throughout. It builds on previous generic technologies and is language independent. Furthermore, its novelty lies in the focus on structuring many sources of related linguistic data within a common phonotactic context. It lends itself easily to expansion, i.e. the inclusion of additional data tapes not specifically mentioned here, whilst maintaining its common structure. Moreover, its use of XML means that this structure is easily manipulated into additional formats if required. The second system presented, *LeXMLicon*, also facilitates multilingual ubiquitous speech technology, focusing on providing a framework with the sole proviso that a language must be able to be represented using a phonetic notation, it provides a mechanism for generic lexical generation. The third module, *PROMPT*, offers a means of mapping the data in both *multilingual time maps* and *LeXMLicon* from one particular phonetic notation or feature set into another, whilst maintaining structural integrity. It thus increases the versatility of both representation mechanisms and forms an important component in the development of ubiquitous speech technology.

ACKNOWLEDGEMENTS

This material is based upon works supported by the Science Foundation Ireland under Grant No. 02/IN1/I100. The opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Science Foundation Ireland.

REFERENCES

[1] L. Cahill and G. Gazdar, "The PolyLex architechture: multilingual lexicons for related languages," in *Traitement Automatique des Langues,*40(2), 5-23, 1999.

[2] J. Carson-Berndsen, "Multilingual Time Maps: Portable Phonotactic Models for Speech Technology Applications," in *Proceedings of the LREC 2002 Workshop on Portability Issues in Human Language Technology*, Gran Canaria, 2002.

[3] J. Carson-Berndsen, *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition* ,Dordrecht: Kluwer Academic Publishers, 1998.

[4] J. Carson-Berndsen, "A Generic Lexicon Tool for Word Model Definition in Multimodal Applications," in *Proceedings of EUROSPEECH '99, 6th European Conference on Speech Communication and Technology,* Budapest, 1999.

[5] J. Carson-Berndsen and M. Neugebauer , Die Rolle der Phonologie in der multilinguale Sprachtechnologie, *Proceedings of the GLDV-Frühjahrstagung 2003, Sprachtechnologie für die multilinguale Kommunikation,* 2003.

[6] J. Carson-Berndsen, U. Gut and R. Kelly, Discovering regularities in non-native speech. In: A. Wilson, P. Rayson and D. Archer (eds), *Corpus Linguistics Around the World*, Rudopi, 2003.

[7] J. Carson-Berndsen and M. Walsh, "Generic techniques for multilingual speech technology applications," in *Proceedings of the 7th Conference on Automatic Natural Language Processing,* Lausanne, 2000.

[8] J. Carson-Berndsen and M. Walsh, "Interpreting Multilinear Representations in Speech," in *Proceedings of the 8th Australian International Conference on Speech Science and Technology,* Canberra, 2000.

[9] R. Evans and G. Gazdar, "DATR: a language for lexical representation," in *Computational Linguistics* vol 22,2 pp. 167-216, 1996.

[10] J. Goldsmith, *Autosegmental and Metrical Phonology,* Basil Blackwell, Cambridge, MA, 1999.

[11] C. Tiberius and R. Evans, "Phonological feature based Multilingual Lexical Description," in *Proceedings of TALN 2000,* Geneva, 2000.

[12] M. Walsh, S. Wilson and J. Carson-Berndsen, "XiSTS – XML in Speech Technology Systems," in *Proceedings of the 2nd Workshop on NLP and XML,* Taipei, 2002.