Easier Dictionary Search
Anne David, PhD, Michael Maxwell, PhD

Executive Summary

The Problem
Language analysts must often work with languages in which they are not experts, and they frequently rely on dictionaries for interpretation. But how do you find a word in the dictionary when you don’t know which form of the word to look up, or how to spell it? This problem is particularly acute for languages like Bengali, Urdu, and Pashto, where the rules of word formation are more complicated than English, or where the spelling is not obvious from the sound.

The Solution
Computational tools can help analysts—particularly analysts with limited knowledge of the language—solve this problem of looking up words. At the University of Maryland Center for Advanced Study of Language (CASL), we have created morphological parsers for the Bengali and Urdu languages, and are now working on a parser for Pashto. A morphological parser is a computational tool which analyzes a word into its meaningful parts and relates this form of the word to the parts that make up its meaning. For example, a morphological parser for English would analyze the word *cats* into *cat* plus the plural suffix –s.

This paper describes four key elements of our solution:

- Grammar writing, supporting the building of robust morphological parsers.
- Enhancements to electronic dictionaries.
- Automatic “smart” spell correction.
- Safeguards against software obsolescence.

While we believe we have developed a good approach for the present, the vast numbers of languages for which one would like dictionary lookup aids means that this method must be supplemented by other techniques, such as the development of a way to out-source the work.

The Project

Purpose
People working with a language in which they are not fluent must often resort to looking up words in a monolingual or bilingual dictionary. Depending on the language and the available dictionaries, finding the word can be problematic. One reason is that the language may have complex orthography—spelling rules—as with English. Another common problem with dictionary lookup is determining what form of the word to look for. In general, dictionaries list only one head word, called the citation form, for a given word. For example, an English dictionary will have *rake* as the citation form in its entry for that verb, while the verb’s other
inflected forms, *rakes, raked* and *raking*, if they appear at all, will be included only in the entry for *rake*, rather than having their own entries further down the page.

But English rules for forming words, or morphology, are simple. Significant inflectional morphology, as is found in languages such as Pashto, can present more serious difficulties for finding inflected words in the dictionary. Linguists who are fluent and experienced in a language can surmount these problems. Often, however, linguists may have completed language training recently, or been called on to work in a related language they don’t know well, perhaps because of a surge situation. As a result, they may not be able to easily determine a word’s citation form.

Fortunately, there are tools that find the citation form of an inflected word; these tools are called morphological parsers. A morphological parser is an automatic tool that breaks up an inflected word into its morphemes. One of these morphemes will be the stem, from which the citation form of the word can be produced and looked up in an electronic dictionary. The English word *reruns*, for example, would parse into three morphemes: the prefix *re–*, the stem *run*, and the suffix *–s*.

**Findings**

**Grammars and Parsers**

In our project, our main objective is to build a morphological parser to help with dictionary lookup. In support of this goal, we write two kinds of grammars: the first is a traditional descriptive or reference grammar, written in English prose by a linguist, and intended to be read by linguists. The other is a formal grammar, written in a formal specification language by a computational linguist, and intended for conversion into the programming language of a parsing engine. These two grammars complement each other. For example, descriptive grammars can be—and generally are—ambiguous, or leave important facts undescribed. While human readers may overlook such problems, computers are less tolerant; it would be impossible to build an accurate parser from such a grammar without first fixing the errors. By building the formal grammar and parser at the same time as we write our descriptive grammar, we fix the problems as we go. The result is a refined descriptive grammar and an accurate parser.

Validating the accuracy of these parsers ensures that language analysts can rely on them for accurate performance in dictionary lookup. We tested each parser for its ability to parse words taken from two sources: the examples in our descriptive grammar, and a collection of annotated text. Analysis of the results indicates that many of the errors arise from problems with the electronic dictionaries used by the parsers, which led us to an additional (and initially unanticipated) task, that of revising the electronic dictionaries; we describe this task below.

Our team is now writing descriptive and formal grammars for Pashto, a language which presents its own challenges: a lack of descriptive resources, and underdocumented dialectal variation. We expect to face these challenges in part through actual field work with native speakers of various dialects of Pashto. Fortunately for us, there are many Pashto speakers in this region. The result of this fieldwork will be not only a published grammar which covers multiple dialects, but a better understanding of Pashto dialectal variation for other linguistic researchers.
Dictionary Enhancements

The importation of printed dictionaries into electronic form often introduces errors which impede electronic searches. These problems arise chiefly due to the differences in the representation of dictionaries on paper and in electronic form. Many of these errors turned up during testing of the parser, so our team has also taken on the task of electronic dictionary enhancement.

Some of these errors have to do with the way characters are represented inside the computer. While the use of Unicode systematizes the representation of characters, there are still choices which must be made, and the choices were not always made consistently in the dictionaries we worked with.

Other problems arise when information needed for the parser is extracted from the electronic dictionary. Part-of-speech information, for example, may be represented inconsistently in print dictionaries. While this causes little trouble for human users of the dictionary, the grammatical information must be normalized for the parser. Finally, some of the information in print dictionaries was simply lost when the dictionaries were imported into electronic form, and must be replaced.

Spell Correction

In working with potential users of our parsers, we have found it necessary to also provide a spell correction capability. This need is due both to problems that non-native speakers have in hearing unfamiliar sounds or sound distinctions in the languages we are working with, and to the fact that there are multiple letters for the same sounds in some of these languages. The use of multiple letters for a single sound is similar to the use of ‘ph’ and ‘f’ in English to represent the /f/ sound. This problem is rampant in languages like Urdu due to a large number of words borrowed from Arabic and Persian, and spelled etymologically even though some of the sound distinctions made in Arabic or Persian are not made in Urdu.

However, spell correction systems work well when they can suggest a small number of correctly spelled words in place of a misspelled word. In part because many vowels are usually left unwritten in languages with Arabic-based scripts, the commonly used techniques for spell correction (simple edit distance algorithms) would result in an excessive number of suggestions. Accordingly, we use a more linguistic method for suggesting spell corrections. Based on similar work done at CASL for Arabic, we are implementing spell correctors which base suggested corrections on known “confusions.” The confusions can be due to sounds which are difficult for native English speakers to hear, keys which are adjacent on the keyboard, or letters which are easily confused (particularly letters which differ only in the presence of dots above or below).

Software Obsolescence

The remarkable pace of change with computers has a downside: software which once worked may cease to work on newer systems, and data which was accessible under previous systems may not be interpretable by newer software. If we were to simply build parsers with today’s software, someone would need to re-write our grammars in ten or twenty years. We conceived our project with exactly this problem in mind. Our formal grammars are not written in a particular parsing engine’s programming language, but are instead encoded in XML, a formalism for data interchange. In order to convert the formal grammars into parsers, we have written a
converter program which reads in the XML grammar, and outputs the result in the programming language of the parsing engine which we are using. When this parsing engine is superseded by another engine with a different programming language—as it is certain to be—the only change that will be necessary is to rewrite the half of the converter program that outputs the parsing engine’s programming language, and all the formal grammars that have been written using this formalism can then be re-purposed for the new parsing engine.

The other part of this obsolescence problem is that when it comes time to revise our converter program to support a future parsing engine, someone will need to understand what the formal grammar constructs mean—and formal grammars are notoriously difficult for people to understand. Our use of parallel descriptive and formal grammars was explicitly intended to support human understanding of the formal grammar; by embedding the formal grammar into the descriptive grammar, the explanation of what each piece of the formal grammar means is immediately available to the reader.

Long-Term Methods
While our approach is attractive for the reasons outlined above, the scale of the problem—the hundreds of languages for which dictionary lookup aids are desirable—means that our methodology must in the long run be supplemented by other techniques. One such technique which we plan to explore is that of out-sourcing the development of the descriptive or formal grammars (or both). We believe there are many researchers or research organizations which can, with technical and linguistic assistance, do the work. Quality control is an obvious issue, but one which we believe can be surmounted using modern electronic communications.

Relevance
Linguists working with a language in which they lack fluency face a daily challenge with dictionary lookup. Using our principled approach of building morphological parsers based on parallel grammars, we are creating tools that will help linguists, while developing methods that will make such linguistic tool building more efficient and cost-effective. We are also making word search easier through spell correction, enhancing the underlying electronic dictionaries, and safeguarding against obsolescence of our tools. The payoffs include not only more efficient and capable linguists, but also a methodology that can be used to build grammars and improve language analysis tools for other languages in the future.