Introduction to computational phonology

Bird, Steven

Published in:
Computational Linguistics

Published: 01/01/1994

Document Version
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Bird, S. (1994). Introduction to computational phonology. *Computational Linguistics*, *20*(3), 1-8.

Introduction to Computational Phonology

Steven Bird[*]
University of Edinburgh

1. Overview

Despite being the oldest discipline in linguistics, phonology remains largely unexplored from a computational standpoint. While phonology gave us such innovations as the 'distinctive feature', now heavily used in computational linguistics, phonology itself is yet to reap the benefits of the formal and technological developments it gave rise to.

Recently however, computational phonology has been rapidly gaining recognition as an independent area of inquiry within computational linguistics. The *ACL Special Interest Group in Computational Phonology* (SIGPHON) was formed in 1991 and has served as a focus for ongoing work in the area. In June of that year I proposed that there be a special issue of *Computational Linguistics* dedicated to computational phonology, since there were many good-quality papers in circulation which had no obvious venue for publication. The resulting collection which you have before you is a representative sample of this work; some submissions not ready in time for this volume will appear in subsequent regular issues. Other work in this area is to be found in the *Proceedings of the First Meeting of the ACL Special Interest Group in Computational Phonology*, published by the ACL in 1994, and two edited collections (Bird 1991, Ellison & Scobbie 1993).

The purpose of this short piece is to introduce computational phonology and the special issue. I shall begin by presenting some background to the field, followed by a survey of the research themes currently under investigation. Next, an overview of the papers in this collection is given, concluding with an explanation of the one-page commentaries which follow each paper. So, what is phonology, and why should computational linguists care about it?

2. Background

Phonology is the study of the systems of sounds which are manifested by natural languages, the significant contrasts between sounds that are relevant to meaning. As such, phonology stands at the interface between grammar, broadly construed, and speech. Much of the richness and complexity of phonology derives from the place it occupies between categorical symbolic systems and parametric physical behaviour. Several excellent textbooks are available for readers who wish to learn more about phonology.

Now, why should computational linguists care about phonology? First, phonology is an equally valid area of study for a computational linguist as syntax or semantics. Solutions in one area may generalise to other areas, as we see, for example, where strings of segments are parsed using the same machinery that is used for syntactic parsing (e.g. chart parsing, Church 1987), or where a formal system which was developed for semantic representations of tense is applied to the temporal structure of phonology (Bird

[*] University of Edinburgh, Centre for Cognitive Science, 2 Buccleuch Place, Edinburgh EH8 9LW, UK

& Klein 1990), or where complex arrangements of phonological features are represented in the familiar notation of attribute-value matrices (Wiese 1990). Thus, phonology provides a fresh source of applications for the techniques and technologies of computational linguistics.

However, this only demonstrates a flow of information from computational linguistics to phonology. Can we hope for payoffs in the other direction resulting from a wholesale integration of phonology into computational linguistics? It is instructive to consider *The Sound Pattern of English* (Chomsky & Halle 1968) in this regard. Although it was intended as a contribution to phonological theory, SPE was also directly implementable on computer (e.g. Bobrow & Fraser's 'phonological rule tester', 1968) and it was an important foundation for work in speech technology (e.g. Allen et al. 1987). Via the work of Johnson (1972), Koskenniemi (1984) and Kaplan & Kay—the latter in circulation since the early eighties but appearing in published form for the first time in the present collection—one could reasonably argue that SPE gave rise to finite-state morphology (Antworth 1990, Ritchie et al. 1992, Sproat 1992).

The formal framework of SPE was a good deal more explicit and rigorous than most of what came after, and so the prospects for a repeat performance coming from phonology have never been particularly bright. However, I feel it is now time for computational linguists to take another look at phonology. A quarter of a century has gone by since SPE, and there is much of interest to be found in the pages of *Phonology* and similar publications. A good place to start is the literature on computational phonology itself, since it interprets the theoretical proposals of phonology in a way that is more accessible to computational linguists. The stakes are high, since it would not be surprising if phonology is still to play an important role in bridging natural language technology and speech technology. For even though there is a methodological and sociological divide, there remains an imperative to develop fully integrated language and speech systems and an enduring need for sources of creative ideas to relate the discrete to the continuous.

Undoubtedly, there will always remain sceptics who think that natural language systems which deal just with the written word can afford to ignore phonology. In a limited sense they are correct. However, in the longer term, I am convinced that the interest in multilingual and multimodal systems will require a more enlightened view of phonology. Many languages have genuinely phonological phenomena evident in the orthography, such as Finnish (Koskenniemi 1984). Even in English we find cases where a spelling rule needs to be sensitive to phonological information. For example, the orthographic rule which selects a vs. an breaks down when a following word begins with a *written* vowel but a *spoken* consonant or vice versa, as in a uranium compound and an ytterbium compound. Although it largely works for English and a handful of other languages, the assumption that phonology can be ignored by natural language systems will collapse for many of the world's languages (e.g. Finnish, Turkish and Arabic).

One reason why computational phonology has not had a high profile is that work in this area has often been dealt with under the heading of computational morphology. However, much of what passes as finite-state morphology is actually *morphophonology*—the phonological factors which influence the appearance of morphemes—or even phonology proper. Moreover, the central computational device in finite-state morphology, the finite-state transducer, is not used for specifying the distribution of morphemes (i.e. morphotactics), the other main task of morphology. Therefore, that part of finite-state morphology which is expressed in terms of finite-state transducers, namely morphophonology and phonology, is largely coextensive with the domain of SPE. Perhaps the appearance of Kaplan & Kay's paper in this collection is symbolic of the recognition that there is a close interplay between computational morphology and phonology.

A second reason why attention to phonology is warranted is that much of phonology is actually *not* subsumed by computational morphology and speech technology. In general, work in these two fields has focussed on SPE-style phonology alone and has not, by and large, connected with current phonological theory or addressed purely phonological concerns. Again, computational phonology should provide usable implementations of more recent models so that they can be incorporated into computational work on morphology and speech.

Finally, one might reasonably ask why a phonologist ought to be interested in computational phonology. At the most obvious level, computational phonology should provide support for developing theories and testing them against data, removing some of the hackwork involved in achieving formal and empirical adequacy. Additionally, computational phonology may be able to provide formal devices that are useful in phonology proper, as in the case of the information-theoretic evaluation metric (Ellison 1993) which is intended to replace the naïve symbol-counting version. One can also observe that phonology has its own divide between theoreticians who work on abstract models which are supported by small collections of data drawn from a wide variety of languages, and investigators working on large scale analyses of individual languages (such as the work of the Summer Institute of Linguistics on the orthographies of approximately 1,100 languages, D. Crozier *pers. comm.*). To this observer, it seems like there could be more communication of new data from the field phonologist to the theoretical phonologist and, in the reverse direction, communication of new hypotheses and useful theoretical devices which would play an active part in the search for interesting new data. It seems plausible that computational systems which let phonologists experiment with large amounts of data and a variety of theoretical models have an important part to play in bridging the gap between the 'theory people' and the 'data people'.

3. Research Themes

I have attempted to identify four strands of work and to cite a representative smaple of work within each. Unfortunately, much valuable, relevant work has had to be omitted from the citation lists below for reasons of space.

Formal reconstruction and language-theoretic results. Work in this area seeks to provide coherent and well-understood formal frameworks in which phonological theories can be expressed. Some work takes an existing theory as its starting point and seeks to refine it and express it in increasing levels of formality, while other work begins from an existing formalism and tries to adapt its expressive capabilities to the needs of phonology. Since most work contains a mixture of both, I shall not attempt a classification. Rather, I shall loosely classify a sample of the work based on the formal method used: **unification** (Carson 1988, Chung 1990, Coleman 1991, Scobbie 1991, Broe 1993, Walther 1993), **predicate logic** (Bird 1990, Bouma 1991, Russell 1993), **modal logic** (Bird & Blackburn 1991, Calder & Bird 1991), **type theory** (Klein 1991, Mastroianni 1993), **categorial grammar/logic** (Wheeler 1981, Dogil 1984, van der Linden 1991, Oehrle 1991, Steedman 1991, Moortgat & Morrill 1993), **finite-state devices** (Kay 1987, Kornai 1991, Wiebe 1992, Bird & Ellison 1994), **electrical circuitry** (Gilbers 1992) and **formal language theory** (Ristad 1990, Kornai 1991, Ritchie 1992, Wiebe 1992).

This work addresses phonological theories such as autosegmental, metrical, underspecification and government phonology. The paper by Kaplan & Kay in this collection is another example of work in this general vein.

Implementations. Work in this area is directed at producing computer programs that can be used by phonologists to develop and test theories. A variety of SPE implementations exist (independently of the finite-state transducer model) starting from (Bobrow

& Fraser 1968) and including models for applying rules in reverse (Bear 1990, Maxwell 1991). Other theoretical frameworks which have been implemented to a greater or lesser extent include **lexical phonology** (Williams 1991), **autosegmental phonology** (Bird 1990, Albro 1994, Bird & Ellison 1994), **diachronic phonology** (Hewson 1974, Eastlack 1977, Lowe & Mazaudon 1989), **inheritance-based models** (Daelemans 1987, Reinhard & Gibbon 1991) and **connectionist models** (see the next paragraph on learning). The paper by Lowe & Mazaudon in this collection is an example of other work under the heading of implementations.

Automatic learning. This work aims to provide models to (i) simulate human behaviour and test models of human language acquisition, and (ii) provide the working phonologist with useful generalisations about a certain body of data under study. Examples of the first type are (Lathroum 1989, Touretzky & Wheeler 1993, Gupta & Touretzky 1992, Hare 1990, Gasser & Lee 1990, Gasser 1992, Shillcock et al. 1992, Goldsmith 1993, Larson 1992), and these all use connectionist models. Examples of the second type are all symbolic (Johnson 1984, Dresher & Kaye 1990, Ellison 1993, forthcoming, Bird 1994). Daelemans, Gillis & Durieux have contributed a paper to the present collection which fits into this category of automatic learning.

Interfacing to grammar and speech. The final grouping contains work which is intended to integrate computational models of phonology with computational models of grammar and of speech. Concerning the phonology-grammar interface, all this work is covered under the paragraph on formal reconstruction above. The assumption is that if phonological models are formalised and employ the same computational model as is used for computational syntax and semantics, interfacing to grammar ought to be relatively straightforward. Another instance of this work is the contribution to the present collection by Bird & Klein. Recent work on integrating phonology with speech synthesis includes (Hertz 1990, Coleman 1992, Dirksen 1992), and there is also a large literature on the phonology of intonation as it relates to synthesis (e.g. Anderson et al. 1984, Ladd 1987).

This concludes the discussion of the various current research themes in computational phonology. As chance would have it, each of these themes is manifested by one of the papers in the present collection. We now go on to briefly survey these papers. The reader is referred to the commentaries for more detailed overviews of the contributions.

4. Brief Survey of Contributions

These papers are given in the same order as the categories of the previous section and in the order in which they appear in the collection itself.

Kaplan & Kay: Regular models of phonological rule systems. Without question the flagship paper of this collection, Kaplan & Kay have finally provided the "widely cited but notoriously unpublished work" (Ritchie et al. 1992:20) which establishes the mathematical foundation for finite-state computational phonology and morphology.

Lowe & Mazaudon: The Reconstruction Engine: a computer implementation of the comparative method. This paper presents an implementation of a technique from diachronic linguistics, known as the comparative method, for comparing word forms taken from cognate languages in order to reconstruct aspects of the ancestor language from which the languages are derived. The system is applied to data from a group of Tibeto-Burman languages spoken in Nepal.

Daelemans, Gillis & Durieux: The acquisition of stress: a data-oriented approach. This paper consists of a rather striking demonstration that an empiricist learning model actually performs better than the nativist 'Principles and Parameters' approach, concerning the task of assigning primary stress to a corpus of around 5,000 Dutch words.