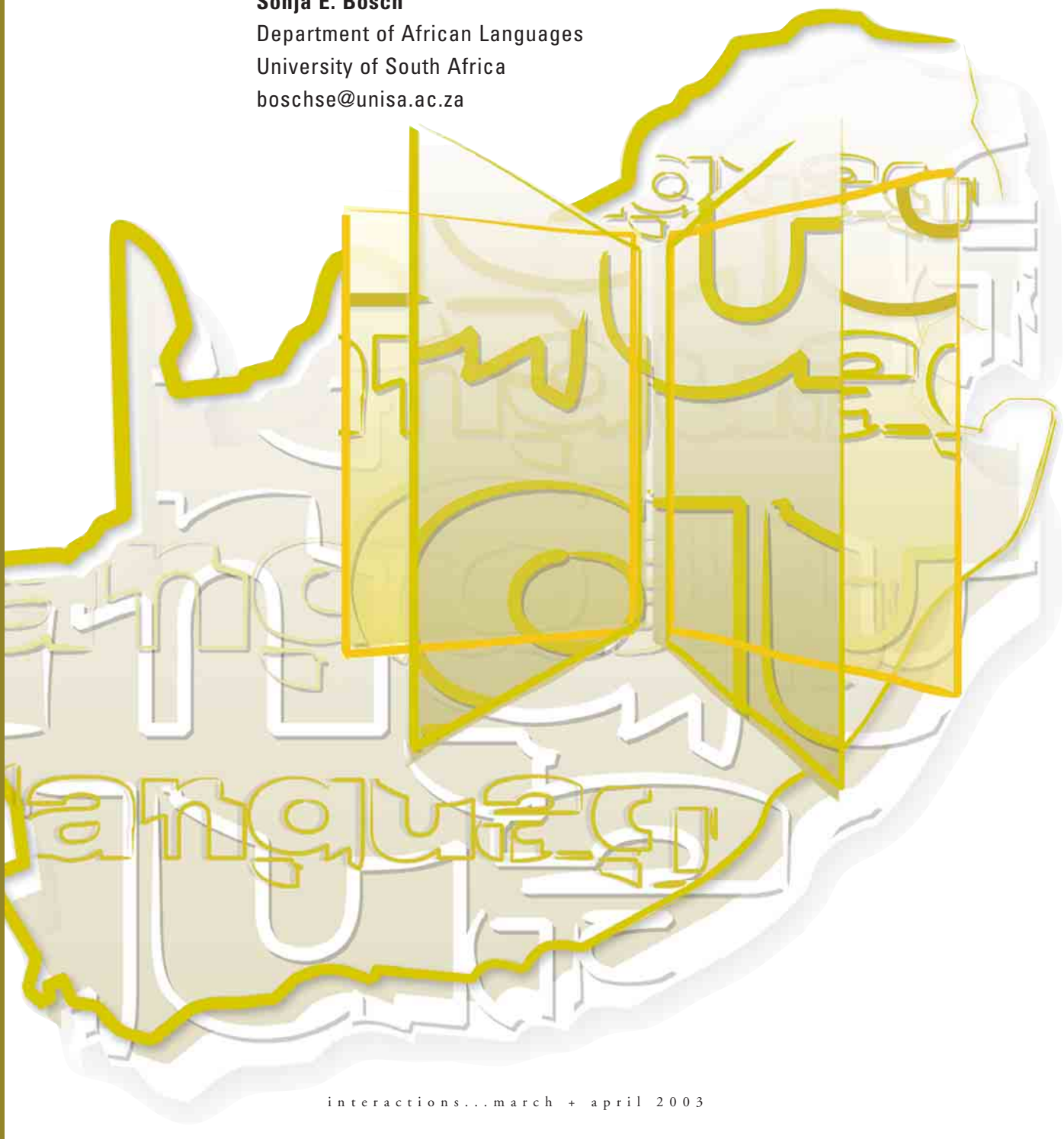
Enabling Computer Interaction in the Indigenous Languages of South Africa: The Central Role of Computational Morphology

Laurette Pretorius
Department of Computer Science and Information Systems
University of South Africa
pretol@unisa.ac.za

Sonja E. Bosch
Department of African Languages
University of South Africa
boschse@unisa.ac.za



Introduction

Ubiquitous computing, the Web, and the ever-increasing processing power of computers have made the study of human-computer interaction (HCI) and the design of intelligent human-computer interfaces fields of crucial importance. If we further assume a human-centered approach to interface and systems design, then human preferences for modality of interaction become increasingly important. Moreover, given that humans interact and communicate most easily and effectively by means of natural language, either spoken or written (i.e., the auditory and vocal and the visual modalities), we must recognize and acknowledge the fundamental role that natural language plays in HCI. Indeed, humans generally prefer communicating or interacting with computers in natural language, and computers should therefore be able to understand and synthesize it.

All people, even the illiterate or semiliterate, are empowered to become part of the information society more readily if they are able to use their own languages. Therefore, languages for which no adequate computer processing is being developed run the risk of being marginalized in the global information society, "or even disappearing, together with the cultures they embody, to the detriment of one of humanity's great assets: its cultural diversity"[9].

The discipline that addresses the design and implementation of computational techniques and computer systems that understand and/or synthesize spoken and written natural language is natural-language processing (NLP). Also included in this discipline are speech processing (recognition, understanding, and synthesis), information extraction, handwriting recognition, machine translation, text summarization, and language generation.

In principle, therefore, HCI and NLP are complementary. However, the exploitation and realization of the potential benefits of NLP relative to HCI largely depend on the availability and continued development of increasingly sophisticated techniques and tools for NLP.

In Africa, with its high degree of illiteracy, and where indigenous languages face the constant threat of marginalization, developing indigenous languages at a technological level is critical. Article 17 of the Cultural Charter for Africa of the Organization of African Unity states:

> *"The African States recognize the imperative need to develop African languages which will ensure their cultural advancement and accelerate their economic and social development...."*

It is also recognized that "there is no alternative to the use of the African languages for literacy and for ensuring mass participation in development"[1].

Linguistic Profile of South Africa

South Africa, with a population of 40.5 million people, is in an unusual position. It is a multilingual country that has more national official languages than any other country. Besides English and Afrikaans, the 11 official languages include the indigenous languages Southern Sotho, Northern Sotho, Tswana, Zulu, Xhosa, Swati,

Ndebele, Tsonga, and Venda. These indigenous languages, all of which share common linguistic features, belong to the Bantu language family. Bantu consists of more than 400 languages spoken on the southern half of the African continent. Figure 1 breaks down the South African official languages as mother tongues [7].

Although English ranks only fifth (nine percent) as a mother tongue, national leaders, politicians, businesspeople, and officials tend to use English more frequently than any other language. In a national survey on language use and language interaction conducted by the Pan South African Language Board (PanSALB) in 2000, only 22 percent of respondents indicated that they fully understood speeches and statements made in English, and 19 percent indicated that they seldom understood information conveyed in English [12].

This finding emphasizes the basic need for mother-tongue communication and interaction (human-human, as well as human-computer) in South Africa, at both the official and personal levels.

Challenges for Natural-Language Processing in South Africa

Given the reality of 11 official languages, the basic need for communication and interaction in the mother tongue, as well as the relatively high illiteracy rate, human-language technologies need to be developed and put in place.

Developing higher-level applications such as voice access to information systems and machine-aided translation systems is one challenge. Such higher-level applications will play an important role, for instance, in the e-Government Gateway project in South Africa, the purpose of which is delivery of services to citizens through a single service point. The aim of the project is to provide each citizen access to electronic services in his or her preferred mode and language, using both text and/or speech.

A second challenge is developing electronic lexicons, which are essential aids in any language training or automated language interaction. By electronic lexicon we mean a lexicography knowledge base from which diverse types of information can be extracted, usually by means of appropriate user interfaces. Electronic dictionaries could play a central role in improving literacy, especially for users with emerging literacy and little familiarity with dictionaries.

Such lexicons are particularly significant for the indigenous languages of South Africa. In most of these languages looking up a word in a conventional dictionary requires a certain degree of grammatical (morphological) knowledge because words are usually not simply listed by their first letter. This is one of the main reasons why any kind of user-friendly and usable electronic dictionary for the indigenous languages of South Africa is not easily available.

Critical Importance of Computational Morphological Analysis

Meeting the two aforementioned challenges depends on the availability of tools to perform computational morphological analysis. In the case of higher-level applications, computational morphological analysis serves as an enabling technology, in other words, a tech-

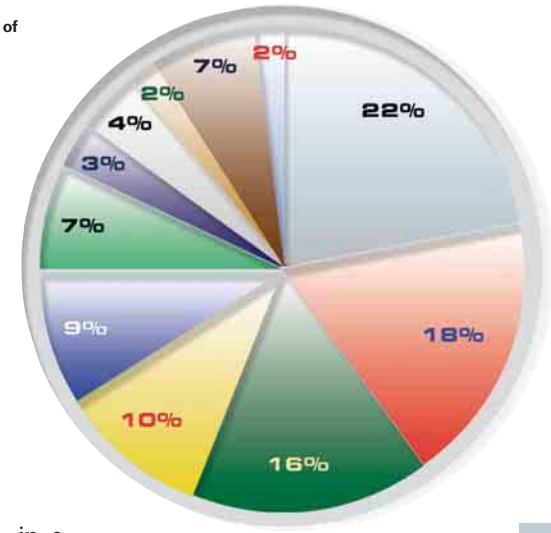


Figure 1. Distribution of speakers of mother tongues in South Africa, by official language (40.5 million speakers).

nology that facilitates the development of further tools and practical applications, including part-of-speech tagging, parsing, text-to-speech systems, information extraction, and machine translation. For electronic dictionaries the computational morphological analyzer provides the missing link between the user who lacks the grammatical knowledge necessary to find a word and the complexity of the dictionary format of the languages belonging to the specific language family.

Computational aids for morphological analysis exist for many European languages, including English, French, German, Spanish, Portuguese, and Italian. Significant work has already been done for Basque, Turkish, Arabic, Finnish, Swedish, Norwegian, Danish, Irish, several Eastern European languages (for example, Hungarian), and Swahili. However, morphological analyzers still need to be developed for the commercially less important languages of the world [9]. Among these languages are those belonging to the Bantu language family, which have not yet received much attention in terms of natural-language processing.

As mentioned earlier, the indigenous languages of South Africa are characterized by their complex morphological structure. Compared with a language such as English, for instance, which has a relatively limited variation of word forms, Bantu languages are quite different. These languages are mainly agglutinating, which means that they extensively use prefixes and suffixes to form words.

In the Bantu languages, the basis for constructing a noun is the root. The root is the constant core element in words or word forms, and the rest is inflection and derivation. Therefore, in a dictionary one would look up the root. Let us look at the complex nature of the monosyllabic noun root *-zi*, meaning "village." The root can appear in various forms within nouns, for instance:

umuzi "village"
imizi "villages"
emzini "in the village"
umzana "small village"
umuzikazi "large village"

In order to look up any of the foregoing nouns in a dictionary the user needs to know the root of the word. In the noun *umzana*, for instance, it is not at all obvious that the root is identical to that of all the other words, namely *-zi*. Without morphological analysis, identifying the noun root *-zi* in such examples is difficult.

By automating the process of morphological analysis, looking up any word in a language such as Zulu becomes a routine task that requires no specialized grammatical knowledge. In fact, the automatic grammatical (morphological) analysis of a word, as it appears in natural language text, could be included in the electronic lexicon. So, instead of the user's looking up *umzana* under the noun root *-zi*, the full word *umzana* would yield the following output:

u- preprefix class 3
-mu- basic prefix class 3
-zi noun root "village"
-ana diminutive suffix

In order to automate morphological analysis we need to computationally model two linguistic phenomena, namely the following:

1. *Morphotactics*, or word-formation rules, which means that morphemes that make up words cannot combine at random but are restricted to certain combinations and orders. A morphological analyzer needs to know which combinations of morphemes are valid.
   - *English example:* The morphemes *pity* and *-less* may combine to form the intermediate morphophonemic string *pityless*.
   - *Zulu example:* The morphemes *u-*, *-mu-* and *-ngane* may combine to form the intermediate morphophonemic string *umungane*, meaning "friend."

2. *Morphological alternations*, which means that one and the same morpheme may be realized in different ways depending on the environment in which it occurs. Again, a morphological analyzer needs to recognize the correct form of each morpheme.
   - *English example:* The English alternation rule, stating that *y* is realized as *I* when followed by the string *-less*, modifies *pityless* to the correct form *pitiless*.
   - *Zulu example:* The Zulu alternation rule, stating that *-mu-* is realized as *-m-* when followed by a polysyllabic word root, such as *-ngane*, modifies *umungane* to the correct form *umngane*.

Therefore, in order to automate the morphological analysis of Zulu requires computational techniques and tools for modeling the morphotactics, as well as the morphological alternations.

This brings us to the focus of our work, the development of a computational morphological analyzer for Zulu.

Finite-State Computational Morphology

Since the 1950s the mathematically equivalent notions of regular formal languages, regular expressions, and finite-state networks have been the subject of extensive research and applications including circuit design, pattern matching, and text processing. Two of the main reasons for this attention are their mathematical elegance and the efficient implementation possibilities that they offer [see, for example, 2, 3, 8, 13, 14]. These methods are increasingly used in various fields of NLP and form the basis of the modern approach to computational morphology [see for example 10, 18].

Research in finite-state computational morphology in organizations such as Xerox Research Centre Europe [18] and AT&T Laboratories [5] is based on the fundamental insight that the complexities of word-formation rules as well as morphological alternations can be modeled and implemented extremely efficiently using finite-state networks.

The Xerox finite-state calculus [6] is a powerful, sophisticated, state-of-the-art set of algorithms and programming languages for building finite-state solutions to a variety of problems in natural language processing. The Xerox software tools we used to build a morphological analyzer for the Zulu language are briefly described as follows:

- The purpose of the **lexc** tool, which stands for for lexicon compiler, is to specify the required and essential natural-language lexicon, as well as the morphotactic structure of the words in the lexicon. The resulting finite-state network produced by **lexc** generates morphotactically well-formed, but rather abstract, morphophonemic or lexical strings.
- The purpose of the **xfst** tool is to formulate the alternation rules necessary for mapping the abstract lexical strings into properly spelled surface strings of natural language, using regular expressions. These regular expressions are then also compiled into a finite-state network.
- Finally, the **lexc** and **xfst** finite-state networks are compiled, or composed, into a single network, called a lexical transducer. The lexical transducer contains all the morphological information about the language being analyzed, including derivation, inflection, alternation, compounding, and so forth and constitutes our computational morphological analyzer. In our examples a morphological analyzer for English would map the morpheme sequence *pity* + *-less* to the string *pitiless*, and the Zulu morphological analyzer would map the morpheme sequence *u-*, *-mu-*, and *-ngane* to *umngane*.

It should be emphasized that once the morphotactics and alternation rules of the language have been correctly specified, the morphological analyzer can recognize and analyze only words of which the roots have been explicitly included.

In order to systematically update and extend the root list of the morphological analyzer, the Xerox finite-state tools allow you to build a so-called guesser, a variant of the morphological analyzer that contains all phonologically possible roots [18]. The guesser variant of the morphological analyzer is a particularly useful computational tool for exploring (new) language corpora. By applying the guesser to any corpus as a potential source of new word roots, new (that is, as yet unlisted) word roots are detected, analyzed, and marked for possible inclusion in the current word root list. These new word roots are then scrutinized by the human lexicographer. The suitable and correct word roots are then added to the word root list of the morphological analyzer so that any future occurrence of such a root will be recognized and appropriately analyzed.

In a nutshell, our *ultimate purpose* is to model the morphological structure of Zulu in such a way that *all the Zulu words are included* (generated) and *correctly*

The development of **interfaces** for all the **indigenous languages** of South Africa constitutes a **major challenge** for the future.

analyzed, and that *all character strings* that do *not* represent words in the *real language* are *excluded*.

In other words, the morphological analyzer for Zulu comprises

- *A comprehensive list* of Zulu word roots
- *All the word formation rules* (morphotactics) that apply in the language.
- *All the alternation rules* required to produce well-formed words in the Zulu language.

What does this mean in terms of the dynamic changing nature of natural language? First, it means that the morphological analyzer reflects the *stable part of natural language*, that is, the morphological structure, which includes both the morphotactics and the alternation rules. Second, it reflects the *growth and dynamic nature of natural language*, that is, it contains a comprehensive, current list of word roots that should be systematically extended and enriched as new words are created and come into use.

Thus, the availability of a morphological analyzer and a guesser variant of the analyzer, together with a machine-readable lexicon, provide us with the tools for systematically and scientifically exploring the available Zulu text corpora, thereby representing complete, up-to-date lexical information in a format that is accessible to other applications in Zulu NLP.

Conclusion

Our project may be extended in various ways. Future work in natural-language processing includes developing similar tools for other indigenous languages of South Africa and using the tools to build higher-level NLP applications for these languages. Such applications may range from sophisticated high-level machine translation systems to voice-operated educational or commercial systems that can be used by illiterate people, or from applications in education and training to public service (e-governance) and e-commerce applications.

In order to make the lexical information embodied in the computational morphological analyzer accessible to humans as a Zulu electronic dictionary, a novel and versatile human-computer interface needs to be developed. Indeed, the study, design, and development of language and culture-specific lexical and language interfaces for all the indigenous languages of South Africa, and for the whole of Africa, constitute a major challenge for the future.

One exciting development involves the strategic plan for the development of human-language technologies (HLT) in South Africa [17], commissioned by the South African government, which sets out to facilitate the implementation of NLP applications such as those mentioned earlier in order to promote multilingualism and to develop previously marginalized indigenous languages.

A number of project teams are working on the development of human-language technologies in South Africa.

A number of project teams are already working in development in the field of HLT in South Africa. The African Speech Technology (AST) project is the first of its kind in South Africa. It concerns a multilingual, telephone based, information retrieval system, supported financially by the Department of Arts, Culture, Science and Technology of the South African government [4]. Other projects include the development of language corpora and spell checkers for the indigenous languages of South Africa [11], the promotion and development of multilingual terminologies [16], and our project on the computational morphological analysis of Zulu [15] and Xhosa, together with the development of machine-readable lexicons for Zulu and Xhosa.

It is our collective vision that these and similar developments will empower the people of South Africa, as well as the rest of Africa, to actively participate in the continent's economy and to become part of the information society by providing optimum use of multilingual information and communication technology with easy access to information in the most natural way—that is, through language and speech.

ABOUT THE AUTHORS

Laurette Pretorius has a D.Sc. in applied mathematics from the University of Potchefstroom, and is a Professor of Computer Science at the University of South Africa, where she teaches Numerical Methods, Computer Graphics, Automata theory and Formal Languages, and Computability Theory. She is particularly enthusiastic about multi-disciplinary research, in particular natural language processing, and also the social and ethical aspects of computing.

Sonja Bosch holds the DLitt et Phil from the University of South Africa where she is presently Associate Professor in the Department of African Languages. She is currently leading a computational morphology project, involving three Bantu languages. She is also Convenor of the Special Interest Group for Language and Speech Technology Development in South Africa and a Member of a Ministerial Advisory Panel on Human Language Technologies.

REFERENCES

1. *ACALAN: African Academy of Languages Special Bulletin.* January 2002. Director of Publication: Adama Samassekou, acalan@malinet.ml., pp. 14 & 24.
2. Aho, A.V. Algorithms for Finding Patterns in Strings. In Van Leeuwen (ed.), *Handbook of Theoretical Computer Science. Volume A: Algorithms and Complexity*, Elsevier, Amsterdam, 1990, pp. 255-300.
3. Aho, A.V. and Ullman, J.D. *The Theory of Parsing, Translation, and Compiling. Volume 1: Parsing.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1972.
4. African Speech Technology (AST) Project. Available at www.ast.sun.ac.za (accessed on 11/4/2002).
5. AT&T Laboratories. AT&T Labs-Research, FSM Library. Available at www.research.att.com (accessed on 11/4/2002).
6. Beesley, K. and Karttunen, L. *Finite-State-Morphology: Xerox Tools and Techniques.* CSLI Publications, Stanford, CA, forthcoming.
7. Bosch, S.E. and Roux, J.C. HLT Initiatives in South Africa-Implications for Human Development, Empowerment and Democratization. In *Proceedings of the International Symposium: Text in Context: African Languages between Orality and Scripturality.* University of Zurich. Rüdiger Köppe Verlag, Cologne, forthcoming.
8. Cohen, D.I.A. *Introduction to Computer Theory.* John Wiley & Sons, Inc., New York, 1997.
9. Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V. *Survey of the State of the Art in Human Language Technology.* Cambridge University Press, Cambridge, 1997, pp. 16.
10. Dale, R., Moisl, H., and Somers, H. *Handbook of Natural Language Processing.* Marcel Dekker, Inc.. New York, 2000.
11. ELC for ALL. Available at www.up.ac.za/academic/libarts/afrilang/spellcheckers.htm (accessed on 11/4/2002).
12. Language Use and Language Interaction in South Africa. A Sociolinguistic Survey. Pan South African Language Board Occasional Paper, 2000.
13. Lewis, H.R. and Papadimitriou, C.H. *Elements of the Theory of Computation.* 2nd ed. Prentice-Hall International, Inc., London, 1998.
14. Perrin, D. Finite Automata. In Van Leeuwen (ed.)., *Handbook of Theoretical Computer Science. Volume B: Formal Methods and Semantics.* Elsevier, Amsterdam, 1990, pp. 1-57.
15. Pretorius, L. and Bosch, S.E. Finite-State Computational Morphology-Treatment of the Zulu Noun. *South African Computer Journal 28* (2002), pp. 30-38.
16. Terminology Coordination Subdirectorate. Available at www.dac.gov.za/about_us/cd_nat_language/terminology/terminology.htm (accessed on 11/4/2002).
17. The Development of Human Language Technologies in South Africa: Strategic Planning. Available at www.dac.gov.za/about_us/cd_nat_language/language_planning/hlt_strategic_plan/hlt_strategic_plan2.htm (accessed on 11/4/2002).
18. XRCE MLTT: References to Finite-State Methods in Natural Language Processing. Available at www.xrce.xerox.com/research/mltt/fst/fsrefs.html (accessed on 11/4/2002).