Lexical Processing in the CLARE System

David M. Carter*
SRI International
Cambridge Computer Science Research Centre
23 Millers Yard
Cambridge CB2 1RQ, U.K.

1 Introduction

In many language processing systems, uncertainty in the boundaries of linguistic units means that data are represented not as a well-defined sequence of units but as a lattice of possibilities. This is often the case in speech recognition, syntactic parsing and Japanese kana-kanji conversion. In contrast, however, it is often assumed that, for languages written with interword spaces, it is sufficient to prepare an input character stream for parsing by grouping it deterministically into a sequence of words, punctuation symbols and perhaps other items.

But for typed input, spaces do not necessarily correspond to boundaries between lexical items, because of errors and other, linguistic, phenomena. This means that a lattice representation, not a simple sequence, should be used throughout front end (pre-parsing) analysis. The CLARE system under development at SRI Cambridge uses such a representation, allowing it to deal straightforwardly with combinations or multiple occurrences of phenomena that would be difficult or impossible to process correctly under a sequence representation. This paper concentrates on CLARE's ability to deal with typing and spelling errors, which are especially common in interactive use, for which CLARE is designed.

The word identity and word boundary ambiguities encountered in the interpretation of errorful input often require the application of syntactic and semantic knowledge on a phrasal or even sentential scale. Such knowledge may be applied as soon as the problem is encountered; however, this brings major problems with it, such as the need for adequate lookahead, and the difficulties of engineering large systems where the processing levels are tightly coupled. To avoid such problems, CLARE adopts a staged architecture, in which indeterminacy is preserved until the knowledge needed to resolve it is ready to be applied. An appropriate representation is of course the key to doing this efficiently.

2 Spaces and Word Boundaries

In general, typing errors are not just a matter of one intended input token being miskeyed as another one. Spaces between tokens may be deleted or inserted. Multiple errors, involving both spaces and other characters, may be combined in the same intended or actual token. A reliable spelling corrector must allow for all these possibilities.

However, even in the absence of "noise" of this kind, spaces do not always correspond to lexical item boundaries, at least if lexical items are defined in a way that is most convenient for grammatical purposes. For example, "special" forms such as telephone numbers or e-mail addresses, which are common in many domains, may contain spaces. In CLARE, these are analysed using regular expressions, which may include space characters.

The complexities of punctuation are another source of uncertainty: many punctuation symbols have several uses, not all of which necessarily lead to the same way of segmenting the input. For example, periods may indicate either the end of a sentence or an abbreviation, and slashes may be simple word-internal characters or function lexically as disjunctions.

CLARE's architecture and formalism allow for all these possibilities, and, as an extension, also permit multiple-token phrases, such as idioms, to be defined as equivalent to other tokens or token sequences. This facility is especially useful when CLARE is being tailored for use in a particular domain, since it allows people not expert in linguistics or the CLARE grammar to extend grammatical coverage in simple and approximate, but often practically important, ways.

3 CLARE's Processing Stages

The CLARE system is intended to provide language processing capabilities (both analysis and generation) and some reasoning facilities for a range of possible applications. English sentences are mapped, via a number of stages, into logical representations of their literal meanings, from which reasoning can proceed. Stages are linked by well-defined representations. The key intermediate representation is that of *quasi logical form* (QLF), a version of first order logic augmented with constructs for phenomena such as anaphora and quantification that can only be resolved by reference to context. The unifica-

*CLARE is being developed as part of a collaborative project involving SRI International, British Aerospace, BP Research, British Telecom, Cambridge University, the UK Defence Research Agency, and the UK Department of Trade and Industry.

tion of declarative linguistic data is the basic processing operation.

In the analysis direction, CLARE's front end processing stages are as follows. A sentence is divided into a sequence of *clusters* separated by white space. Each cluster is then divided into one or more *tokens*: words (possibly inflected), punctuation characters, and other items. Tokenization is nondeterministic, and so a lattice is used at this and subsequent stages. Next, each token is analysed as a sequence of one or more *segments*. For normal lexical items, these segments are morphemes. The lexicon proper is first accessed at this stage. Various strategies for *error recovery* (including but not limited to spelling/typing correction) are then attempted on tokens for which no segmentation could be found. After this, edges without segmentations are deleted; if no complete path remains, sentence processing is abandoned. Further edges, possibly spanning non-adjacent vertices, are added to the lattice by the phrasal equivalence mechanism mentioned above. Finally, morphological, syntactic and semantic stages apply to produce one or more quasi logical forms (QLFs). These are checked for adherence to sortal (selectional) restrictions, and, possibly with the help of user intervention, one is selected for further processing.

4 Segmentation and Spelling Correction

English inflectional morphology is sufficiently simple to allow CLARE to use a fairly simple affix-stripping approach to token segmentation. One major advantage of this is that spelling correction can be interleaved directly with it. Root forms in the lexicon are represented in a discrimination net for efficient access. When the spelling corrector is called to suggest possible corrections for a word, the number of simple errors (of deletion, insertion, substitution and transposition) to assume is given. Normal segmentation is just the special case of this with the number of errors set to zero. The mechanism nondeterministically removes affixes from each end of the word, postulating errors if appropriate, and then looks up the resulting string in the discrimination net, again considering the possibility of error.

Interleaving correction with segmentation promotes efficiency in the following way. As in most other correctors, only up to two simple errors are considered along a given search path. Therefore, either the affix-stripping phase or the lookup phase is fairly quick and produces a fairly small number of results, and so the two do not combine to slow processing down. Another beneficial consequence of the interleaving is that no special treatment is required for the otherwise awkward case where errors overlap morpheme boundaries; thus *desigend* is corrected to *designed* as easily as *deisgned* or *designde* are.

If one or more possible corrections to a token are found, they are preserved as alternatives for disambiguation at the later syntactic or semantic stages. The lattice representation allows multiple-word corrections (involving both the insertion and the deletion of spaces) to be preserved along with single-word ones. The choice is only finally made when a sortally coherent QLF is selected.

5 An Evaluation

To assess the usefulness of syntactico-semantic constraints in CLARE's spelling correction, the following experiment was carried out. Five hundred sentences falling within CLARE's current lexical and grammatical coverage were taken at random from the LOB corpus. Although CLARE's core lexicon is fairly small (1600 root forms), it consists of the more frequent words in the language, which tend to be fairly short and therefore have many candidate corrections if misspelled. The sentences were passed, character by character, through a channel which transmitted a character without alteration with probability 0.99, and with probability 0.01 introduced one of the four kinds of simple error. This process produced a total of 102 sentences that differed from their originals. The average length was 6.46 words, and there were 123 corrupted tokens in all.

The corrupted sentence set was then processed by CLARE with only the spelling correction recovery method in force and with no user intervention. Up to two simple errors were considered per token. No domain-specific or context-dependent knowledge was used.

Of the 123 corrupted tokens, ten were corrupted into other known words, and so no correction was attempted. Parsing failed in nine of these cases; in the tenth, the corrupted word made as much sense as the original out of discourse context. In three further cases, the original token was not among the corrections suggested. The corrections for two other tokens were not used because a corruption into a known word elsewhere in the same sentence caused parsing to fail.

Only one correction (the right one) was suggested for 59 of the remaining 108 tokens. Multiple-token correction, involving the manipulation of space characters, took place in 24 of these cases.

This left 49 tokens for which more than one correction was suggested, requiring syntactic and semantic processing for further disambiguation. The average number of corrections suggested for these 49 was 4.57. However, only an average of 1.69 candidates (including, because of the way the corpus was selected, all the right ones) appeared in QLFs satisfying selectional restrictions; thus over 80% of the wrong candidates were rejected. Treating all candidates as equally likely in the absence of frequency information, syntactic and semantic processing reduced the average entropy from 1.92 to 0.54, removing 72% of the uncertainty. Comparisons of parsing times showed that a lattice could be parsed many times faster than separate alternative strings when the problem token is towards the end of the sentence and/or has several syntactically plausible candidate corrections.

The corpus on which the experiment was carried out consisted only of sentences of which CLARE could parse the uncorrupted versions. However, the figures presented here give grounds to believe that false positives – a wrong "correction" causing a spurious parse of an unparsable original – should be rare. If the replacement of one word by another only rarely maps one sentence inside coverage to another, then a corresponding replacement on a sentence *outside* coverage should yield something within coverage even more rarely.