

Traversing Phonological Feature Trees

Moritz Neugebauer and Stephen Wilson

Department of Computer Science

University College Dublin

Republic of Ireland

Abstract

Current research in computational linguistics makes frequent use of tree-based data structures for documentation as well for analysis. Proposals for annotation standards focus mostly on natural language processing with syntactic and morphological data. This paper shows how even the treatment of phonological data can benefit from representations of this kind. For this purpose we describe original work on two integrated modules: the first defines a multilingual feature set within a tree based structure represented in XML, the second traverses this feature tree and generalises over the data contained in it, highlighting feature implications. The mapping component of the integrated system then takes the information contained within the feature tree and uses it to augment a specific phonological representation, the *multilingual time map*. The integrated system described here takes the form of a graphical user environment, which presumes no knowledge of the technologies used on the part of the user.

1 Introduction

In sharp contrast to research on other subfields of natural language processing (NLP) where numerous schemes for the annotation of linguistic information have been proposed, information concerning the phonological level of linguistic description lacks comparable proposals for a standardised scheme. Based on the format we present in this paper, computational linguistic tools can access as well as share phonological data in a uniform, clearly defined representational format. The need for research on standards of this kind has been expressed from a more general point of view in Ide (1999, 17) and in the following sections we will try to contribute to this body of research by encod-

ing phonological data. Thus, the two following points constitute the principal motivation of our work:

- formats: development and use of consistent and coherent encoding formats for data representation, as well as standardised schemes for annotation of linguistic information
- tools: development of reusable, integrated systems and tool architectures for language processing and analysis, including the corresponding development of a data architecture to best suit research needs.

The Extensible Markup Language (XML) is used frequently to model natural language data in large-scale applications. A common application task is to parse a marked up document to retrieve its data. The Document Object Model (DOM) is essential to working with XML since it defines a set of interfaces for referring to, retrieving, and changing items within an annotated structure (cf. Kesselman et al. (2000)). In this paper we use Java technology to create an object model from an XML file, make changes to it, and retrieve the output.

XML-documents can be represented in one of two fashions: as a flat structure or as a tree structure. The `NodeIterator` interface declares the methods that allow programmers to traverse a flat representation of an XML document. The `TreeWalker` interface declares methods that allow programmers to traverse a tree representation of an XML document. The `NodeFilter` interface allows programmers to create filters that select which items from the XML document are included in the logical view of the application. In this paper, we exploit this technology and present an abstract data model for linguistic annotation and tools for the traversal of particular

phonological annotations. With regard to the content of our database we consider both dimensions of phonological structure, that is segmental feature associations and phonotactic finite-state networks encoding possible groupings of sounds into syllables.

In the following sections we will present two modules facilitating the generation and retrieval of an annotation format introduced in earlier work as the *multilingual time map* (cf. Carson-Berndsen (2002), Walsh et al. (2002), Carson-Berndsen and Neugebauer (2003)). The multilingual time map format represents an extension to finite-state networks which model only phonotactic constraints since each transition now includes information about phonemes but also graphemic, allophonic and articulatory feature information. With respect to the application of our tools to corpus-specific studies our format also offers space for adding average duration, frequency and probability for each individual segment entry.

2 Defining Traversable Phonological Feature Trees

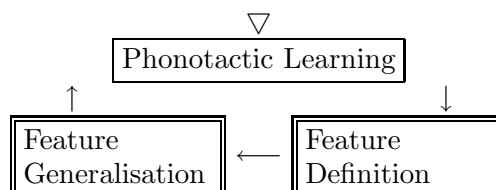
Phonological feature trees are documents that consist of a set of user defined symbol-to-feature attribute mappings. The creation of phonological feature trees was motivated by the desire to create repositories of phonological feature information that was explicitly linked to particular symbol sets and which could be sourced for use during document generation and mapping. The module described here provides an environment for user driven acquisition of such repositories, but more importantly it annotates and stores the information in a useful and coherent data structure.

Once the associations have been defined, they are stored in a structured XML tree called a feature profile. These profiles are intended to be multilingual resources in the sense that they should provide a full inventory of phonological features for a complete symbol set across a number of languages. While the feature set is shared by all languages, symbol sets are language-dependent. The process of acquiring this inventory is designed to be incremental: each defined set of symbol-to-feature associations is annotated with respect to a particular language, and indicates those languages for which the par-

ticular combination of features is valid. Thus the association of a particular set of features with some symbol may be present in a number of languages and will be annotated accordingly. Smaller subsets of associations, which can be considered as individual profiles for particular languages, can naturally be extracted as required.

The advantages of structuring the information in XML are twofold: firstly, the data is readily accessible for processing and its the set of annotation tags is able to express more inherent logical structure than were it stored in a flat document. Secondly, using XML as the exchange format means that the information contained within feature profiles is easily reusable by a wide group of people and applications, as user-required idiosyncratic structures and program-specific formats can be readily generated from the data.

The phonological trees discussed in this paper are used in the augmentation and management of a number of specific phonological representations. They form the second and third phases of the cycle shown below describing the production of a multilingual time map. We seek to partially learn the time map’s structure in phase 1 of the above diagram; this structure is then used to generate the interface for the feature definition module (phase 2); once feature-to-symbol associations have been created in phase 2., the optimisation phase generalises over the feature set, adding additional information regarding logical relations among individual as well as sets of features (phase 3). The diagram below shows all three phases, while in this paper we focus on the *Feature Definition* and the *Feature Generalisation* components of the cycle.



Visualisation of the annotation cycle

The cycle’s initial phase sees a set of syllables input to an in-house tool named *Phonotactic Automaton Learner* (PAL) which automatically generates a finite state machine that models all the legal combinations of sounds within

the supplied domain. This structure forms the basis for the multilingual time map transducer. We wish to augment each transition within the time map with an additional tape that supplies information about phonological feature overlap relations and the feature definition module provides the means to do so.

Taking the multilingual time map transducer that is output from phase 1, the feature definition module extracts every unique occurrence of a phonological symbol from the finite state network and dynamically creates a graphical user environment that allows the user to define feature associations for those symbols. Automatically creating an interface means that users define associations only for those symbols that occur in the data set. As the acquisition of fully specified multilingual time maps – like feature profiles themselves – is intended to be completed over periodic stages, subsequent passes through the growing finite state network dynamically create feature input interfaces for symbols that do not yet appear in the feature profile inventory. In this way, repetition of user input is avoided and the inventory comes closer to describing a full symbol set. The actual process of definition and association of symbols and features is described below.

The module is a graphical user environment that provides several avenues enabling users to create symbol to feature attribute associations. Users initially select whether their feature profile shall consist of unary, binary or multilevel feature structures. 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; multilevel 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.

In the case of a feature profile consisting solely of unary features, the user inputs the full set of features required. This set is stored internally and a Document Type Definition (DTD) is automatically generated from the data and it will be used in the creation and validation of subsequent profiles comprising the same feature set. Below we provide a selection of a DTD for a multilevel feature profile where

an attribute like *manner* might take values describing various broad sound classes.

```
<!ELEMENT featureProfile (featureAssociations)*>
<!ELEMENT featureAssociations (symbol,features*)>
<!ELEMENT symbol (#PCDATA)>
<!ATTLIST symbol notation ( IPA | SAMPA) #IMPLIED>
<!ELEMENT features (phonation?,manner?)>
<!ELEMENT phonation (#PCDATA)>
<!ELEMENT manner (#PCDATA)>
<!ELEMENT place (#PCDATA)>
```

Example: Document Type Definition

Once the full feature set has been input, a graphical table representing the dataset is generated. Although the symbols have an underlying IPA-Unicode representation, the module contains a notation transducer which allows for the smooth mapping of symbols from this representation to a number of alternative phonetic alphabets, (e.g. SAMPA, Worldbet, ARPabet etc). Any mappings between notations maintain the structural integrity of the feature profile, meaning that the combinations of phonological features as defined by the user get mapped and associated with the new transduced symbol. Alternatively, users may explicitly select a notation other than IPA before they create a new feature profile. Once a symbol has been selected, the feature set that has been input displays, and the user simply points and clicks to associate features with that symbol. At all times, the list of associated features can be reviewed, edited or deleted. The symbol with its associated features gets added to the feature profile subject to user confirmation. Feature profiles consisting of binary feature associations are constructed in a similar fashion.

For multilevel feature structures, the user must first input the required set of tiers, and for each tier of information the required set of features. For example, users may wish to create a feature profile with tiers of information describing phonation, manner and place of articulation etc., and such tiers might have as features: voiced, nonvoiced; fricative, plosive; labial, glottal and so on. As with unary and binary feature profiles, a DTD which will be used for generation and validation of future profiles is automatically created from the data. The association of symbols with features proceeds as described above, with the addition of an

extra step, whereby the user must first select a tier before selecting features.

It is important to note that no knowledge of XML is required on the part of the user, and indeed that all background processing of the information remains hidden at all times. The feature definition module's environment provides the user with a number of graphical interfaces for the following:

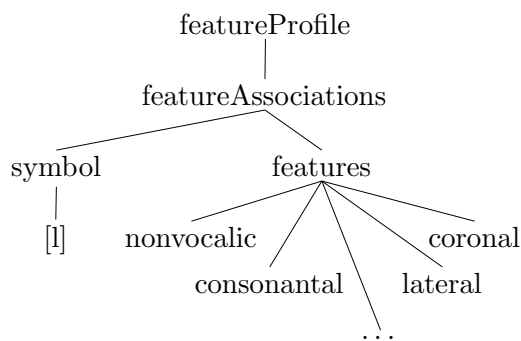
- an interface for displaying the structure of the phonological feature tree and its all its nodes containing articulatory information as well as the language for which it is defined
- an interface which allows the user to graphically edit the information contained within the feature profile
- an interface for selecting particular functions for the manipulation of the data contained within the profile, e.g. extract language specific associations from the superset of all feature associations contained within the profile.

Other functional interfaces include means for transducing new feature profiles based on different phonetic notations, and for augmenting multilingual time maps. It is thus clear, that in addition to processing and data exchange benefits, the structuring of the profile's data in XML lends itself to manipulation for graphical display. The following sections show how we take the information contained within the feature profile and make generalisations over it, seeking to optimise the data.

3 Applications for Phonological Tree Traversal

The above motivations for 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, we acknowledge the fact that such a rich set of features – despite its descriptive 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. Segment entries of the following kind – here for the segment [l] – represent the input data for our automated

method to extract information about feature distributions in our database.

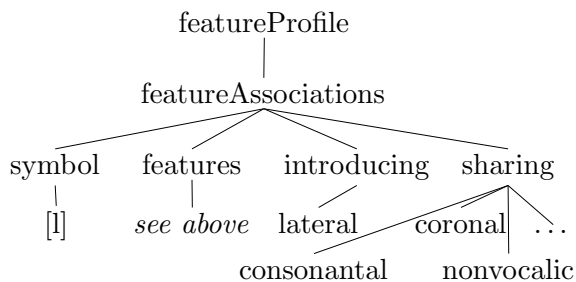


Entry for [l] after the first module

To obtain this valuable information while equally eliminating the need for manual effort, we propose a computational method 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, we traverse these trees with the aim to perform as much deterministic inference as possible. We apply our algorithm 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 feature formalisms such as DATR (cf. Evans and Gazdar (1996)) or LKB (cf. Copestake (2002)) the operational semantics of XML does not allow to express multiple inheritance. We therefore choose to "multiply out" every single combination of features to achieve its extent in terms of phonological segments. Finally this information is used to enrich the current phonological feature profiles with two elements distinguishing between bi-directional and unidirectional implications. To carry out efficient updates on our lexical knowledge base we use XSL which is a stylesheet language for transforming XML documents. In our case we intend 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 our lexical knowledge base (note: the "features" subtree has been omitted):



Entry for [l] after the second module

We can see from this single entry that we are now able to say that the segment [l] introduces the feature [lateral] to our feature trees which in turn means that we can infer the presence of a segment [l] and all its additional features simply given the featural information [lateral]. Furthermore, we can observe 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. All this information is based on automatically generated feature hierarchies as described in Neugebauer (ms). The core of this work consists in an algorithmic method to deduce hierarchies which encode inheritance relationships among sets of segments and features for a single language or even for different languages. For the purposes of this paper, the following three steps summarise the procedure which augments the *introducing* and *sharing* nodes in the feature trees.

1. for each feature defined in the feature profile, traverse the individual feature associations to determine the extension (the segments for which the feature is defined) of each single feature
2. create all the sets of segments which are denoted by single features and if the set contains only exactly one element, add an *introducing* node to the feature tree of that particular element
3. compute the complement of that feature and store the result as a value of the created *sharing* node of the feature tree

Our method does not only serve to establish implications which hold for the domain of single segment entries but since we generalise over the all entries we also achieve similar information for whole sets of sounds. Consider for instance the following implicational generalisations which are provided in the table below: if we know the features in the leftmost column, we can infer the features to their right. The set of sounds in the final column is the set of sounds which share the union of unique and shared features; in the last row all elements which carry the feature [round] are displayed which maps onto the set of round voiced vowels which is expressed in the following implication rule: [round] → [voiced, vocalic].

high	voiced, vocalic, round	{iy, uw}
front	voiced, vocalic, round	{iy}
back	voiced, vocalic, round	{ao, uw}
semilow	voiced, vocalic, round	{ao}
round	voiced, vocalic	{iy, ao, uw}

Logical relations between articulatory features (selection)

The information within the final feature profile trees is used in the augmentation of a specific phonological representation, the multilingual time map. Multilingual time maps can be conceptualised as multi-tape finite state transducers, that define well formed combinations of phonological segments within the syllable domain. The fully specified feature associations for each symbol within a particular feature profile tree are extracted and used to construct an additional input tape for the multilingual time map transducer. The mapping component of the module traverses the multilingual time map and inserts a tape containing the phonological feature information for each occurrence of the associated symbol within the network. Similarly, once the feature profiles have themselves been augmented with information regarding optimised feature sets, data tapes indicating feature redundancy or unicity can be extracted and dispersed throughout the multilingual time map.

Our approach integrates a novel formulation of phonological feature trees with tree traversal yielding a fine-grained characterization of

segmental units. Apart from these individual descriptions we also account for generalizations over the set of all lexical entries which allow us to split the set of characteristic features for each segment into shared ones and features which are unique for a specific phonological segment. 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.

4 Summary and Outlook

This paper focussed on the computational linguistic aspects of phonological feature tree traversal while we also already made reference to ongoing work from an application-oriented perspective. We aim to employ the insights we presented in this paper to the generation and application of phonological treebanks as described in Neugebauer and Wilson (submitted). Over the past years, treebanks have become important in various areas of computational linguistics ranging from syntactic to morphological and semantic-pragmatic applications. The continuing popularity of XML as a data exchange format and the concurrent rise of treebanks as natural language resources within the above domains have naturally led us to extend their domain of application to phonological data. Typically, treebanks are a language resource that provides annotations of natural languages at various levels of structure and in this paper we presented a tree-based format to capture phonological information at the segmental level including articulatory information.

Therefore, the most significant aspect of future research has to be the development of phonological treebanks considering syllable and word level while using the annotation format for segments which we described in this paper. In this final section we sketch necessary steps of this work by beginning with the common distinction between tagged and parsed corpora while we assume that the latter corresponds to what is usually referred to as *treebanks*. Our current feature profiles (including the feature associations) can be employed to create tagged corpora in the sense that all segmental units will be tagged with respect to detailed articulatory information. The next step which has not been

described in this paper consists in the acquisition of corpus-specific information such as duration of segments, syllables and words. Considering further linguistic applications, corpus queries may be carried out determining the distribution of a feature/segment in a spoken language corpus, that is its corpus frequency and prosodic context.

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

- Julie Carson-Berndsen and Moritz Neugebauer. 2003. Die Rolle der Phonologie in der multilingualen Sprachtechnologie. *LDV-Forum: Journal for Computational Linguistics and Language Technology*, 18(1,2):199–216.
- Julie Carson-Berndsen. 2002. Multilingual time maps – portable phonotactic models for speech technology applications. In *Proceedings of the LREC 2002 Workshop on Portability Issues in Human Language Technology*, Las Palmas.
- Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. Stanford: CSLI.
- Roger Evans and Gerald Gazdar. 1996. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22(2):167–216.
- Nancy Ide. 1999. Encoding linguistic corpora. In *Proceedings of the Sixth Workshop on Very Large Corpora*.
- Joe Kesselman, Jonathan Robie, Mike Champion, Peter Sharpe, Vidur Apparao, and Lauren Wood. 2000. Document Object Model (DOM). Level 2 traversal and range specification. Technical report, W3C, <http://www.w3.org/TR/2000/REC-DOM-Level-2-Traversal-Range-20001113>.
- Moritz Neugebauer and Stephen Wilson. submitted. Phonological treebanks - issues in generation and application. submitted to the

Fourth International Conference on Language Resources and Evaluation (LREC 2004).

Moritz Neugebauer. ms. Computational phonology with set descriptions and lattice algorithms. article in preparation.

Michael Walsh, Stephen Wilson, and Julie Carson-Berndsen. 2002. XiSTS – XML in speech technology systems. In *Proceedings of the COLING 2002 Workshop on XML & NLP*.