Project ET-10/63: Work Package 3 Report

Roger Garside, Paul Rayson
Department of Computing, University of Lancaster
Tony McEnery
Department of Linguistics, University of Lancaster

1 Introduction

In constructing automatic Natural Language Processing (NLP) systems, we need to supply resources of various kinds to provide information about the linguistic facts of the language or sub-language being processed, generally centering around some form of lexicon. The generation of such a lexicon for a new subject area is a large and expensive task, requiring a linguist to extract information from texts or knowledge experts and encode it in an appropriate format.

This work package seeks to investigate the automation of part of the process of building this lexicon, by extracting information from a corpus of text from a particular subject area. It is possible to envisage a fully automatic approach, whereby a corpus text in a particular domain is analysed to produce the required lexicons and other resources, to be used without further human intervention in an automatic NLP system. Our approach has been more modest; to investigate how information may be extracted from a corpus which has been annotated in a fairly straight-forward way (in the case of this project, with each word of running text assigned a part-of-speech marker), and how this information may be used to supplement and correct a linguist's intuition in the production of resources for such an NLP system. Nowadays, part-of-speech taggers (generally based on HMM techniques) provide a cheap and reasonably accurate way of annotating a corpus with this level of detail, so we believe that it is reasonable to assume such a level of annotation.

The investigation is divided into two sections:

1. argument frame extraction

2. semantic clustering

We expended most of the effort available on implementing an argument frame extraction procedure and in evaluating the results. The work we performed in the semantic clustering was preliminary; though promising, it requires evaluation and further work to yield results useful to semi-automatic lexicon development.

2 Argument Frame Extraction

2.1 Introduction

This part of the work package is concerned with the automatic derivation of subcategorization features for verbs from corpus texts. The description falls naturally into four parts. We begin with a brief discussion of the concept of subcategorization, and review the forms it takes in theoretical syntactic study. Since noun phrases are an important feature of verb subcategorization, we seek to mark them before attempting to observe the syntactic patterns of verb subcategorisation; the next section therefore discusses techniques for doing this. The techniques used in the present study to extract potential verb subcategorisation patterns will then be presented. Finally the results gained from the work will be presented and evaluated, and we briefly discuss the future of the work and suggest improvements to the techniques. A more detailed evaluation appears in the accompanying paper (Maxwell and Peters 1993) [7].

Subcategorization features of verbs may roughly be defined as the grammatical patterns surrounding, and determined by, a given verb. A common feature modelled as part of a verb's subcategorization is transitivity. Traditionally a verb may be transitive, requiring both a subject and object, or intransitive, in which case a direct object does not occur. Needless to say, many verbs, depending on the context in which they occur, are either transitive or intransitive, making this distinction far from simple to observe automatically. We may extend these simple subcategorization classes to include such other classes as bitransitivity, complement consisting of noun phrase and infinitive, clausal complements of various types, etc. Over and above the marking of transitivity, verbs may be pre- and post-modified by optional elements, such as adverbs or adverbial phrases, which further obscure the patterns of subcategorization.

In this study we concentrate on what follows the verb—its pattern of complementation. In saying this, not everything that follows the verb is part of its complementation. It is only those elements following the verb which are obligatory for the use of the verb to be grammatical which form part of the complementization. Furthermore, the complementation patterns will be obscured by such processes as passivization and subject-verb inversion, although the latter is less likely to be a problem in the types of technical text we are analyzing as part of ET-10/63. In essence the question we investigate is this: what grammatical elements may follow a verb, and which of those are optional and which are mandatory?

Even when the features to be marked have been discovered, a further question remains. How is the subcategorization information to be encoded? In theoretical syntactic models there seems to be a broad split on how this should be achieved, as noted by Schieber (1987) [8].

Certain theories, such as HPSG, encode the subcateorizations as lexical features—which words can follow the verb. Somers (1990) [9], on the other hand, notes that grammatical theories in the Chomskyan tradition encode subcategorization features as syntactic patterns around the verb. So we have two broad approaches: grammatical theories which encode subcategorization in terms of lexical entries, and those which encode in terms of grammatical entries. The program under discussion in this section produces output which could be of use to either form of theory. It is possible to generate frequency lists of words surrounding the verb, thus making the encoding of a HPSG style lexicon possible. Yet the program also generates syntactic patterns surrounding the verb, allowing a more syntactically oriented account of subcategorization to be made, which may conform with some of the features of Standard Theory, Extended Standard Theory or GB (Government Binding) Theory.

2.2 Marking Noun Phrases

For the purposes of this project we want to extract subcategorisation frames from part of the ET-10/63 corpus. This corpus is the same as the one used for the Eurotra programme, namely a corpus on Telecommunications consisting of documents from the International Telecommunications Union (ITU), plus a section from the 'CCITT Blue Book' (CEE), as discussed in the ET-10/63 Report on WorkPackage 1 (p4). For this work we used that part of the corpus in English. This part of the corpus is annotated with part-of-speech markings provided by an automatic run of the CLAWS tagging suite, but with no higher level syntactic information about constituent boundaries. As was pointed out above, noun phrase boundaries are important in verb subcategorisation; so we have attempted to insert them using an automatic procedure defined by Church (1988) [3] and claimed by him to identify noun phrases with high accuracy.

In CLAWS a Hidden Markov Model (HMM) is applied to disambiguate potential part-of-speech tags (Garside et al 1987) [4]. This model looks at the context surrounding a word and bases its prediction on a matrix of probabilities derived from previously tagged corpora (such as the Lancaster-Oslo/Bergen (LOB) and Brown corpora). A similar HMM approach is used to identify noun phrases; we extract a matrix of probabilities from a previously annotated corpus, and then use it to assign probabilities to each potential analysis of a sentence (in this case, each potential pattern of noun phrase begin and end markers).

Training the model The trained model consists of a table that shows the frequencies of a noun phrase beginning or ending between each possible pair of CLAWS part-of-speech tags. A small sample of the table is as follows:

```
NN1   VVGK  : NONE   0 : [     0 : ]      1 : ][     0 : TOT    1
NN1   BCS   : NONE   0 : [     0 : ]      1 : ][     0 : TOT    1
IW    MC1   : NONE   0 : [    28 : ]      0 : ][     0 : TOT   28
IW    MC    : NONE   3 : [   560 : ]      0 : ][     0 : TOT  563
IW    RR    : NONE  14 : [    42 : ]      0 : ][     0 : TOT   56
IW    NN2   : NONE   1 : [   300 : ]      0 : ][     0 : TOT  301
IW    AT    : NONE   6 : [  1233 : ]      0 : ][     0 : TOT 1239
IW    DD    : NONE   0 : [    42 : ]      0 : ][     0 : TOT   42
```

The first pair of columns give the two CLAWS part-of-speech tags. Then appear the counts for no intervening noun phrase bracket (labelled `NONE`); for an opening bracket (labelled `[`); for a closing bracket (labelled `]`); for both types of bracket (labelled `][` - there were none in this section of the table); and the total number of occurrences of the part-of-speech pair (labelled `TOT`).

This portion of the table shows, for example, that the probability of no noun phrase brackets between the tags IW and MC (a number of some sort) is quite small (3/563), and that the probability of opening a noun phrase between the tag IW (preposition *with*) and the tag AT (definite article) is very high (1233/1239). The frequencies were derived from an AP (Associated Press) corpus of over one million words of newswire stories, since we judged this to be a corpus of reasonably "general" English. This corpus had already been automatically tagged by the CLAWS system, the tags checked by hand, and then each sentence manually parsed according to a convention of marking only the most significant constituent boundaries, which we called "skeleton" parsing. This level of annotation (Leech and Garside 1991) [6] is a compromise between delicacy of annotation on the one hand and speed (and hence corpus size) and accuracy of the annotation process on the other. An example of an annotated sentence from the AP corpus is:

```
[N The_AT officers_NN2 N][V[V& called_VVD [ for_IF [N the_AT
occupants_NN2 N][ to_TO surrender_VV0 ]]V&] and_CC [V+ were_VBDR
greeted_VVN [P with_IW [N submachine-gun_JB fire_NN1 N]P]V+]V] ._.
```

The simple version of the model which we implemented attempts to insert only one level of noun phrase boundary markers. Consequently we store statistics only for single beginning and ending markers of noun phrases. When the training phase encounters brackets which indicate that multiple noun phrases begin or end at the same place, it conflates them respectively to opening or closing a single noun phrase constituent. No distinction is made for the wider context in which a bracket occurs, whether it marks the open or close of a top level N$''$ or low level N. Brackets indicating other constituents (including unlabelled brackets indicating an unknown constituent) are ignored.

Predicting Noun Phrases Once trained, the model is used in a Viterbi algorithm to insert brackets to indicate the beginning and end of noun phrase constituents. Theoretically, the algorithm calculates all possible patterns of insertion of a single level of noun phrase brackets and choses the "best" one (i.e. the one with the highest probability).

For example, the part-of-speech sequence IW RR (i.e. *with* followed by adverb) has 5 possible bracket patterns (here the square brackets mark the beginning or ending of a noun phrase):

```
... IW RR ...
... IW [ RR ...
... [ ... IW RR ... ] ...
... [ ... IW ] RR ...
... [ ... IW ] [ RR ... ] ...
```

The sequence for example "`... [ ... [ ... IW ] RR ... ] ...`" is ruled out as we do not allow nested noun phrase constituents in this simple version of the technique.

Several modifications had to be made to this simple algorithm. As with the CLAWS tagger, we do not wish to disallow any possible sequence of noun phrase brackets, in principle we want to generate them all and choose the most likely. We therefore generate transition values for all possible part-of-speech sequences, not just the ones seen in the training data, with a bias towards opening and closing no noun phrase if there is no evidence to the contrary.

There were a number of minor problems with the annotated corpus we chose to use, because there was not exact consistency between the tag set used in tagging the AP corpus and that in use the the tagging undertaken as part of the ET-10/63 project. A more serious flaw was that the statistics collected from the skeleton parsing of the AP corpus did not exactly match the probability model of the single level of the type of noun phrase bracketing we were trying to insert. To get round these discrepancies we make some systematic changes to the way we extracted statistics from the AP corpus, before applying it in the Viterbi algorithm. More generally we are in the process of rewriting the algorithm so that it is capable of inserting nested noun phrase brackets, and we believe that this will increase the accuracy of the rests obtained. We are continuing this work, and work on rules to make up for deficiencies in the probabilistic approach, in other research projects.

2.3 Extracting the Patterns

In order to find subcategorisation frames for a verb we have to go through the following steps:

- Find all occurrences of the verb.

- Find the complementation patterns of the verb.
- Determine whether a given subcategorisation frame (SF) is associated with the verb.

Finding a verb is very simple as we have the CLAWS part-of-speech tags to rely on (these are listed in Appendix B). Since un-postedited CLAWS tagging is roughly 96-7% accurate, we can be sure that anything tagged as V* (that is, any tag starting with V) will be a verb with a high degree of accuracy. This gives us a better verb detection rate than Brent 1991 [2], who uses a Case Filter technique to find verbs, before looking for the complementation pattern. Brent says the Case Filter method has roughly 0.5% error rate in comparison to the Penn Treebank tags, but the efficiency is quite low and he loses over 80% of the verbs by choosing only "good" examples. We have a smaller corpus to work with (just over 1 million words as opposed to his 2.6 million from the Wall Street Journal) but our higher detection rate gives us more examples to work from.

Finding the subcategorization is the next task, which is discussed in the next section. Once we have a candidate subcategorisation frame then in principle a statistical test could be applied to determine whether the observations can be assumed to indicate the complementation pattern of the verb. Instead of applying a statistical test, we have used the lists of patterns directly as supplement or correction to a linguist's intuitive understanding of a verb's complementation pattern.

2.4 Finding the Subcategorisation Pattern

We decided to start our investigations on a corpus which was already tagged, manually postedited and skeleton parsed, since this gives an indication of the usefulness of the technique without having to consider the inaccuracies introduced by the automatic tagging and noun phrase marking. The corpus we used was the Associated Press (AP) corpus mentioned above.

Initially we extracted all syntactic patterns from the "focus" verb which we were investigating to the end of the sentence, with the noun phrases discovered in the preceding phase represented simply as "[N]". To reduce the number of alternative patterns we had to consider, we modified the procedure to scan from the "focus" verb up to one of a list of obligatory elements, in the process scanning over a series of possible optional items.

The patterns produced are then sorted into groups based on the obligatory element that has occurred terminating the pattern. The specification of what items are optional, what are obligatory, and how they are to be grouped is not built into the program, but can be easily changed. It can be specified as a part-of-speech tag, a group of part-of-speech tags (by giving a regular expression), or as a part-of-speech tag and word combination. The current grouping of the obligatory elements is:

```
RP
RG*
RL
by_II
II
other_I*
J*
sentence end
to_TO
CS*/RRQ
```

```
RR*
V*
NN*
CC*
Other patterns
```

Thus the first group is those patterns consisting of the verb in question, a sequence of zero or more optional items, and a word tagged RP (i.e. prepositional adverb or particle, such as *about* or *in*). This was included as a separate group because the sequence usually indicates a phrasal verb. The second is terminated by any item whose part-of-speech tag commences "RG" (i.e. some sort of degree adverb). The fourth group is terminated by the word *by* (tagged as a preposition), since such patterns include passive verbs followed by a *by*-phrase, which would not indicate a complementation pattern for the verb in question. The second last group catches all patterns containing a coordinating word; the complexities of coordination structure cannot be handled by these techniques (indeed they are a problem for all parsing techniques), so this group collects patterns of this sort which we cannot handle. The ordering of the groupings is important, as patterns may be candidates for two or more groups; the procedure allocates an instance to the first group it finds in the above list. The final group is for any patterns not caught be a previous pattern; in the development of this list of groups we have investigated what has been deposited in this group, with a view to further refinement of the group list.

The extraction program counts and lists all the different patterns of optional items for each group of obligatory items, giving the proportion this pattern is of the group and of the count for all occurrences of the verb. For each pattern of optional items the program gives a reference for the first occurrence of that pattern, to allow retrieval for further study, or it can print the first example sentence in full. A portion of the pattern list for the verb *indicate* is:

```
Pattern: sentence end: local total = 46
.                          23     50.00%   E0000005 001    1.26%
Sentence: [ The_AT multiplexing_NN1 ] and_CC modulation_NN1 methods_NN2
 associated_VVN with_IW [ them_PPHO2 ] are_VBR also_RR indicated_VVN
 ._.

[N] .                      20     43.48%   E0000003 001    1.10%
Sentence: -_- The_AT intersection_NN1 of_IO [ this_DD1 line_NN1 ] with_IW
 [ the_AT scale_NN1 ] Go_VV0 indicates_VVZ [ the_AT antenna_NN1
 gain_NN1 ] . _.
```

Here the obligatory item is the end of the sentence, and the group contains 46 examples of this from the portion of the corpus investigated. The first pattern is `indicate ._.` (i.e. no optional items between verb and obligatory item) and this occurs 23 times, which is half (50%) of the local group total, and 1.26% of the overall number of patterns for *indicate*.

The pattern groups have been revised a number of times over the lifetime of the project, and the above list indicates the version as in November 1993. A further development would be to vary the groupings depending on the number of examples found, so that for a common verb the program could give a more detailed sub-division of patterns, and for a rarer verb a less detailed list.

2.5 Evaluation and Future Work

A detailed analysis and evaluation of the pattern extraction program PATEXT appears in Maxwell and Peters (1993) [7]. Here we make some general comments on the overall usefulness of the work.

In the output from the pattern matcher, the needs of lexically based theories such as HPSG can be met. The needs of those theories oriented towards syntactic pattern matching are less well met, however. The emphasis here is on making wide ranging generalizations about subcategorization, and utilising those to generate a compact rule set for dealing with the feature; to quote Gazdar et al (1985:33) [5] the aim is to provide "rules which allow the relevant item to be introduced" into the phrase structure of a sentence.

Yet often this involves a combination of syntactic form analysis, syntactic function analysis and semantic analysis. The current program concentrates on syntactic form and to a lesser extent on a lexical approach to semantic analysis. The program assumes that any noun phrases appearing in a syntactic subcategorization pattern will be an object or indirect object. Obviously this need not be the case, with passivization and adjuncts being obvious examples of constructs which would lead to exceptions. So the issue of functional analysis is clearly somewhat obscured in the current program, yet this is an important observation to make, as it clearly has ramifications for the semantic analysis. If we are assuming that the noun phrases we discover are objects (whether direct or indirect) then we are assuming that the head nouns we discover within them are in fact patients as opposed to agents. Again, if the assumption that we are observing objects does not hold, neither does the assumption that we are discovering patients. The same point devolves down to the level of syntactic form analysis. How can we guarantee, baring a full blown parse, that the pattern we detect following the verb is actually part of its sub-categorization, and not just a proximate, but unrelated, feature?

3 Semantic Clustering

3.1 Introduction

There are several approaches to attempting to establish semantic clusters of nouns in a new subject area by investigating an appropriately chosen corpus of text. The approach we have taken is to attempt to build on the argument frame extraction ideas in the other half of the work package. The basic idea is that we can assume that all the nouns which occupy a similar syntactic position with respect to a particular verb are members of the same semantic group. We therefore wished to extract groups of nouns occupying the same syntactic position with respect to a number of verbs, and to see if a similar semantic grouping was induced by different verbs. Another approach is decribed in Maxwell and Peters (1993) [7]; we are investigating a third in a current research project at Lancaster, based on probabilistic semantic disambiguation of text.

In order to carry out this procedure, we need some way of singling out individual nouns which can be clustered together. In order to do this we implemented a procedure which, when given a noun phrase, would indicate which was the head noun of that phrase. We describe this procedure in the next section. In the following section we describe our procedure for clustering these head nouns, and in the final section we indicate the results we have obtained to date and the directions for future work.

3.2 Marking Head Nouns

The noun phrase in English is composed of at least one element. That indispensible element is known as the head of the noun phrase. It is that head which other elements of the noun phrase modify, and determines the agreement features of the phrase. These English phrases are typical endocentric constructions, with a range of premodifiers and postmodifiers available.

Premodification in English can be lexical or phrasal, whereas postmodification tends to be phrasal. The head word identification procedure relies on this observation, by waiting to identify an element which typically signifies either the beginning of a constituent which is typically a postmodifier (e.g. a preposition introducing a prepositional phrase) or the beginning of a constituent which is not actually part of the postmodification (e.g. a verb). By relying on the observation that noun phrases do not postmodify noun phrases, and by looking for key elements that indicate that a phrase other than a noun phrase has begun, the algorithm can generally discover the head noun, as it will be the last element which is an immediate constituent of the noun phrase.

The procedure was described in section 2.2 by which noun phrase boundaries are inserted in a text which has been annotated for part-of-speech information. We now establish the head noun by scanning the noun phrase from left to right, continuing to scan until the end of the phrase is reached, or any word with a part-of-speech tag which could not be the immediate constituent of a noun phrase (such as a preposition). We have taken the simple-minded approach that the head of a noun phrase is a single word, whereas we might, for example, wish to treat the whole sequence of words making up a naming expression such as *Mr John Smith* as the head; this would be a relatively trivial modification to the algorithm, but one we did not make as we were interested in the semantic clustering only of common nouns. In this algorithm the usual premodifiers (determiners, adjectives, etc) are ignored, and the last noun in a sequence of nouns would be marked as the head.

This algorithm works surprisingly well, and in fact often provides the appropriate head word in cases where the marking of the noun phrase boundaries is incorrect.

Where skeleton parsing and corrected tagging have been undertaken on the corpus, the success rate improves appreciably. In experiments undertaken by Tanaka (forthcoming) [11] it has been shown that a success rate as high as 99% can be achieved in head noun identification, where reliable part of speech tagging and skeleton parsing can be used to augment the rule base. Hence the current work is once again only illustrative of the work which may be done in the future with enhanced resources.

3.3 Finding Semantic Clusters

Given that we have, for each verb, a list of possible complementation patterns, and for those complementation patterns which include a noun phrase we have marked the head word of the phrase, our procedure is to

1. extract the list of head words for each syntactic pattern of the verb.
2. investigate the extent to which different verbs induce different syntactic groupings. We envisaged testing for the similarities in two ways, by applying the χ^2 test and by using the idea of "statistical synonyms" (Sutton 1993) [10].

We have modified the PATEXT program which produces the subcategorisation patterns discussed above, so that it collects a list, for each subcategorisation pattern of a verb, of all nouns appearing as heads in noun phrases which act potentially as part of the complementation pattern of the verb. An example for the word *indicate* would be:

```
[N] between_II           8      2.25%   E0000014 001   0.44%
Heads: path(1) relationship(1) link(1) correspondence(2)
interdependence(1) differences(1) boundary(1)
Sentence: [ The_AT solid_JJ lines_NN2_HEAD ] indicate_VV0 [ a_AT1
         possible_JJ physical_JJ path_NN1_HEAD ] between_II [ the_AT
         PLMNs_NN2_HEAD ] through_II [ the_AT PSTN_NNJ_HEAD ] ._.

[N] on_II                8      2.25%   E0000012 001   0.44%
Heads: absence(1) failure(1) call(1) numbers(1) route(2) direction(1)
circuit(1)
Sentence: [ ii_MC ] )_) In_II [ this_DD1 case_NN1 HESd_NP1_HEAD ]
         is_VBZ initially_RR disabled_JJ ,_, and_CC remains_VVZ so_CS
         unless_CS [ a_AT1 sign al_NN1_HEAD ] is_VBZ received_VVN
         from_II Exchange_NN1 E_ZZ1 indicating_VVG [ the_AT
         absence_NN1_HEAD of_IO echo_NN1 suppressor_NN1 ] on_II [
         the_AT outgoing_JJ circuit_NN1_HEAD ] ._.
```

Thus in the small text sample tested here, heads of noun phrases in the pattern "`indicate` noun phrase `between`" include *path*, *relationship*, *link*, etc. OWe then run a program (CLUSTER) which takes as input any number of files of the above form, and extracts and summarises the lists of head nouns. Thus this preliminary work at present does not consider individual syntactic patterns, and looks at the overall patterns of different verbs by performing a pairwise comparison of all the verbs chosen. Each test uses the χ^2 statistic to compare frequency of occurence of a particular head noun with each of the two verbs currently under analysis. The χ^2 test involves calculating the expected frequency of each head and then gives a result which shows how far the values are from the norm. The larger the χ^2 value the more disimilar the two verbs. In this way we can construct a table giving similarity of each verb to another. This is the stage reached at the end of the project, where the CLUSTER program had been written and debugged, but no time remained for running it over a large enough number of verb instances (including examples of verbs which would be expected to show similarity of object noun clustering) to obtain statistically significant results.

It was hoped that it would be possible to form groups of similar verbs (semantic clusters) by linking low χ^2 values together, although we cannot of course assume that because the χ^2 value is low for verb A compared to verb B and low for verb B compared to verb C that it will also be low for verb A compared to verb C. One possible source of such lack of transitivity could of course be the presence of more than one sense of each verb. Further work is also needed on the head noun extraction procedure, in order to conflate nouns with morphological inflection (or more drastically, to conflate all nouns with a cognate stem).

A Program Details

```
Noun phrase extractor: NPEXT
Usage (1) Train stats and predict NPs at the same time:
    npext [options] stats-infile infile outfile
Usage (2) Train stats only.  A 'stats.table' file is produced.
    npext [options] -train stats-infile
```

```
Usage (3) Take a trained model (from 'stats.table') and predict NPs.
    npext [options] -model infile outfile
where:    'stats-infile' contains parsed text for training
          'infile' is parsed text (for testing of prediction)
          'infile' is CLAWS tagged output (with the -claws flag)
          'outfile' shows text with predicted Noun Phrases
options:  -claws : read CLAWS output rather than parsed files
          -po    : produce imitation parsed-output
          -head  : mark heads of noun phrases

Pattern extractor: PATEXT
Usage:   patext [WORD | -i] [-word X | -c Y] parsed_infile outfile
X is the threshold above which word_tag patterns are shown
Y is the amount of following context to show in the patterns
WORDs are: ask, come, give, know, win, originate
           transmit, indicate, send, interwork, orient, multiplex, initiate.
Option -i allows you to type in a list of words to search for.
           Type in each one followed by [RETURN]

Semantic clustering: CLUSTER

Usage: cluster infile.1 ... infile.n
where   'infile.x' is an output file from the PATEXT program.
```

All three programs are written in C on a Sun SLC (OS 4.1)

B The CLAWS tagset

(as at November 1993)

```
!          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
:          punctuation tag - colon
;          punctuation tag - semicolon
?          punctuation tag - question mark
APPGE      possessive pronoun, pre-nominal (e.g. my, your, our)
AT         article (e.g. the, no)
AT1        singular article (e.g. a, an, every)
BCL        before-clause marker (e.g. in order (that),in order (to))
CC         coordinating conjunction (e.g. and, or)
CCB        adversative coordinating conjunction ( but)
CS         subordinating conjunction (e.g. if, because, unless, so, for)
CSA        as (as conjunction)
CSN        than (as conjunction)
CST        that (as conjunction)
CSW        whether (as conjunction)
```

DA	after-determiner or post-determiner capable of pronominal function (e.g. such, former, same)
DA1	singular after-determiner (e.g. little, much)
DA2	plural after-determiner (e.g. few, several, many)
DAR	comparative after-determiner (e.g. more, less, fewer)
DAT	superlative after-determiner (e.g. most, least, fewest)
DB	before determiner or pre-determiner capable of pronominal function (all, half)
DB2	plural before-determiner (both)
DD	determiner (capable of pronominal function) (e.g any, some)
DD1	singular determiner (e.g. this, that, another)
DD2	plural determiner (these,those)
DDQ	wh-determiner (which, what)
DDQGE	wh-determiner, genitive (whose)
DDQV	wh-ever determiner, (whichever, whatever)
EX	existential there
FO	formula
FU	unclassified word
FW	foreign word
GE	germanic genitive marker - (' or's)
IF	for (as preposition)
II	general preposition
IO	of (as preposition)
IW	with, without (as prepositions)
JJ	general adjective
JJR	general comparative adjective (e.g. older, better, stronger)
JJT	general superlative adjective (e.g. oldest, best, strongest)
JK	catenative adjective (able in be able to, willing in be willing to)
MC	cardinal number,neutral for number (two, three..)
MC1	singular cardinal number (one)
MC2	plural cardinal number (e.g. sixes, sevens)
MCGE	genitive cardinal number, neutral for number (two's, 100's)
MCMC	hyphenated number (40-50, 1770-1827)
MD	ordinal number (e.g. first, second, next, last)
MF	fraction,neutral for number (e.g. quarters, two-thirds)
ND1	singular noun of direction (e.g. north, southeast)
NN	common noun,neutral for number (e.g. sheep, cod, headquarters)
NN1	singular common noun (e.g. book, girl)
NN2	plural common noun (e.g. books, girls)
NNA	following noun of title (e.g. M.A.)
NNB	preceding noun of title (e.g. Mr., Prof.)
NNJ	organization noun, neutral for number (e.g. council, department)
NNJ2	organization noun, plural (e.g. governments, committees)
NNL1	singular locative noun (e.g. island, street)
NNL2	plural locative noun (e.g.islands, streets)
NNO	numeral noun, neutral for number (e.g. dozen, hundred)
NNO2	numeral noun, plural (e.g. hundreds, thousands)
NNT1	temporal noun,singular (e.g. day, week, year)
NNT2	temporal noun,plural (e.g. days, weeks, years)
NNU	unit of measurement,neutral for number (e.g. in, cc)
NNU1	singular unit of measurement (e.g. inch, centimetre)
NNU2	plural unit of measurement (e.g. ins., feet)
NP	proper noun, neutral for number (e.g. IBM, Andes)
NP1	singular proper noun (e.g. London, Jane, Frederick)

NP2	plural proper noun (e.g. Browns, Reagans, Koreas)
NPD1	singular weekday noun (e.g. Sunday)
NPD2	plural weekday noun (e.g. Sundays)
NPM1	singular month noun (e.g. October)
NPM2	plural month noun (e.g. Octobers)
NULL	the null tag, for words which receive no tag
PN	indefinite pronoun, neutral for number (none)
PN1	indefinite pronoun, singular (e.g. anyone, everything, nobody, one)
PNQO	objective wh-pronoun (whom)
PNQS	subjective wh-pronoun (who)
PNQV	wh-ever pronoun (whoever)
PNX1	reflexive indefinite pronoun (oneself)
PPGE	nominal possessive personal pronoun (e.g. mine, yours)
PPH1	3rd person sing. neuter personal pronoun (it)
PPHO1	3rd person sing. objective personal pronoun (him, her)
PPHO2	3rd person plural objective personal pronoun (them)
PPHS1	3rd person sing. subjective personal pronoun (he, she)
PPHS2	3rd person plural subjective personal pronoun (they)
PPIO1	1st person sing. objective personal pronoun (me)
PPIO2	1st person plural objective personal pronoun (us)
PPIS1	1st person sing. subjective personal pronoun (I)
PPIS2	1st person plural subjective personal pronoun (we)
PPX1	singular reflexive personal pronoun (e.g. yourself, itself)
PPX2	plural reflexive personal pronoun (e.g. yourselves, themselves)
PPY	2nd person personal pronoun (you)
RA	adverb, after nominal head (e.g. else, galore)
REX	adverb introducing appositional constructions (namely, e.g.)
RG	degree adverb (very, so, too)
RGQ	wh- degree adverb (how)
RGQV	wh-ever degree adverb (however)
RGR	comparative degree adverb (more, less)
RGT	superlative degree adverb (most, least)
RL	locative adverb (e.g. alongside, forward)
RP	prep. adverb, particle (e.g about, in)
RPK	prep. adv., catenative (about in be about to)
RR	general adverb
RRQ	wh- general adverb (where, when, why, how)
RRQV	wh-ever general adverb (wherever, whenever)
RRR	comparative general adverb (e.g. better, longer)
RRT	superlative general adverb (e.g. best, longest)
RT	quasi-nominal adverb of time (e.g. now, tomorrow)
TO	infinitive marker (to)
UH	interjection (e.g. oh, yes, um)
VB0	be base form (finite i.e. imperative,subjunctive)
VBDR	were
VBDZ	was
VBG	being
VBI	be infinitive (To be or not... It will be ..)
VBM	am
VBN	been
VBR	are
VBZ	is
VD0	do base form (finite)
VDD	did

VDG	doing
VDI	do infinitive (I may do... To do...)
VDN	done
VDZ	does
VH0	have base form (finite)
VHD	had (past tense)
VHG	having
VHI	have infinitive
VHN	had (past participle)
VHZ	has
VM	modal auxiliary (can, will, would, etc.)
VMK	modal catenative (ought, used)
VV0	base form of lexical verb (e.g. give, work)
VVD	past tense of lexical verb (e.g. gave, worked)
VVG	-ing participle of lexical verb (e.g. giving, working)
VVGK	-ing participle catenative (going in be going to)
VVI	infinitive (e.g. to give... It will work...)
VVN	past participle of lexical verb (e.g. given, worked)
VVNK	past participle catenative (e.g. bound in be bound to)
VVZ	-s form of lexical verb (e.g. gives, works)
XX	not, n't
ZZ1	singular letter of the alphabet (e.g. A,b)
ZZ2	plural letter of the alphabet (e.g. A's, b's)

References

[1] Black E., R. Garside and G. Leech (Eds.) (1993): *Statistically-driven Computer Grammers of English: The IBM/Lancaster Approach.* Amsterdam: Rodopi.

[2] Brent M. (1991) *Automatic Acquisition of Subcategorisation Frames from Untagged Text.* ACL 29th meeting Berkeley.

[3] Church, K. (1988) A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing.* Austin, Texas.

[4] Garside R., G. Leech and G. Sampson. (1987) *The Computational Analysis of English: A Corpus-Based Approach.* London: Longman.

[5] Gazdar G., E. Klein, G. Pullum and I. Sag. (1985) *Generalized Phrase Structure Grammar.* Blackwell, Oxford.

[6] Leech, G. and R. Garside (1991) Running a Grammar Factory: The Production of Syntactically Analysed Corpora or "Treebanks". In *English Computer Corpora: Selected Papers and Research Guide* edited by S. Johansson and A. Stenstrom. Brelin: Mouton de Gruyter.

[7] Maxwell, K. G. and Peters, W. T. (1993) *Automatic Extraction for Verbs in the ITU Corpus and Semantic Clustering of Their Arguments* Final Report ET-10/63 Work Package 3

[8] Schieber, S. (1987) Seperating Linguistic Analyses from Linguistic Theories. In Whitelock, Wood, Somers, Johnson and Bennett (eds) *Linguistic Theory and Computer Applications* Academic Press, London.

[9] Somers, H. (1990) Subcategorization Frames and Predicate Types. In Schmitz, Schutz and Kunz (eds) *Linguistic Approaches to Artificial Intelligence* Peter Lang, Frankfurt, 1990.

[10] Sutton, S. (1993) *An Investigation into Statistically-based Lexical Ambiguity resolution* Lancaster University PhD Thesis.

[11] Tanaka (1994) *Resolution of Anaphora* MPhil Dissertation, Department of Linguistics, University of Lancaster.