Doing Lushootseed morphology by analogy

Deryle Lonsdale
Brigham Young University

Analogical modeling (AM) is an exemplar-based general modeling theory that is being applied to an increasing range of natural language processing problems. This paper introduces AM as a viable approach to morphological analysis for Lushootseed, and shows results from applying the system to analyze the contents of the transcription of a well-known Lushootseed story. Subsequent discussion mentions the strengths and current weaknesses of the approach. Possible improvements and future applications are also sketched.

1 Lushootseed and morphological processing

Lushootseed (along with the other Salish languages) exhibits a high degree of morphological complexity. Derivational and inflectional affixation is pervasive, partial reduplication is common, and instances of unusual morpheme types (e.g. lexical suffixes) occur abundantly. Consequently, words can often be very complex, consisting of a handful or more of juxtaposed morphemes.

Describing word structure can be a difficult task for such languages. Grammars traditionally present complex words with their constituent morphemic structure diagrammed or at least broken out in template form. Many language texts and readers also present words with their morphemic composition indicated and sometimes with English glosses for the morphemes in question.

Morphological analysis or parsing consists of breaking down a word into its constituent morphemes as shown above. The set of decisions used in parsing a word can sometimes be open to discussion: whereas almost everyone would agree that the English word "dogs" can be parsed into the singular "dog" and a plural morpheme "-s", determining an exact boundary for words such as "hysterically" may be challenging. Indeed, often other linguistic areas (e.g. phonetics, phonology, and even syntax) are implicated in the determination of which morphemes make up a word. Clearly this process requires knowledge of the language's grammar and vocabulary, of morphological principles, and of other areas of linguistics that play a role in morphology.

Given that this process is already difficult for humans, how reasonable is it to expect that computers could handle the task of morphological parsing? Fortunately much research has been carried out in this area, with often impressive results. Several approaches have been implemented to carry out morphological analysis. One of the most successful involves finite-state techniques borrowed from the field of computer science. In a finite-state approach to morphological analysis, linguistic rules describing phonological and morphological variation are established, lexicons specifying possible morphemes are supplied, and constraints are developed describing how words are formed. This information is compiled into finite-state transition tables from which automata can be built. These automata analyze an incoming word letter-by-letter, validating the input against the morpheme lexicons and any variational changes specified by the rules. A wide variety of languages, ranging from morphologically straightforward to complex, have been the target of finite-state implementations. Recent work has shown that these techniques work well for Lushootseed (Lonsdale, 2001) and other Salish languages (Lonsdale, 2003).

One problem with finite-state methods for the morphological analysis of a language is that providing the data is an extremely complex process requiring knowledge of linguistics, computational morphology, and computational techniques. Such approaches are thus appropriately called "knowledge-based" because of the extensive levels of knowledge required. A "knowledge acquisition bottleneck" arises when systems cannot be developed fast or accurately enough because of the inherent complexity of the knowledge and data required by the system and its developers.

Recently, researchers have studied ways of getting around the knowledge acquisition bottleneck for morphological processing. The Boas project (Oflazer et al., 2001) allows two people—a linguist and a language informant—to develop knowledge sources by answering the queries of a specially-designed interactive system. The system acquires and learns certain aspects of a language's structure (including morphology) by testing hypotheses it develops based on the two experts' input. The two humans accept or reject rules developed by the system, and their input is used in further investigation. The end result is a semiautomatically-generated finite-state engine for the language in question.

Other machine learning techniques have also been used to guess where a word can be broken into its constituent morphemes—a process often called morphological boundary identification or morpheme discovery. Some approaches induce boundaries by comparing inflected words with their uninflected root counterparts (Theron and Cloete, 1997), with other related inflections (Baroni et al., 2002), with lexically-specified morphemes (Sharma et al., 2002), or with word lists (Snover et al., 2002; Neuvel and Fulop, 2002). Other work leverages various technical algorithms such as expectation minimization (Peng and Schuurmans, 2001), minimum description length (Goldsmith, 2001), maximum likelihood (Creutz and Lagus, 2002), latent semantic indexing (Schone and Jurafsky, 2000), memory-based learning (van den Bosch et al., 1996), and genetic algorithms (Kazakov, 1997).

Most of these approaches require annotated human input consisting of several instances or exemplars indicating where the morpheme boundaries occur. Some also require morpheme lexicons for beginning or bootstrapping the process of morpheme identification. From this foundation of information a typical system tries to arrive at appropriate procedures for positing morpheme boundaries. Such approaches are called supervised ones since a human provides some of the initial data. Unsupervised approaches, where no annotated data is used, are also being addressed in current work (including some of the references cited above) , but the issues are too technical to be considered here.

2 Analogical modeling

This paper proposes another approach to morpheme boundary discovery: analogical modeling (AM). AM has not been applied to this type of problem, though it has been very successfully used in other language modeling problems involving lexical selection, phonology, and morphology (Skousen et al., 2002). Analogical modeling is a data-driven, exemplar-based approach to modeling language and other types of data. It has no rule-based component, either explicit or implicit, requires no explicit knowledge representations beyond the set of exemplars, and is more flexible and robust than many traditional approaches to language modeling. Several linguistic applications have been reported using analogical modeling as the basic approach including Spanish diminutives, Danish compounds, Turkish morphophonemic alternations, Arabic lexical selection, and Finnish verb tense formation.

The system operates as follows. A set of exemplars that address and illustrate a particular linguistic issue is prepared; each instance has a fixed-length feature-vector encoding that represents salient (and perhaps nonsalient or questionable) properties for that instance. Each instance is labelled with an outcome that is used by the system to analyze how that instance behaves with respect to the issue in question. At run time, the user first inputs into the system the set of exemplars with their outcomes. Then the user inputs one or more queries in the form of a similarly encoded feature vectors. The system matches the input queries with the exemplar base, and generates one or more probabilistically weighted outcomes for each test item.

The system is able to tolerate noisy, contradictory, or incomplete data and computes its results differently from the approaches mentioned above. More details on the system's use in language applications are available elswhere (Skousen, 1989); the statistical foundations and processing metrics (Skousen, 1992) are also beyond the scope of this paper.

3 The approach

This section reports on a series of limited experiments carried out to demonstrate the use of AM for predicting morpheme boundaries of various categories. A sketch is given of the basic problem, where the data was obtained from, and how the input files were prepared.

The basic strategy used in these experiments is to encode the instances in such a way that analogical effects can be seen and leveraged across data instances. For example, the prefix "łu-" is an aspectual prefix used to denote an irrealis or future. In morphologically analyzing an utterance such as "łupətidg^{w}əsbid čəd" *(I'll think about it)*, a person would create a morpheme boundary between the prefix "ɬu-" and the root "pətid". Of course, not all words beginning with this two-letter sequence employ it as a prefix: "ɬuʔum" is a root meaning *dog salmon*, "ɬub" is a root meaning *to feed*, "ɬuyab" means *become scared, afraid*, "ɬuk^{w}aɬ" means *sun*, and so on. Thus the two-letter sequence "ɬu-" may or may not be followed by a morpheme boundary, depending on the surrounding context (e.g. other letters and word boundaries). Crucially, the decision about whether this position separates different morpheme can be made on the basis of certain cases that are already clearly known. Complicating the process is the (commonly occuring) situation that the morpheme ɬu- actually has variant forms: "ɬə-" and "ɬ-" also occur in certain phonological (or orthographic) environments.

3.1 Exemplar instances

In using analogical modeling, the representation of data instances is an important consideration. This decision centers around how many characters should be used to encode each feature of a given data instance, and what linguistic features they represent.

For this work, a minimal representation was tried first: a simple vector of 10 single characters closely tied to the orthography of the word in question. Thus each feature in the data instance is filled by a letter or letter/phonological feature combination. For example, note this example of a data instance which has an outcome (the character 0), and which uses ten features encoded in a vector, one letter character per feature:

```
0 ==pastEd==
```

This instance has a vector representing the word "pastəd" (*white person*, romanized as `pastEd`). Since the vector needs to be 10 characters long, it is padded by equal signs (which represent a null value). The vector is preceded by its outcome, `0`, which means that between the letters `s` and `t` in the vector there should be no morpheme boundary.

For use as its instance base, the system was given about 250 Lushootseed words of varying complexity with their morpheme boundaries already identified. This exemplar base was obtained by taking all of the subentry heads (e.g. derivations, reduplications, and forms with lexical suffixes formed from main entries) from one portion of the definitive Lushootseed dictionary[1] (Bates et al., 1994). The portion of the dictionary where these words were taken from is the section listing all words beginning with the first letter of the alphabet: the glottal stop. This section is 22 pages long, or just less than 10% of the alphabetic section of the printed dictionary.

Each word was first converted to ASCII-based romanization using, for example, E for schwa, W for labialized secondary articulation, ? for the glottal stop, | for a stress mark, and X, S and C for the x-wedge, s-wedge, and c-wedge

[1] Hereafter referred to as "the dictionary"

```
^ =====?|uXW          ^ =====?absS
0 ====?|uXWt          0 ====?absSa
0 ===?|uXWtx          0 ===?absSad
0 ==?|uXWtxW          0 ==?absSadE
0 =?|uXWtxWy          * =?absSadEb
- ?|uXWtxWyi          0 =?absSadEb
0 |uXWtxWyic          0 ?absSadEb=
0 uXWtxWyic=          0 absSadEb==
- XWtxWyic==          - bsSadEb===
0 WtxWyic===          0 sSadEb====
- txWyic====
```

Figure 1: This is a set of vectors. The vectors on the left represent morphological boundary information for the (romanized) word "?uXWtxWyic", and those on the right specify boundaries for the word "?absSadEb".

respectively. Next, each romanized word was converted to vectors. Vectors were 10 features long, with each feature being represented by one ASCII character. Each vector consists of a 5-character left context and a 5-character right context surrounding a potential morpheme boundary. Since each letter pair in the word must be considered in turn, there are several vectors created for a given word: in particular, n vectors for a word consisting of n letters.

For example, consider the word "ʔux̌^{w}tx^{w}yic" *(take it for me)*; the dictionary gives its morphemic decomposition as "ʔux̌^{w}-tx^{w}-yi-c". Its romanization form is "?uXWtxWyic", and its vectorization is as shown in the left-hand side of Figure 1.

Each vector is preceded by its outcome, a one-character value representing the type of morpheme boundary[2] found between the fifth and sixth positions of the vector. The first vector states that there is no morpheme boundary (hence outcome 0) at the beginning of the word (after five null features and before the letters "?|uXW". The sixth vector, though, specifies that there is a regular morpheme boundary (hence outcome –) between the letter sequences "?|uXW-" and "-txWyi". The ninth vector above specifies a similar morpheme boundary between the sequences "XWtxW" and "yic", and the last vector specifies a morpheme boundary before the final letter "c".

Similarly, the right-hand column of Figure 1 shows the set of vectors for the word ?absšadəb *(take a step)*.

3.2 Testing the model

Tested against itself, the system performed at 100% accuracy. In other words, when the system is tested on data that it has already assimilated, it does not

[2] * represents a lexical suffix boundary, + a reduplication boundary, ^ a root boundary, – a normal affix boundary, and 0 no boundary

make any mistakes. This is to be expected, since the exemplars used in testing are identical to those in the instance base. It is no surprise that when the system's data exemplar base is taken from the first part of the dictionary (the glottal-stop-initial roots), it will perform well on words taken from that part of the dictionary.

A more interesting evaluation of the system's capabilities to perform analogy appropriately is to apply it to new data that does not necessarily constitute the instance base—vectors that the system has not processed in its store of exemplars. This requires the system to evaluate contextual similarities between given and novel data to arrive at a decision about which outcome is more appropriate. In our case, the system would be forced to decide whether, between any two given letters in a given word, a morpheme boundary should be posited, and if so which type of boundary should be used (reduplication, root, normal, or lexical suffix).

Accordingly, a widely-known story from Lushootseed culture was taken to test the system. This test data consisted of the story "Young Mink and Tutyeeka" as told by Edward Sam and subsequently transcribed (Hess, 1995).

Approximately 275 Lushootseed words long, the story contains almost 500 morphemes. Though the morpheme boundaries are not identified in the source text, they were inserted manually based on conventions used in the dictionary. Processing by the vectorization code (which is written in Perl) resulted in a set of just over 1700 test instances corresponding to the story.

The system then processed the test instances by comparing each instance against the set of original exemplars. Statistics were kept on each, and results were output to a file. Figure 2 shows the summary and analysis of the system performance for one such run.

3.3 Interpreting the results

The results shown in Figure 2 show that the system has performed quite well at the morpheme boundary identification task, achieving overall accuracy of almost 89% for the over 1700 decisions it had to make concerning the absence or presence of boundaries in all possible locations. These figures compare favorably with figures obtained for similar tasks in other languages. A vector length of 10 seemed optimal; experiments run with shorter vectors yielded lower accuracy, and longer vectors did not improve performance.

AM's performance is very good when deciding that a morpheme boundary is not present in a given situation (with an accuracy of over 96%), though 2.62% of the time it incorrectly posits a normal morpheme boundary (outcome –) where in fact no boundary appears. Even better is the system's determination of reduplication boundaries (outcome +), with an accuracy of 97.44%.

However, closer inspection shows that the system almost always missed identifying the beginning of a root, with a paltry accuracy of 3.57%. There are three reasons for this. The first reason is that only a small percentage of the language's roots—those from the glottal stop section—were used in the exemplar set. Identification of roots improved dramatically (rising to about 75%) when all

```
Correct prediction made 88.67% (1519/1713) of the time

Test items with outcome 0 were predicted as follows:
96.31% 0        (1436/1491)
 2.62% -        (39/1491)
 1.07% ^        (16/1491)

Test items with outcome - were predicted as follows:
66.09% 0        (76/115)
33.91% -        (39/115)

Test items with outcome ^ were predicted as follows:
92.86% 0        (52/56)
 3.57% -        (2/56)
 3.57% ^        (2/56)

Test items with outcome + were predicted as follows:
 2.56% 0        (1/39)
97.44% +        (38/39)

Test items with outcome * were predicted as follows:
66.67% 0        (8/12)
33.33% *        (4/12)
```

Figure 2: Sample output results from an experiment. 1713 test instances were evaluated against the exemplars with varying success for each type of morpheme boundary; overall accuracy was 88.67%.

of the dictionary's headwords (i.e. roots, primarily) were added to those from the glottal stop section. Another reason for poor preliminary performance on roots is that the dictionary did not identify the beginning of a root for words that had no prefixes: the usual square root sign (romanized here as ^) only appears when a prefix is present. So the exemplar data was missing most of these instances. The third reason is that the test text (the mink story) was annotated following this same practice—only adding ^ when prefixes were present. Subsequent experiments were performed where the exemplar data (i.e. the dictionary subentry terms) was given an explicit square-root sign in all instances, as were the words from the test data (the mink story). The result was dramatic: roots can now be predicted with 90.51% accuracy.

Another difficulty noticeable in Figure 2 is that the system missed two thirds of the instances of lexical suffixes (outcome *). This is probably due to the paucity of exemplar data, and would be substantially improved if a larger portion of subheadings from the dictionary were used to provide exemplars.

Most problematic is that the system was only able to identify just over one third of the normal morpheme boundaries (i.e. the ones that signal aspectual and tense-related affixes, valency suffixes, possessive clitics, and so on). Adjusting the exemplar set in various ways did not improve this situation markably. It is possible that extending the exemplar data beyond one section of the dictionary (the glottal stop section) to include a richer variety of word vectors will improve accuracy. Choosing more linguistically informative features for the vectors, beyond simple romanized orthography characters, will also almost certainly improve results. Clearly more work needs to be pursued in this area.

4 Prospects and future work

If the problems mentioned above can be overcome, AM-based morpheme boundary detection will be a quicker way to develop morphological analysis engines than by employing knowledge-based methods. AM morphology engines could conceivably be used for automatically glossing text or for parsing input for text understanding systems.

Though this work has only considered Lushootseed morphology, AM can be straightforwardly applied to other Salish languages. The approach might also prove useful in other language problems; ongoing work is investigating the prediction of stress patterns in Salish languages via AM.

One disadvantage in using exemplar-based systems is that examples must be readily available. This means that textual material (e.g. lexicons and corpora) must be in machine-readable form and accessible to researchers working in language modeling. Unfortunately, for less common languages like those in the Salish family, appropriate resources are not widely available. Though concerted efforts have been made to centralize and disseminate material in other languages[3], publicly available Salish materials are almost all in printed form. This factor may limit the applicability processing Salish languages using the state-of-the-art research being done in corpus-based and exemplar-based methods.

References

Baroni, M., Matiasek, J., and Trost, H. (2002). Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In Maxwell, M., editor, *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, Philadelphia, PA.

Bates, D., Hess, T., and Hilbert, V. (1994). *Lushootseed Dictionary*. University of Washington Press.

Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In Maxwell, M., editor, *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, Philadelphia, PA.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Hess, T., editor (1995). *Lushootseed reader with introductory grammar, Vol. 1*. University of Montana Occasional Papers in Linguistics. Summer Institute of Linguistics.

[3] See, for example, www.ldc.upenn.edu or www.elda.fr.

Kazakov, D. (1997). Genetic algorithms and MDL bias for word segmentation. In *Proceedings of the European Summer School in Logic, Language, and Information (ESSLLI-97)*.

Lonsdale, D. (2001). A two-level morphology engine for Lushootseed. In *Proceedings of the 36th International Conference on Salishan and Neighbouring Languages*, University of British Columbia Working Papers in Linguistics.

Lonsdale, D. (2003). Two-level engines for Salish morphology. In *Proceedings of the Workshop on Finite-state methods in natural language processing, 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.

Neuvel, S. and Fulop, S. (2002). Unsupervised discovery of morphology without morphemes. In Maxwell, M., editor, *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, Philadelphia, PA.

Oflazer, K., Nirenburg, S., and McShane, M. (2001). Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computational Linguistics*, 27(1):59–85.

Peng, F. and Schuurmans, D. (2001). A hierarchical EM approach to word segmentation. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS-2001)*.

Schone, P. and Jurafsky, D. (2000). Knowledge-free induction of morphology using Latent Semantic Analysis. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 67–72.

Sharma, U., Kalita, J., and Das, R. (2002). Unsupervised learning of morphology for building lexicon for a highly inflected language. In Maxwell, M., editor, *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, Philadelphia, PA.

Skousen, R. (1989). *Analogical Modeling of Language*. Kluwer, Dordrecht.

Skousen, R. (1992). *Analogy and Structure*. Kluwer, Dordrecht.

Skousen, R., Lonsdale, D., and Parkinson, D. B., editors (2002). *Analogical Modeling: An exemplar-based approach to language*, volume 10 of *Human Cognitive Processing*. John Benjamins, Amsterdam.

Snover, M., Jarosz, G., and Brent, M. (2002). Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In Maxwell, M., editor, *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, Philadelphia, PA.

Theron, P. and Cloete, I. (1997). Automatic acquisition of two-level morphological rules. In *Proceedings of the 5th Conference on Applied Natural Language Processing*.

van den Bosch, A., Daelemans, W., and Weijters, T. (1996). Morphological analysis as classification: an inductive-learning approach. In Oflazer, K. and Somers, H., editors, *Proceedings of the Second International Conference on New Methods in Natural Language Processing, NeMLaP-2*, pages 79–89, Ankara, Turkey.