

A Multilingual Phonological Resource Toolkit for Ubiquitous Speech Technology

Daniel Aioanei, Julie Carson-Berndsen, Anja Geumann,
Robert Kelly, Moritz Neugebauer, Stephen Wilson

Computer Science Department
University College Dublin
Belfield Dublin 4, Ireland

{daniel.aioanei, julie.berndsen, anja.geumann,
robert.kelly, moritz.neugebauer, stephen.m.wilson}@ucd.ie

Abstract

This paper outlines the generation process of a specific computational linguistic representation termed the Multilingual Time Map, conceptually a multi-tape finite state transducer encoding linguistic data at different levels of granularity. The first component acquires phonological data from syllable labeled speech data, the second component defines feature profiles the third component generates feature hierarchies and augments the acquired data with the defined feature profiles and the fourth component displays the Multilingual Time Map as a graph.

1. Introduction

This paper presents the first prototype of a toolkit which supports the development of multilingual phonological resources. The toolkit consists of a number of components which interact to provide phonological resources for many languages. This is a companion paper of two other papers presented at this conference (Carson-Berndsen and Kelly, 2004; Neugebauer and Wilson, 2004). The paper is best read in conjunction with those papers. It serves to document the system as a whole and to demonstrate how multilingual resources of structured phonological information can be built. The basis for the resources is the phonotactic automaton (Carson-Berndsen, 1998) which describes all permissible combinations of sounds in a language within the syllable domain.

The toolkit consists of four main components: a Phonotactic Automaton Learner, a Feature Definition Component, a Feature Hierarchy Induction Component, and a Visualisation Component. The functionality of the toolkit will be demonstrated at the conference using a number of speech databases. In the demonstration, we will show how the feature augmentation and feature induction components are employed to build a more compact representation in our knowledge base for individual speech sounds as well as for complete sets of lexical items.

2. Phonotactic Automaton Learner

The Phonotactic Automaton Learner (PAL) infers a finite-state representation of the permissible combinations of sounds found in syllable labelled speech data. This represents the first stage in acquiring phonological resources using the toolkit. The finite-state structure inferred through PAL is thus the foundation structure on which the resulting *Multilingual Time Map* (MTM) is built.

Since the number of possible syllables in a language is finite and also since syllable-based phonotactics has been shown to be representable as a finite-state structure called a phonotactic automaton (Carson-Berndsen, 1998), it is possible to infer the structure of the phonotactics for a language

from a positive sample of syllables from that language. While future work requires that we experiment with different inference procedures the current incarnation of PAL makes use of an implementation of the ALERGIA regular inference algorithm (Carrasco and Oncina, 1999). Using the syllable samples, ALERGIA infers a deterministic minimal stochastic finite-state automaton that accepts at least the training set of syllables. Space prohibits a full discussion of the ALERGIA algorithm however further details concerning the inference algorithm applied to the task of learning phonotactic automata can be found in (Kelly, 2004) and (Carson-Berndsen and Kelly, 2004).

While the ALERGIA algorithm produces stochastic automata, PAL has been extended to produce an MTM, an XML representation which extends the finite-state structure of the phonotactic automaton to multiple tapes. Each arc of the MTM contains a segment label, a frequency of occurrence of this segment in this phonotactic domain and a probability. The MTM can be extended to include further tapes denoting an average duration and a standard deviation of duration for the segment appearing on the associated segmental tape if timing information is available for the syllables in the sample.

An example of the header and one arc (transition) in an MTM is shown in figure 1. There are 4 tapes of information on the transition from the state labelled 0 to the state labelled 2; a segment label (phoneme /S/¹), a frequency for this transition, an average duration associated with the phoneme /S/ in the phonotactic context of the transition from state 0 to state 2 and finally a weight tape denoting an inverse log probability for the transition.

While the inference procedure discussed above for obtaining the initial MTM structure is fully automatic it requires that a corpus of syllable labelled utterances be available. Thus, the possibility of applying the technique outlined in this paper to obtain a MTM for a particular language is dependent on the existence of a corpus for the language which is annotated at the syllable level. Since cor-

¹shown in SAMPA notation

```

<MTM language="ENG">
  <startStates>
    <state>0</state>
  </startStates>
  <finalStates>
    ...
  </finalStates>
  <transition>
    <sourceState>0</sourceState>
    <destinationState>2</destinationState>
    <phonemeTape>S</phonemeTape>
    <frequencyTape>2</frequencyTape>
    <durationTape>72</durationTape>
    <weightTape>2.302</weightTape>
  </transition>
  ...
</MTM>

```

Figure 1: Portion of the XML representation of an MTM.

pora are usually annotated at the segment and word level but not the syllable level, compiling a syllable labelled corpus, either from scratch or by extending the annotations of an extant corpus, may be time consuming and expensive. This can be overcome somewhat by extending PAL such that it can be used as an annotation assistant in a semi-automatic approach to deriving syllable labelled data from phoneme labelled data. This is discussed further in (Kelly, 2004) and (Carson-Berndsen and Kelly, 2004). Also, since the inference procedure is fully automatic, the quality of the learned automata will be dependent on the quality of the corpus annotations. Fortunately, the need for high quality annotated corpora is now recognised and has become an essential part of speech and language technology research. Much emphasis is being placed on the production of multi-level annotated corpora in particular. While we assume here that a high quality annotation is available, we are developing techniques for automatic verification and consistency checking of annotated corpora to ensure that this is the case. In addition, the completeness of an MTM inferred using the technique discussed here is dependent on the completeness of the supplied training sample of syllables. Thus, if a rare but valid sound combination is not detailed in the training sample then it will never be represented in the inferred MTM. This is a problem faced by all inference procedures and is typically overcome through the use of some generalisation technique. A simple technique is discussed in (Kelly, 2004) which imposes the basic onset-peak-coda syllable structure onto the training set of syllables. In addition, it is possible to make further use of linguistic information regarding the segments labelling the transitions to achieve further phonological generalisation. This requires that the transitions of the initial MTM be augmented with a further tape specifying information regarding the features associated with the corresponding segmental tape.

3. Feature Definition Component

A feature definition module enables users to define a multilingual feature inventory for a particular symbol set across a number of languages. The inventory is modelled

as an XML tree, termed a feature profile

The module combines a data driven approach to interface generation with user driven acquisition of phonological information, facilitating the efficient encoding of symbol-to-feature attribute mappings for those symbols that occur in the MTM. Symbol-to-feature mappings refer to the explicit linking of phonetic symbols with a number of phonological features at varying levels of granularity.

Features may be defined as unary, binary or multilevel entities. Unary features may be considered to be properties that on their own can be assigned to segments; binary features are attribute/value pairs which have two mutually exclusive values; multi-value feature structures consist of a number of tiers of information, each of which has an associated set of phonological features as parameters, from which one is chosen. The module automatically generates a symbol input interface by extracting all unique occurrences of symbols within the MTM. Since the process of acquiring a fully specific MTM is incremental, subsequent passes through the growing finite state representation create input interfaces for those features that do not yet exist in the inventory. In this way we seek to avoid input repetition as the inventory gradually approaches a full description for a symbol set.

In the case of unary features, users must input the full set of possible feature values. A Document Type Definition (DTD) is automatically generated from this data and is used in validating future feature profiles that are created using the same feature sets. A feature input interface is subsequently generated from the DTD. Symbol-to-feature associations are created graphically by selecting a symbol from the symbol input interface and clicking on those features that are to be associated with it. Associations between symbols and binary features are made in a similar fashion. For multi-value feature structures the set of tiers required along with the possible values for each tier are input. The module creates a DTD and interface from the data and the process of association proceeds as above.

The symbols used have an underlying IPA-Unicode representation, but a notation transducer allows any feature inventory that has been defined to be mapped into a number of phonetic alphabets (e.g. SAMPA, ARPAbet, etc.).

Interfaces for the graphical editing of data within the inventory, including display, deletion and modification of tiers and nodes, are provided as well as interfaces for the selection of particular functions for the manipulation of data, e.g. extracting language specific feature profile from the superset of all feature associations.

All user activity takes place in the graphical environment; all processing remains hidden and we presume no knowledge of the technologies used on the part of the user. By representing the information within an XML based structure it is readily accessible for use by a wide variety of groups, users and applications for any number of purposes.

4. Feature Hierarchy Induction

Phonological feature profiles lead to an expressive knowledge base which provides a fine-grained level of description for the modelling of individual phonological segments. However, a rich set of features - despite its descrip-

tive value - might not be easily accessible for manual optimisation, such as identification of implicational relations between individual features as well as possible combinations of features.

To obtain this valuable information while equally limiting the need for manual effort, a computational method is proposed based on automated deduction to deliver correspondences between individual features and furthermore between all sets of sounds created by combinations of those. Once the phonological feature trees have been defined via the previous module, these trees are traversed with the aim to perform as much deterministic inference as possible. The algorithm is applied to automatically generate feature hierarchies similar to type hierarchies in unification-based grammar formalisms, where features are ordered with respect to the size of their extents, i.e. the segment set they describe.

In contrast to other implementations of feature formalisms such as DATR (Evans and Gazdar, 1996) or LKB (Copestake, 2002) the denotational semantics of XML do not allow to express multiple inheritance. Therefore every single combination of features is "multiplied out" to achieve its extent in terms of phonological segments. Finally this information is used to enrich the current phonological feature profile with two elements distinguishing between bi-directional and unidirectional implications. To carry out efficient updates on the lexical knowledge base XSL, a stylesheet language for transforming XML documents is used. In this case the intention is to unify the set of all feature profiles with generalizations over this particular set yielding a more expressive feature profile

The following example displays the feature association tree for the segment [l] after it has been enriched with all logical implications gained from multiple tree traversal in the lexical knowledge base (note: the "features" subtree has been omitted):

```
<featureAssociations>
<symbol>l</symbol>
<features> ... </features>
<introducing>lateral</introducing>
<sharing>consonantal, nonvocalic
, ..., coronal </sharing>
</featureAssociations>
```

Figure 2: Augmented segment entry

It can be seen from this single entry that it is now possible to state that the segment [l] introduces the feature [lateral] to our feature trees which in turn means that the presence of a segment [l] can be inferred simply given the featural information [lateral]. Additionally, it can be observed that all features apart from [lateral] do not imply presence of the segment in question since they also occur in feature associations of other segments. In Figure 3 is given a tree representation which encodes this information.

The information within the final feature profile trees is used in the augmentation of the MTM. The mapping component of the module traverses the MTM and inserts a tape containing the phonological feature information for each

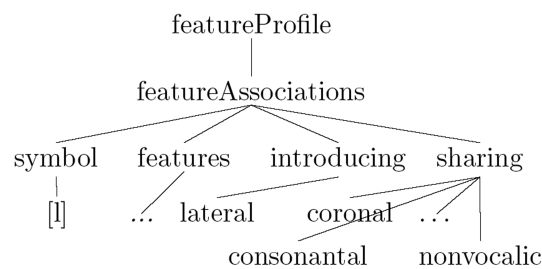


Figure 3: Tree representation for Figure 2

occurrence of the associated symbol within the network. Similarly, once the feature profile has themselves been augmented with information regarding optimised feature sets, data tapes indicating feature redundancy or unicity can be extracted and dispersed throughout the MTM.

This section integrated these individual descriptions in terms of phonological feature trees with an account for generalizations over the set of all lexical entries which allow the set of characteristic features for each segment to be split into shared ones and features which are unique for a specific phonological segment (Neugebauer and Wilson, 2004). By these means, even a fairly large multilingual feature set can be maintained as well as mined for language-dependent and language-independent phonological implications.

5. Visualisation Component

The final component visualises the MTM after each stage of processing. It offers various views of the MTM from the basic topology i.e. just the nodes and the arcs, to fine grained views showing all the information annotated on the arcs such as phonemes, features, average durations, etc. The output format is Scalable Vector Graphics (SVG).

The visualisation component consists of three modules, each of which transforms one representation of the MTM to another, the final one being SVG. The first module transforms the XML representation of the MTM, as it was output by any of the previous three components, into a DOT language representation (Gansner et al., 2002). It does so by taking only the source and destination states of every transition, and the phoneme tape to label the transition in the resulting graph. In the case where there are multiple transitions between two states in the XML representation, it creates only one transition in the DOT representation, and labels it with a comma separated list of the phonemes from the individual transitions in the XML representation. The reason for doing this is to increase the readability of the resulting graph (see the /uw/, /ae/ example in Figure 4).

The second module takes the DOT representation generated by the first module and uses the *dot* application (Gansner et al., 2002) to transform it into an SVG file

The third module modifies the SVG representation generated by the second module by introducing mouse events so that all the information associated with a certain transition in the XML representation is displayed when the mouse hovers over the phonemic label of that specific transition. The extra information displayed for each phoneme is the frequency and the weight of that phoneme, and all

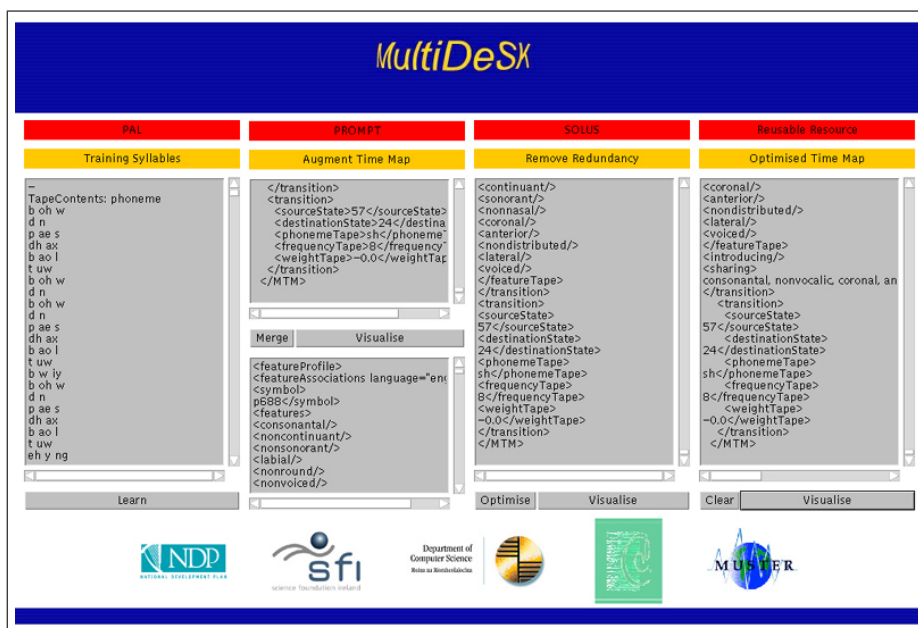


Figure 5: Screenshot of the toolkit.

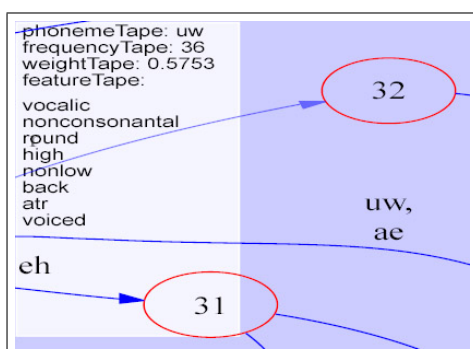


Figure 4: Detail of the visualisation tool. The graph is zoomed in so that the relevant information appears in the viewport. The mouse hovers over the /uw/ phoneme.

the features that define it. Figure 4 shows the details for the /uw/ phoneme.

The features that are unique for the current phonemic segment are indicated with a star (*) symbol next to them. All the other features are shared with other phonemic segments. The visualisation allows easy access to different levels of information, which can be a useful tool in learning/teaching computational linguistics, and for easy access to language documentation.

6. Conclusion

This paper describes an integrated toolkit (see Figure 5) of the individual components presented above, which will be shown as a poster and as a demonstration during the practical session.

²shown in ARPAbet notation

7. 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.

8. References

- Carrasco, Rafael C. and Jose Oncina, 1999. Learning deterministic regular grammars from stochastic samples in polynomial time. *ITA*, 33(1):1–19.
- Carson-Berndsen, Julie, 1998. *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition*. Dordrecht, Holland: Kluwer Academic Publishers.
- Carson-Berndsen, Julie and Robert Kelly, 2004. Acquiring reusable multilingual phonotactic resources. To Appear in Proceedings of the 4th International Conference on Language Resources and Evaluation.
- Copestake, Ann, 2002. *Implementing Typed Feature Structure Grammars*. Stanford: CSLI.
- Evans, Roger and Gerald Gazdar, 1996. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22(2):167–216.
- Gansner, Emden, Eleftherios Koutsofios and Stephen North, 2002. *Drawing graphs with dot*. AT&T Labs - Research.
- Kelly, Robert, 2004. A language independent approach to acquiring phonotactic resources for speech recognition. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*. CLUK04.
- Neugebauer, Moritz and Stephen Wilson, 2004. Phonological treebanks - issues in generation and application. To Appear in Proceedings of the 4th International Conference on Language Resources and Evaluation.