SGML entities. A number of further reformating procedures were required at this stage. One example was that, with the insertion of an SGML tag to indicate a sentence break, certain parts of the SGML structure of the original document might be invalidated; for instance, a <hi> tag indicating a font change (perhaps for emphasis) which crossed a sentence boundary. In situations like this the post-processing program has to replicate the <hi> tag on either side of the sentence break. Another issue was the use of 'portmanteau tags'. This is a tag indicating that the Claws choice between two tags is particularly uncertain, so that the tag allocated to the word is a combination of the two possibilities. The post-processing program recognizes such situations (of a pair of tags with probability figures from Claws lying within a certain range of each other) and adjusts the tag accordingly (for more details see Leech, Garside and Bryant 1994a).

## 11.6 Conclusion

An error analysis has been performed on the automatic tagging of a portion of the BNC, using the smaller (C5) tagset (Leech and Smith 1995). This shows that there is an error rate of about 1.7%, and about 4.7% of the tags are portmanteau tags. The report gives further details, such as error rates for particular tags and for particular tag pairs.

We are currently engaged in an EPSRC-funded project (GR/K14223) to improve the tagging of the BNC, making use of the manually post-edited 'core' corpus. This is being used as a source of revised Claws resources which will be used to re-tag the BNC; it is also being used as a source of information about error patterns, from which we are developing 'patching' techniques for correcting the main corpus.

# 12 Template analysis: bridging the gap between grammar and the lexicon

STEVE FLIGELSTONE, PAUL RAYSON and NICHOLAS SMITH

## 12.1 Introduction

Since the late 1970s, Geoffrey Leech and the UCREL team at Lancaster have made a notable contribution to the field of Natural Language Processing (NLP) through the development of probabilistic techniques of grammatical analysis. Claws, which has been developed continuously[1] throughout this period, is not only regarded as a pioneering program, but is still one of the most accurate part-of-speech taggers available. Geoffrey Leech has also been involved in, and to a large extent responsible for, various projects leading to significant progress in the more complex problem of automatic syntactic parsing (see Garside, Leech and Sampson 1987; Black, Garside and Leech 1993).

A central feature of all these efforts, which to a certain extent defines the UCREL probabilistic paradigm, has been the use of Hidden Markov Model (HMM)-based statistical algorithms. Though many researchers were slow to embrace them, such empirically oriented statistical techniques have gained currency as a primary means of facing the challenges posed by the complexities of 'real language'. This focus on statistical techniques notwithstanding, since the early days of Claws development, UCREL NLP systems have used additional strategies to achieve accurate output. In this paper we consider what is perhaps the most significant supplementary methodology, which has emerged as an important facet, not only of Claws, but of a number of other software development projects undertaken by UCREL, and which is increasingly showing signs of becoming a practically and theoretically significant approach to NLP in its own right. We refer to this method as 'template analysis', and the best-known example of UCREL's use of the approach in question is the 'idiom-tagging' module of Claws itself.

In the following sections we explain what we mean by 'template analysis', and report on a number of projects in which the method has featured prominently. We consider also some of the reasons why such techniques are useful. We argue that the template analysis paradigm is a useful means of (partially) modelling language, which blends in a pragmatic and conceptually plausible way, aspects of language processing conventionally regarded as belonging to two distinct, albeit interwoven, domains, the lexicon and the grammar.

## 12.2 Template analysis

The basic template analysis paradigm is a simple one, and differs from conventional parsing techniques in that there is no in-built concept of sentence 'structure', but rather an emphasis on the tagging of fragments of text, be these single words, whole phrases, part-phrases or even fragments of text spanning part of one phrase and also part or all of another. The tagger uses a table of template rules to identify significant sequences of text, and apply some specified annotation to that text. A rule consists of a sequence of one or more patterns, each specifying, somewhat in the manner of a standard regular expression, a set of possible input strings which may match that part of the rule in question. A second component of each rule is an action or set of actions to be carried out if the pattern specifications of the rule are met in full. This action is normally specified explicitly within the rule itself, so that each rule may be said to consist of a pattern and an action. Thus a rule applying to a sequence of three input strings, has the canonical form:

*pattern* ACTION, *pattern* ACTION, *pattern* ACTION

Note that for any of the actions in the rule to be carried out, the entire template, i.e. each *pattern* in the rule, must be matched by something in the input. Although in theory all *actions* are optional, at least one action is specified somewhere within the rule (or is implicitly associated with the rule).

In its simplest form, such a rule may consist just of a sequence



of words to be matched, and a sequence of labels, or *tags*, to be applied:

*word* TAG, *word* TAG, *word* TAG

In practice, however, many rules have greater coverage than this. For example, a 'wild card' may be used in the pattern, so that it could match any of several strings. For example, the pattern *depart** would match any of the strings *depart, departs, departed, departing, departure,* etc. Furthermore, in most current applications template rules are used not with raw text as input, but with pre-analyzed text, so that any existing annotations, such as part-of-speech (POS) tags, are also available as target patterns for a rule template.

Thus a rule

*word-tag* NEWTAG, *word-tag* NEWTAG, *word-tag* NEWTAG

or even:

*word* NEWTAG, *tag* NEWTAG, *word* NEWTAG

could be used to modify the tagging of some partially specified pre-analyzed input. This can produce quite broad coverage rules, such that a template *N**, *N**, *V** will match any sequence of words bearing tags indicating (any class of) noun, noun and verb, respectively.[2]

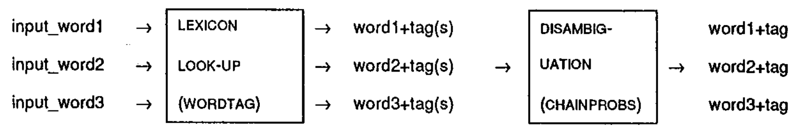
The action component of a rule may also be used to mark other linguistic features in text such as syntactic information or semantic field tags (this is described in section 12.5, below).

## 12.3 Template analysis as a corrective to Claws

We make no claim for the originality of a template approach. Indeed, a system using 'context frame rules' was the tag disambiguation method used in the first major corpus-tagging project, that undertaken with the Brown Corpus (Greene and Rubin 1971, Francis 1980), and as we shall see (section 12.8, below) very fruitful lines of enquiry are currently being pursued by several investigators in the field.

Whilst UCREL set about the task of developing statistical methods for disambiguating tag sequences, it was nonetheless found useful to apply template analysis techniques in order to enhance the performance of the Claws word tagging system. Various template rules, in the guise of an 'idiom-tagging' module, were applied to effect an analysis preferred over that which the Claws HMM-based algorithm would otherwise produce. At various times this part of the tagging system received relatively little attention, whist the 'real work' of developing the statistically driven software progressed. But in recent years the emphasis on idiom-tagging has strengthened once more (see section 12.4, below). The initial development and uses of this part of the Claws system are described in Blackwell (1987), but below we offer a brief summary for the benefit of readers unfamiliar with the basic design of Claws.

If we imagine a Claws system without its idiom-tagging module, and without some of its pre-processing functions, we could represent it schematically thus:




The word-tagging module would use the lexicon and various morphologically oriented rules to produce a list of 'candidate tags' (or, in the case of a grammatically unambiguous word, a single tag) for each word of input. An ambiguously tagged sequence of words would then be input to the probabilistic CHAINPROBS module,[3] which outputs for each word the most likely of the candidate tags, based on a calculation of the probabilities of the various possible tag sequences.

The point of application of the idiom-tagging rules is between the word-tagging module and the probabilistic disambiguation. Any part of WORDTAG's output, that is to say, the words themselves or any of the candidate tags assigned by the program, may be tested against the idiom-tagging rules for a match. It is thus used to influence the actions of the disambiguator by altering the output from WORDTAG which would otherwise be fed to it. In order to achieve this it may execute a number of actions. For a given word:

- it may delete one or more of the candidate tags assigned by WORDTAG
- it may alter the probabilities associated with the tags assigned to it



- it may replace or augment the pre-assigned tags with different tags, not given in the lexicon

Examples of the first case would be the 'strong' rules:

every, little DA1, helps VVZ
shocked JJ, at

Here there is only a single candidate tag for the grammatically ambiguous words *little*, *helps* and *shocked* and the idiom-tagging rule is taking precedence over the other parts of the system, effectively dictating what the output will be.

A common use of the third case is the assignment of 'ditto tags' to sequences regarded for tagging purposes as single entities, such as compound prepositions or certain adverbial expressions (e.g. *at large*, *of course*).

The following rule incorporates all of the aforementioned IDIOMTAG devices, and still leaves work for the probabilistic disambiguator to do:

in II% RR21, vain JJ RR22

Here, the rule eliminates any lexicon-derived candidate tags other than II (preposition) and JJ (adjective) for *in* and *vain*, respectively. Furthermore, it introduces as new candidates the ditto tags RR21 and RR22 (for a compound adverb), and finally, whilst confirming the candidature of the II and JJ tags, it adjust the probability associated with II (the percentage sign indicates a down-grading of the normal probability of a given tag). Since the likelihood of II RR22 or RR21 JJ being selected as sequences is set to zero, the CHAINPROBS program will be left to decide on the basis of the broader context which of the sequences II JJ and RR21 RR22 is the more likely, taking into account the adjustment in probabilities brought about by the rule.

The primary motivation for introducing these rules into the Claws system was to reduce the error-rate associated with the main procedures of lexicon look-up followed by HMM disambiguation. For several years, the number of such rules used at any one time was just a few hundred, but more recently, the number, power and flexibility of the idiom-tagging rules has increased substantially.

What these rules enable Claws to achieve, in a way which eludes the first-order HMM, is to base tagging decisions on information concerning word or word-class sequences with characteristics that cannot be adequately modelled on the basis of adjacency likelihood. This is particularly useful with longer spans of text with some dependency operating between non-adjacent words, and with sequences of words which, due to lexically specific factors are not in fact as ambiguous as they might appear, as it were, to a HMM. In this way, Claws may be said to contain a phraseological component, and thereby a means of dealing with what have been termed 'prefabricated patterns' (Hakuta 1974) or 'speech formulas' (Pawley and Syder 1983)

Because it has been developed to fulfil a specific role in a particular piece of software, the precise workings of the IDIOMTAG module are a little more complex than presented here, and the formalism has developed subtleties which are rather task-specific and not always entirely transparent to the casual observer (the use of implicit actions being a case in point). However, we have presented the essentials in sufficient detail to illustrate the basic mechanism, which is common to all the work described in this paper.

In order to imagine a template analysis module in a more general way, one might envisage the idiom tagger in Claws being brought into play not before the CHAINPROBS disambiguation, but afterwards, as the final processing stage. In this case, the resulting output can be seen as a direct consequence of the actions specified in the template rules. In fact, such a system has been implemented, as a means not of pre-empting, but of correcting errors induced by the probabilistic disambiguation process, and is described in the following section. In other contexts, too, where template analysis has been introduced, development has concentrated on systems where the output is determined directly by actions specified in the template rules.

## 12.4 Recent development of Claws idiom-tagging

The main impetus in expanding the functionality of the Claws Idiomlist in recent years has been the need to tag increasingly large corpora with little or no manual correction. This was particularly the case in the tagging of the 100 million word British National Corpus (BNC).[4] A short passage from the corpus is reproduced below, showing (in bold) a significant proportion of tag assignment decisions that have been made as a result of an idiom rule:



We_PPIS2 passed_VVD the_AT lay-by_NN1 near_II the_AT **Cowal_NP Road_NNL1** junction_NN1 doing_VDG **about_RG** ninety_MC ._. I_PPIS1 watched_VVD as_CSA we_PPIS2 went_VVD by_RP ._. Nothing_PN1 ;_; it_PPH1 was_VBDZ just_RR a_AT1 damp_JJ ,_, deserted_JJ **parking_NN1** place_NN1 with_IW a_AT1 big_JJ new_JJ concrete_JJ litter_NN1 bin_NN1 (_( replaced_VVN with_IW unusual_JJ alacrity_NN1 ,_, in_II **less_RR21 than_RR22** six_MC months_NNT2 )_) ._.
(from BNC, CrowRd:1–7)

### 12.4.1 *Extended rules: modular development*

The template rules which effected the highlighted taggings above were developed in sections arranged on loosely thematic principles, such as geographical and titular nouns, verb strings, foreign expressions and compound nouns. Taking this modular approach analysts can more easily identify gaps and tagging inconsistencies in the coverage of a particular class. While quite a wide catchment is ensured in tagging unseen texts, new patterns or templates identified in sample error analyses can quickly be added to the lists.[5] In this section we explore the wide-ranging nature of the rules in the context of the various thematic modules, while section 12.4.3 deals with the facility of adding or deleting whole lists of entries specific to a particular genre or domain.

#### *The Core list*

A miscellany of items encountered in general English constitute the so-called *Core* Idiomlist. Typical entries disambiguate known problematic words, for example *as*, and idiosyncratic sequences, for example *for certain*, by specifying the appropriate tag(s) in a particular context:

(i) as CSA, follows
(ii) as RG, ([ ])5, as CSA, possible JJ
(iii) for RR21, certain RR22, ![N*]

The first entry is straightforward in that it forces any instance of *as* to be treated as a subordinating conjunction when it precedes the word *follows*. The second rule is more flexible, allowing between up to five unspecified tags (or words) to come in between the *as … as possible* of a comparative construction, assigning the tags RG and CSA to the first and second instances respectively of *as*. The number of unspecified tags can be adjusted for different templates, capturing various long-distance relationships that Markov modelling may treat as ambiguous.

According to the third rule, provided that there is no form of noun tag following it, the sequence *for certain* is taken as an adverbial idiom. Here we employ ditto tagging to capture the integrated quality of the expression with a single duplicated tag. The negation operator '!' is a means of preventing the idiom being applied unrestrictively to cases where ordinary lexical tags (for *IF*, certain *JJ*) would be preferred.

### *Naming expressions*

Our approach to dealing with multiple word sequences representing geographical, personal and titular names — a problematic area in NLP — has been to capture such entities in one of three ways; expressing them either (i) in full, (ii) as a series of alternative words sharing the same tag, or (iii) as word and/or tag templates with options:

(i) Buenos NP1, Aires NP1
Chief NNB, Buthelezi NP1

(ii) New NP1, England/Guinea/Hampshire/Haven/Hebrides NP1

(iii) Fort NNL1, ([WIC])2 NP1, [WIC] NP1
[WIC] NP1, ([WIC])2 NP1, Lake/Lane/Loch/Mill NNL1
Doctor/Dr NNB, ([WIC])2 NP1, [WIC] NP1

The last option is applied in cases where a key word such as *Fort* or *Lane* is regularly followed or preceded by at least one, and up to three, words with initial capital letter (WIC), which can be tagged as Proper noun.[6] Rules such as these have obviated the need to specify individually in the Idiomlists many thousands of multi-word place and titular names, many of which are little heard of (such as *Fort Perch Rock*) or lexically ambiguous (*Dr Bill Smart*).



### *Naturalized foreign expressions*

Ditto tagging has proved especially useful in the area of foreign expressions naturalized into English (cf. Johansson 1986:131):

(i) viva NN121 JJ21 RR21%, voce NN122 JJ22 RR22%
je NN141, ne NN142, sais NN143, quoi NN144

Thus *viva voce*, for example, can function as a nominal, adjectival or adverbial compound. To reflect the comparatively rare usage of this last tag it is given a lower probability (%) marking. We have endeavoured to extend the coverage of this set of expressions as far as possible, to overcome tagger difficulties in guessing an appropriate English POS tag for each of a sequence of generally unfamiliar word forms.

### *Verb strings*

Idiomlist entries have frequently helped to correct the tags assigned by HMM to various phrasal verbs[7] (particularly the particle component):

(i) calm VV0, down RP
switch/switched/switches/switching, off/on/out RP

We have also found templates to distinguish, for example, a past participle (VVN) from a simple past form (VVD), and an infinitive (VVI) from a base form verb (VV0), to be effective:

(ii) [VH*], ([XX/R*])3, [VVD] VVN
(iii) [VM], ([XX/R*])3, [VV0] VVI

Rule (ii) changes VVD to VVN after any form of auxiliary *have*, rule (iii) changes VV0 to VVI after a modal verb. A fair degree of flexibility is allowed in the specification of verb sequences. In these instances up to three negative or adverb tags are permitted in the medial position, while in other templates the permutations of optional tags are more varied. It is, moreover, possible to link rules together, as discussed in the next section.

### 12.4.2 *Pre- and post-HMM idiomlists*

In the past all idioms were processed by Claws in one pass, *before* probabilistic disambiguation. One occasional consequence of this was that some templates designed to improve tagged output actually introduced errors. As an example, a rule from the verb string section:

to TO, ([XX/R*])3, [VV0] VVI

would match on the text sequence *I refer to Report 2a.* — because *Report* is capable of being a verb (as indicated in the lexicon), and it is preceded by *to*. The resultant tagging would be

I_PPIS1 refer_VV0 to_TO Report_VVI 2a_FO ._.

If, on the other hand, one were to place the same rule *after* Markov disambiguation, the action to change a base form verb to infinitive would not occur, since *Report* would already have been disambiguated as a singular noun (NN1). Thus a pass of the idiom-matching routine after HMM does not consider the full set of candidate tags as supplied by the lexicon (or the list of suffixes), but performs an action only on the *preferred* tag as assigned by HMM.

In operational terms Claws merges the thematic lists described above into three sets, two being matched against the text before Markov disambiguation and one after. Some general principles of distribution are: (i) entries which filter the choice of tags, while still leaving it ambiguous, must go in one of the first two sets; (ii) entries which are fairly safe in that they do not require the help of HMM are generally placed before it; (iii) entries which do require HMM should follow it, in the third set.[8]

In this way output of a rule in an early stage of processing may feed into the probability matrix or into a later rule, or both. For example, in the sentence: *You should at least watch!* the first idiom below would mark *at least* as an adverbial, which in turn would ensure that *watch* is disambiguated by HMM as a verb. Application of the second rule below would then ensure that the verb be assigned the infinitive tag:

at RR21, least RR22
[VM], ([XX/R*])3, [VV0] VVI



### 12.4.3 *Genre-specific adaptations*

As an extension of the componential approach to Idiomlist development, we now have the facility to 'slot in' additional or substitutional lists, or to change the way Claws handles the lists, to deal with constructions peculiar to a particular domain or genre.

#### *Scientific/technical*

In tagging two corpora of technical data, the IBM corpus of computer manuals,[9] and the ITU Corpus[10] consisting of telecommunications texts, we added to the Idiomlists entries to improve the accuracy and consistency of tagging of technical terms, such as:

to TO, IPL VVI [11]
the, power NN1, on RP, light NN1
control NN1, channel/device/echo NN1
signalling NN1, [NNT1]

Conversely, certain idioms which may operate in general varieties of English can be omitted, or assigned a lower probability, when processing texts like the ITU. We found, for example, that the informal adverbial sequence

*a RR21, bit RR22* was redundant in the telecommunications corpus, and occasioned errors if retained in the Idiomlists:

In_II this_DD1 mode_NN1 it_PPH1 is_VBZ expected_VVN that_CST behaviour_NN1 of_IO the_AT link_NN1 at_II **a_RR21 bit_RR22** level_JJ is_VBZ not_XX of_IO interest_NN1 ._.

#### *Informal conversational texts*

Approximately five million words of the BNC are of casual, generally unprepared conversation.[12] A set of Idiomlist entries has been compiled to capture colloquial patterns such as slang expressions and non-standard verb forms:

fair JJ, do NN221, 's NN222 (and: fair JJ, does NN2)

cotton VV0, on/onto
did VDD, n't XX, used VMK, to TO

For the processing of spoken texts Claws has been modified to enable idiom pattern-matching in cases of interruption, hesitation and repetition. For example, ditto tags identifying a prepositional unit *according to* no longer need to be contiguous when certain items such as intervening pauses, whether vocalized (e.g. *er* and *erm*) or empty (marked by the SGML <pause> tag), occur between the individual elements:

According_II21 er_FU to_II22 the_AT latest_RRT weather_NN1 report_NN1 ...

Handling of repetition and other features of spoken texts is discussed further in Garside, Chapter 11, this volume.

### 12.4.4 *Future developments*

We are currently increasing the coverage and refinement of the rules outlined above within the existing Claws Idiomlist formalism. One aspect of this work will be to improve, for example, the probability weightings assigned to entries that contain ambiguous taggings.

However, the main area of growth in the future is to be in extending the formalism further, by adopting some of the more powerful template tagging techniques described in subsequent sections of this paper; the principle application of this work will be to produce an enhanced version of the tagged BNC.[13]

## 12.5 Template analysis for Key Syntactic Links (MATRIX)

In a project called 'The Automatic Content Analysis of Spoken Discourse' [14] (see Wilson and Rayson 1993), and continuing in a current project, 'Automatic Content Analysis of Market Research Interview Transcripts'[15] (ACAMRIT, see Chapter 7, this volume), UCREL aims to produce computer-based tools to assist with the linguistic analysis of product surveys of a Market Research company.

We have developed a system which automatically assigns semantic field tags to large bodies of transcribed spoken interviews conducted between the market researchers and members of the public.



Our aim is to allow qualitative analysis to be carried out on a larger scale than is normally possible, along with traditional quantitative evidence only usually found by using fixed multiple choice questionnaires.

The text is POS tagged by Claws, and then semantically tagged. The projects have developed various ways of disambiguating word senses based on rank ordering and domain of discourse. We use a template analysis method to link degree modifiers to adjectives, adjectives to the nouns they modify, and deal with transferred negation. A program called MATRIX (see Rayson 1994) reads user-defined template rules and automatically assigns links between the appropriate words or phrases. The MATRIX formalism is also used for marking auxiliary verbs within the semantic tagger (see below). In the current project we are planning to extend the linguistic analysis to a limited resolution of anaphoric references, and identify key argument-predicate relations using enhanced template analysis techniques.

Previous content analyzers such as General Inquirer have simply provided frequency counts on the content categories for the texts being studied. However, mere frequency counts on individual categories cannot provide information about the referents of adjectives, whether and how adjectives are modified by intensifiers or downtoners, and whether or not assertions are negated. In the past these have had to be annotated by hand.

It should perhaps be said that template analysis here is not intended as a competitor to the method of a full syntactic parse and semantic analysis: rather, it is an attempt to link together the most important descriptive and modifying words in a corpus without the additional computation and manual intervention that the former kind of analysis would require.

Based on the analysis of a preliminary corpus of just over 200,000 words of market research interviews, template rules were developed for the following:

- Linking adjectives to the nouns they modify, e.g.

  The STAFF are HELPFUL

  This is achieved with the template:

  VB*, (XX), (RG*/RR*), J*[<P*/DD1/N*]

where the sequence in square brackets means link the adjective (J*) to the previous pronoun (P*), singular determiner (DD1) or noun (N*).

- Linking degree modifiers to adjectives, e.g.

  The staff are VERY HELPFUL

- Ensuring that negative words are linked to the items which they negate (cf. Wilson 1991), e.g.

  I do NOT LIKE mushrooms

  This also includes transferred negation, where in an indirect statement a negative structurally negates one of a set of introductory verbs (e.g. *think*) but semantically negates the main verb or adjective in the indirect statement, e.g.

  I do NOT think (that) asparagus is TASTY
  I do NOT think (that) Paula LIKES it

  The first example is marked by the template:

  VD*, XX[>J*/V*], (RG*/RR*), think/seem/see, DD*

  where the sequence in square brackets means link the negative (XX) to the following adjective (J*) or verb (V*).

The links for the various kinds of relation form chains connecting the relevant negatives, adjectives, modifiers, and nouns, which are interpreted and applied in producing the final output. These template rules have a success rate of over 90% when run over Claws tagged interview text. With manual postediting of Claws tags, the results are even more satisfactory.

### *12.5.1 The auxiliary/content problem*

A large number of semantic ambiguities in the text are caused by the very common verbs *be, do* and *have*, which are treated as syntactically unambiguous by Claws, but which may have either an auxiliary



function or one or more 'content' (main verb) senses. Rank frequency ordering in these cases is unhelpful as, for example, auxiliary *be* has approximately the same frequency of occurrence as content *be*. However, given that each of the auxiliaries collocate with a particular form of the lexical verb, we have been able to develop template rules based on characteristic sequences of Claws POS tags within which these collocations occur. These rules distinguish the auxiliary uses of these verbs with a high degree of accuracy. For example, the rule:

VH*, (XX), (RR*), V*N

is used to distinguish the auxiliary use *she HAS arrived* from the content sense *she HAS a new car*.

### *12.5.2 Semantic idiom tagging*

The semantic tagger we developed also identifies semantic phrasal idioms such as *all in all* and *up the creek without a paddle* which we mark as one multi-word unit with a particular semantic tag. A human analyst updates a list of templates as new idioms are found (currently we have over 11000 templates). The analyst is able to define which intervening items are allowed between words of the idiom by use of a special template rule, for example:

hand*, {Np/P*/R*}, over RP

which will mark the phrasal verb *handed over* in the sentence *Laura handed the £50 reward money over* because it is interrupted by a noun phrase (Np).[16] The use of templates is an improvement on our previous technique which set a limit on the number of words which could intervene between parts of an idiom but didn't allow us to specify what the words were.

## 12.6 Sharpening Claws (JAWS)

The JAWS[17] program (see Fligelstone, in preparation) applies techniques derived from those in MATRIX (see section 12.5, above) to the task of refining Claws output in such a way as to produce a Claws-like output, but with an enhanced tagset in which there is greater precision in a number of areas. Contextual template rules are used to distinguish not only the main and auxiliary uses of *be, do, have* and *get* (see MATRIX, above), but also to distinguish active past participles from passive participles (identically tagged by Claws), and to distinguish between pronominal and determiner uses of such dual-purpose words as *this, that, more, some, all,* etc. In Claws analysis, no attempt is made to make this distinction, with the tags applied to them being inherently ambiguous. Further rules are used to mark relative pronouns, also indicating case in most instances. In the JAWS variation of the Claws tagset, the pronoun, determiner and verb tags are substantially reworked to take account of the extra information that JAWS is able to include in the output.

The program also carries out lemmatization of the input text[18] so that lexemes, rather than specific word forms, can be used in the formulation of templates, leading to greater economy of rule-writing.

JAWS has also been used experimentally to identify noun phrase heads to a high degree of accuracy, and a further layer of analysis utilizes the linking algorithm of MATRIX to identify a large number of phrasal verbs, linking their roots to their adverbial particles in the form of a dependency link.

## 12.7 Towards a generalized template analysis tool

The above descriptions relate to a number of separate but related language analysis projects in which software development and formalisms have been developed specifically to meet the requirements of each project. However, it is apparent that there is a broad similarity in the approaches, and consequently we feel it is desirable to move towards adopting a standard 'language' for the writing of template rules. The goal is not simply one of notational uniformity, but that of developing more flexible software.

Central to this task is the making explicit of that which is implicit in earlier systems. Whereas it is easy to envisage the use of template rules for the manipulation of various kinds of information, e.g. part of speech, semantic, or dependency, at present each program developed is tailored to work only on levels of annotation relevant to the particular purpose for which it has been written. Insofar as it is possible to define a single set of functions, or actions, which could be used to do all, or at least most, of the template-driven tagging and linking work of the programs discussed in this paper, and a formalism



developed for expressing these functions, then by explicitly stating within a rule the nature of any input or output items specified within it, it should be possible to develop a program which can be used for a variety of analytical purposes, all based on the template analysis approach. For example, a rule cited in section 12.3 could be schematically represented as:

| WRD:every | + | WRD:little | + | WRD:helps |
|---|---|---|---|---|
| | | ↓ | | ↓ |
| | | assign-tag: POS:DA1 | | assign-tag: POS:VVZ |

By allowing the user to inform the program what levels of analysis (i.e. types of tag) will be used for a particular task, a linguistically naïve program can be conceived. The only restrictions on its effectiveness will be the range of tag manipulation functions available, and the appropriateness of the template rules themselves. As well as offering considerable versatility, such a program will provide a useful experimental tool for exploring the scope of template-based methods, as the user is free to invent at will any number of tagsets for simultaneous use, representing an open-ended range of analytical levels.

The task of developing such software is currently being undertaken in the writing of the successor program to JAWS, which is required to carry out a much fuller dependency analysis than merely the identification of phrasal verbs and noun phrase heads. As well as identification of a broad range of dependencies, we are seeking to label heads and adverbials with information concerning their semantic type and semantic role. The 'Template Tagger' is devoid of any 'hard-wired' information concerning tags, their types and their interrelations.

This open-endedness is handled by the use of a user-edited configuration file in which various parameters are set to control the operation of the program:

- a list of the levels of annotation to be matched and/or applied in rules (e.g. POS, lexeme, semantic and head identifying)

- restrictions on the range of valid tokens for any given level (for example, a file may be specified which contains the tagset for use with the level 'POS')

- which rule tables to apply, and in what sequence
- whether (and how many) consecutive passes through a given table should be made
- (for each rule set) whether the program should attempt to apply every rule in the list to a given input, or terminate on the first successful application of a rule.[19]

The tagging operations which may be carried out by a rule's action component include the 'hard' insertion of a tag (overwriting any previous labelling that has been applied), the 'soft' insertion (which succeeds only when the new tag does not conflict with a previously assigned annotation), various operations to apply additional tags, thus increasing ambiguity in the output, and functions which actually modify existing labelling, e.g. in order to subscript major category labels.

At the time of writing a working version of the Template Tagger has been developed[20] incorporating most of the functions referred to above. The Template Tagger at present is essentially a powerful rule-driven 'word-tagging' program, but in due course 'linking' functions, similar to those described in section 12.5 will be added in order to increase its versatility.[21]

## 12.8 Conclusions

The task of text analysis or mark-up is first and foremost a task of disambiguation. A parsing tool which is heavily biased towards grammatical categories of a general kind, and away from individual lexical and collocational knowledge may provide a concise means of defining a set of legal constructs which may be appropriate to the text fragment in question, but it will be prone to generate a large number of legal, but contextually inappropriate, analyses.

But as scholars are increasingly aware, means for rejecting such spurious analyses are available, not only to the human, but also to the machine equipped with some knowledge about what is typical in language use. As currently implemented, probabilistic techniques such as HMMs represent a very useful, but flawed 'catch-all' method of modelling typicality as represented in the likelihood of various sequences, but typically these sequences, too, are defined in terms of



broad grammatical categories and tend to be short-range on grounds of computability.

The template tagging approach allows rules of greater and lesser degrees of coverage to prefer one or a small number of analyses over all others, insofar as they enable typicality, rather than merely 'legality', to be modelled, and also in that such patterns of typicality can be represented efficiently in terms of a combination of lexical, syntactic and semantic features. Whilst it is not argued that a template-based approach described here can ultimately provide a complete method for the automatic analysis of language samples, it does provide a powerful means of ambiguity reduction which can usefully complement other methods of analysis. In the case of Claws we have seen that such methods complement the HMM disambiguation strategy both by altering the input fed to it, and by checking its output for likely defects.

A highly developed and impressive template rule-based methodology is Helsinki's 'Constraint Grammar' approach to syntactic (part-of-speech, dependency and function) analysis (Karlsson *et al* 1995). One point of contrast between our approach and the Constraint Grammar approach concerns the division of labour between the lexicon and the grammar. In the Helsinki approach this distinction is a clear one, with the lexicon being the source of candidate analyses, and the role of the grammar (or rules) being to reduce the ambiguity by eliminating inappropriate analyses. We tend to pay less respect to this distinction. We may use a rule not only to select between conflicting analyses, but to apply an appropriate analysis in the first instance. That is to say, analytical tags may be supplied by the lexicon, or they may be supplied by template rules themselves, and there is no logical necessity in our approach for the lexicon to be the initial source of information. Indeed, it could be argued that there is some merit in allowing more holistic, contextually rich rules to do some processing before the lexicon is consulted, in order to preclude the introduction of unnecessary ambiguity into the analysis. Another way of looking at this is to consider the lexicon as simply a list of very short (normally single element) templates, but not fundamentally different from a more complex template rule, and for the question of the optimal ordering of rules to be based on less abstract considerations that whether something is 'part of the lexicon' or 'part of the grammar'.

Pawley and Syder (1983) suggested that idiomaticity and fluency posed 'two puzzles' for the generative grammarian, and questioned the appropriateness of a strict distinction between grammar and lexicon. In their view, there must be some redundancy in the means to handle a wide variety of language strings, whereby a given string could be produced either by the use of productive grammar rules plus lexicon, or by the invocation of some under-investigated means of producing lexicalized or semi-lexicalized utterances. The template tagging approach turns this around and suggests that exploiting knowledge about 'prefabricated chunks of language' or 'speech formulas' (which might take the form of fixed expressions, but could equally well be stereotypical grammatical constructions) can provide an efficient means of disambiguating text input for the purpose of automatic analysis. In a set of template rules or 'mini-grammars' (Pawley and Syder, op cit., p. 216) which are neither wholly lexical nor wholly grammatical, we may be able to bridge the traditional gap between the grammatical and the lexical components of the NLP system.

Another line of investigation which underlines in a different way the importance of the template approach is that undertaken by Brill (1992 and 1993). Whilst work in Lancaster and Helsinki has depended heavily on the expertise of the linguist to construct meaningful and useful rules, Brill has demonstrated how much can be achieved within a relatively constrained template framework, and, moreover with remarkably little input from the linguist, by developing a tagging system in which an automatically induced set of 'transformation rules' achieve a level of accuracy comparable with HMM-based systems.

At the time of writing, work is under way at Lancaster to apply similar rule-induction techniques (based on a comparison of corrected and uncorrected corpus texts) to create 'patches' (Brill's term) for the task of correcting and enhancing the annotations in the British National Corpus.

We mentioned at the outset that statistical methods had proved strikingly, and perhaps also surprisingly, effective in analysing continuous natural language samples. They achieve this usually in spite of demonstrable weaknesses in their capacity to model human cognitive language production and recognition processes themselves. If the latter term is taken to mean, as is often assumed, the human cognitive 'grammar', then this success may well be connected with the fact that grammar in itself does not provide an adequate explanatory framework for the production and comprehension of human language. Pawley (1994) suggested four other types of knowledge that must be used in the production and comprehension of language by humans: subject matter knowledge, fixed and productive speech formulae and



discourse rules. At least as far as the central pair is concerned (and to some extent also subject matter knowledge), a template approach may provide NLP with a tool that has some features of psychological plausibility. At any rate, it provides a potentially powerful means of modelling 'typicality' features of real language, and thus improving our automatic analyses.

While in the past the primary focus of our NLP research has been on statistical techniques, in recent years template analysis methodologies have gained significantly in prominence. Finding the right blend of rule-based and statistical methods will be one of the interesting challenges for researchers in the coming years.

## Acknowledgements

The authors acknowledge Roger Garside for implementing all programming changes in Claws with regard to idiom processing described here, and for his comments on this paper.

## Notes

1 Although only one version of Claws has been made generally available to the research community, numerous modifications have been made in subsequent versions. The current development version is known as Claws4 (see Leech, Garside and Bryant 1994b), and may be used with a number of different tagsets.

2 Examples concerning part-of-speech tagging are based on the Claws2 tagset, or subsequent variations thereof. In the Claws1 (LOB) tagset, not all verb tags began with 'V'.

3 In Claws1, WORDTAG, IDIOMTAG and CHAINPROBS are actually separate programs. In recent versions, they are distinct modules within a single program.

4 The BNC, compiled between 1990–94, is the result of a collaborative project between two academic institutions (UCREL and Oxford University Computing Services), three publishing companies (Longman Group, Chambers-Harrap and Oxford University Press) and the British Library. The corpus contains 90 million words of written and 10 million words of spoken text, from a wide range of types. Part-of-speech tags (see Appendix to this chapter) were

assigned at Lancaster using Claws.

5 A special-purpose Idiomlist editing tool has been written (by Mark Walsh of the Computing Department, Lancaster University) to enable accurate entering of increasingly complex templates.

6 WICs are usually nouns identified by the lexicon or list of suffixes as being capable of nominal function.

7 But for a more extended catchment of phrasal verbs see section 12.5.2.

8 The actual distribution is not straightforward since rules which apply to the same span of text may need to be separated to avoid problems of overlap (see discussion by Garside in Chapter 11).

9 The IBM Corpus was tagged and parsed by UCREL in collaboration with the IBM T.J. Watson Research Center, New York, USA. See Black *et al* 1993.

10 The proceedings of meetings of the International Telecommunications Union (ITU) were POS-tagged as part of the EU-sponsored ET10-63 and CRATER projects. UCREL collaborated with the University of Essex, IBM Paris, the Paris software house (C2V), and the Universidad Autonoma de Madrid. The projects aimed to provide high quality annotated multi-lingual corpus resources.

11 Where *IPL* means 'Initial Program Load'.

12 These conversations have been sampled demographically from all over the British Isles, and orthographically transcribed with SGML markup. For details see Crowdy 1993.

13 In a EPSRC/DTI-funded project at UCREL called the BNC Tagging Enhancement Project, started in January 1995. (Grant reference GR/K14223).

14 This work was supported by SERC/DTI grant number GR/F36385 to Prof. G. N. Leech, Dr J. A. Thomas and Mr R. G. Garside and ran for 24 months from July 1990.

15 This work is supported by EPSRC/DTI grant number GR/J93733 for 33 months from November 1993.

16 Noun phrases are identified by a separate program called NPEXT, developed by Paul Rayson in the EU-sponsored ET10-63 project at



Lancaster University.

17 JAWS and Template Tagger (see section 12.7) have been developed within the ESRC-funded project 'Lancaster Database of Linguistic Corpora'. The support of the ESRC is gratefully acknowledged.

18 The lemmatization routines were developed by Beale (1989).

19 Other user-configurable features which are being experimented with include a) the ability to make certain input items 'disappear' from the tagger's view in certain circumstances (e.g. quotation marks unless they are explicitly cited in a rule) and b) the assignment for a given rule table of a default tagging function, to allow for greater economy in rule-writing, and to provide the means to compare (by switching the default) different basic tagging strategies.

20 Thanks are due to Mike Pacey of the Computing Department, Lancaster University, for his considerable contribution to the creation of code for the Template Tagger.

21 An early vindication of the Template Tagger approach came with the program's very first run using a full rule set. Independently, Geoff and Fanny Leech had compiled a set of template rules designed to apply partial syntactic bracketing to tagged text as a first stage in a multi-phase approach to automatic parsing. These rules were then translated into the Template Tagger formalism and were applied successfully without any changes being required to the program code. The Template Tagger thus fulfilled the role of obviating the need for special software to be written for each new task, providing the rules can be expressed in the form allowed for by the program.

## Appendix to Chapter 12

Claws7 Tagset

| | |
|---|---|
| ! | punctuation tag — exclamation mark |
| " | punctuation tag — quotes |
| ( | punctuation tag — left bracket |
| ) | punctuation tag — right bracket |
| , | punctuation tag — comma |
| - | punctuation tag — dash |
| ----- | new sentence marker |
| . | punctuation tag — full-stop |
| ... | punctuation tag — ellipsis |