Annotation, retrieval and experimentation

Or: *you only get out what you put in*

Sean Wallis, Survey of English Usage, University College London

1. Introduction

What is the point of annotation?

Let me start with an anecdote. The process of parsing the ICE-GB corpus commenced one year before I joined the Survey in 1995. I observed graduate linguists poring over the results of computation, half-completed tree structures and correcting them. So I asked Professor Sidney Greenbaum what seemed to me to be the obvious question.

> *What are you going to do with this parsed corpus when you're done?*

Parsing a corpus, like any annotation process, may be admirable. It is a challenge of scale, complexity and rigour. But unless we can develop tools and methodologies that can make best use of the annotation, we are left browsing our efforts, skimming the surface of the forest.

2. Three kinds of Corpus Linguistics

There are three different types of answer to my innocent question.

Each type of answer reflects a distinct *methodological stance* towards Corpus Linguistics, determining the status we place on both annotation and corpus data.

Adopting a particular methodological position is independent from the **particular level** of annotation we may be interested in developing. Rather, it shapes how to **elaborate** and **refine** our annotation schemes. In this article we will primarily discuss grammatical annotation. The principles, however, are generic.

2.1 Position 1. Top-down — knowledge is in the scheme

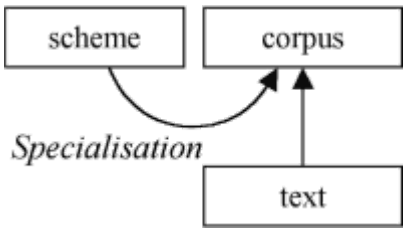

Figure 1. Top-down corpus annotation.

One answer, which could be characterised as the position occupied by much of theoretical linguists, sees the real work in theoretical frameworks and representations. Here, knowledge resides in the scheme. It's *all* in the annotation.

Many linguists in this tradition see corpora as test-beds for their theories at best. They come to corpus linguistics with their schemes fully-formed. This approach has often been described as 'top down', or theory-led, corpus linguistics. Thus 'annotation' is a process which simply applies general theoretical principles in the scheme to specific examples in the text.

From this point of view the scheme is primary, and the corpus secondary.

Natural language processing (NLP) practitioners' principal methodology has been one of computational processing, automatic POS-tagging, parsing and other forms of annotation. Problems that arise in the annotation can be either fixed by altering the algorithm or indeed, excused as the result of 'input noise' or as Chomsky put it, 'performance' (Aarts 2001).

So, from this point of view, there was little point in Greenbaum's team correcting and completing the parsing at all.

2.2 Position 2. Bottom-up — knowledge is in the text

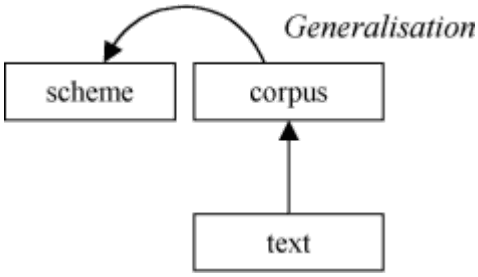

Figure 2. Bottom-up corpus annotation.

The second possible answer to my question is opposite to the first.

Our knowledge of grammar is partial, and quite possibly misleading. There are constant disputes about frameworks (phrase structure versus dependency grammars), information requirements (cf. the Minimalist Program, Chomsky 1995), and classifications within schemes (does an NP include a determiner?). How can one select a single scheme from a universe of possibilities?

Therefore the text is the primary source of knowledge from which all else derives.

This second, 'data driven', school of thought work from their data, upwards. They would argue (correctly) that those who select their facts to fit the theory are ignoring linguistic evidence. A 'true' scientific approach must be able to describe real linguistic utterances and the choices speakers make — not write them off as mere 'performance'. Are linguists arbiters of what is 'grammatical' and what is not?

We have no direct evidence about theoretical competences, so the theory-led are in danger of arid philosophical discussions of imagined abstractions. Interesting phenomena are those we find in real corpus examples, not mere artificial constructs.

From this viewpoint, annotation is definitely secondary, if it has a status. Some might argue that annotation should be avoided entirely, because it could mislead, and we should simply study the sequences of words as uttered. [1]

Certainly, for adherents such as these, there is no point in annotating or correcting the analysis. John Sinclair (1992) famously argued that if parsing is to be carried out, it should not be corrected. This is an eminently sensible decision to take when discussing the parsing of 200 million words of COBUILD. But a necessity is not always a virtue.

The methodologies derived from this perspective base themselves as far as possible on patterns of co-occurring lexical terms, including collocation, concordancing and lexical frames. Note that the role of theory has not disappeared. Rather, responsibility for generalisation is left to the user.

However, the purist position has been under attack from one particular angle: the **success** of wordclass annotation, also known as **part of speech (POS) tagging**. Without manual intervention an accurate classification of over 90% is feasible (the exact figure depends on the scheme).

The results of POS-tagging are extremely useful. If you can distinguish between verb and noun uses of words, users obtain more focused statistical results of far greater value than mere lexical counts. Part of speech analysis permits collocations of a particular verb, colligations and frames, lexicons sorted by wordclass tag, etc.

As a consequence, the text-only position has been watered down. Perhaps the data-driven position is today more accurately characterised as agnostic, a 'minimum necessary' annotation. But then

the question arises, **how much** annotation is necessary or indeed useful? The answer depends on your research goals.

2.3 A pause for reflection

Both perspectives have proved highly productive.

Without practitioners of the first persuasion, parsing and other NLP tools could not have been produced. A focus on the internal logic of grammars provides a theoretical coherence no amount of ad hoc observation can achieve.

The second position has also provided a necessary corrective to the first. Maybe theoretical linguists have focused on rare and abstruse issues. Theoretical linguistics has produced numerous rival traditions, with frameworks endlessly modified. One result has been the growth of a minimalist tendency that parallels that of the data driven perspective.

However, the two traditions have remained apart. Data-driven corpus linguistics has never challenged theoretical linguistics on its own terms.

Let us make a modest proposal. Suppose a parsed corpus could reveal more than a POS-tagged corpus, just as a tagged corpus reveals more than a plain text corpus. In principle, annotation can mislead by conflating occurrences better understood as distinct. There are dangers in this approach.

However some annotation has proved highly useful, particularly in grammar. Later in this paper we discuss some of the ways that a parse analysis can benefit linguistic research. But if you are in the business of proposing an annotation scheme, for example, for pragmatic analysis, how do you know if you're on the right track?

Perhaps the optimum methodology for corpus linguistics is not either/or but **both**. What if the best way for linguistics to proceed is not simply 'top down' or 'bottom up', but a **cycle** in which both generalisation and specialisation processes can occur?

2.4 Position 3. Cyclic — knowledge is in the text and in the scheme

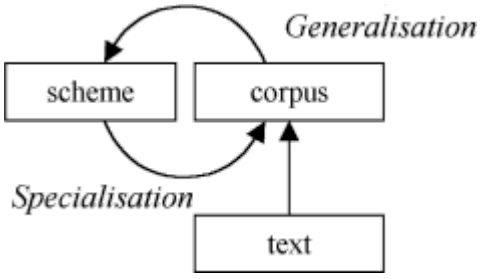

Figure 3. Cyclic corpus annotation.

A cyclic point of view accepts both (a) new observations generalise hypotheses or focus theory and (b) theory is needed to interpret and classify observations.

This loop is not a closed (hermeneutic) circle but an evolutionary cycle. Each passage around the loop enhances our knowledge by refining and testing our theories against real linguistic data. A cycle can involve a single experimenter or a community of linguists debating results and evaluating hypotheses.

Annotation operates **top-down** when a theoretical representation is applied to a text. Consider a simple part-of-speech tagging example, classifying *fishing* as a verb. Suppose that, following automatic tagging, we manually edit the corpus and substitute a compound noun for the analysis of *fishing party*. We have introduced 'situated knowledge' into the corpus at a single point. The tagging tool does not have this compound in its database and no rules exist for how it should be used.

A statistically-based tagging algorithm has a component which 'learns' new definitions and tags from example sentences. These are stored and possibly generalised in the internal 'knowledge base' of the tagger. So alongside a top-down process which classifies lexical items may be a 'learning' process which operates from **bottom-up**.

Given that our initial knowledge is incomplete, correcting automatic annotation by cyclic revision should mean that **both** a more accurate corpus representation is constructed over time **and** a more sophisticated tagger is produced.

While the process may be cyclic, it is not circular. What evolves is a more accurate corpus and (potentially) an improved scheme or algorithm. Knowledge may remain 'situated' in the text, in the form of amended annotations, or be reabsorbed into the tagger or scheme.

Why not simply modify the tagger's knowledge base directly? If you examine a probabilistic tagger you will find a database consisting of a very large number of lexical items and tags, and a series of numbers recording the frequency or probability of their co-occurrence with neighbours. There is value in optimising a tagger (to exploit morphological generalisation, for example) but merely entering examples into a database is suboptimal. It's far easier to edit the corpus and then retrain the algorithm to obtain statistical patterns of inference which can be stored to aid future tagging.

The cyclic perspective lets us take methodologies and tools from both theory-led and data-driven approaches. From this point of view, both automatic and manual refinement of corpus annotation add value. Moreover, as we shall see, improved annotation does not simply lead to a better tagger. Annotation provides a theoretical grip on the lexical corpus. **The corpus becomes a shared topic of linguistic debate.** And on this basis new tools and approaches can be built.

Note that so far the arrow from text to corpus has been assumed to be one-way. However, let us suppose that after annotating a corpus, certain desired phenomena are not captured. In this case, annotation could lead to further text selection.

We now have two cycles, one refining the scheme and knowledge base, and one that identifies the text to annotate. [2]

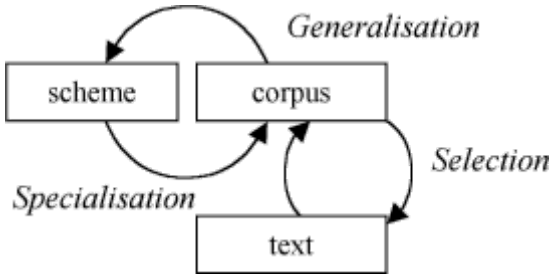

Figure 4. Cyclic corpus annotation showing the text selection cycle.

Text selection is not as trivial as may first appear. How do we achieve a balanced corpus? Which edition of *Twelfth Night* should one parse? What parts of a text or recording should be included or excluded from the annotated sample?

2.5 Summary

In the abstract, any conceivable annotation scheme is compatible with any of the three methodological positions outlined.

1. The top-down position permits a scheme to be elaborated without a requirement for it to describe data. Annotation is an exercise in epistemology. Data only captures performance, not underlying language.
    - **Objection 1**. Internal coherence is insufficient grounds for selecting one representation over another. It is necessary to additionally evaluate a representation by 'usefulness' criteria against data, e.g. **coverage** (how many cases are covered), **precision** (how

accurately are cases classified), and **decidability** (how easy is it to classify a case). Without attempting to apply schemes to corpus data, it is impossible to assess the benefits of one scheme over another.

- **Objection 2**. We do not have direct access to 'Language' through introspective reasoning, computer simulation or neurological methods.

2. The bottom-up position is opposed to prior annotation *per se*. Knowledge resides in the data.
   - **Objection 1**. If generalisation is to be left to the researcher this effectively postpones the decision to annotate. It defers, rather than removes, the problem of annotation.
   - **Objection 2**. Effective induction from text (identifying patterns in the corpus) requires assumptions. These are embodied in an annotation scheme. We should be explicit about assumptions and thereby open them to criticism. An annotated corpus should be the starting point for research.
3. The cyclic position is that corpus annotation is a legitimate field of research inquiry where an annotation scheme can be evaluated against data as part of a **process** of critical engagement. Knowledge is partial and imperfect, and exists in both the corpus text and the annotation.

3. Three stages of Corpus Linguistics

How we want to use an annotated corpus largely determines how our annotation efforts should be focused. The remainder of this paper will sketch out the cyclic approach to corpus linguistics more generally, and some of the implications for how we think about annotation.

We have argued elsewhere (e.g., Wallis & Nelson 2001) that corpus linguistics can be usefully considered according to "The 3A model". This perspective conceives of corpus linguistics at three distinct but interlocking stages, Annotation, Abstraction and Analysis.

- **Annotation** consists of collection, transcription, standardisation, segmentation, processing (POS-tagging, parsing) and often further hand-correction and adding material to text.
- **Abstraction** consists of the process of choosing a research topic, defining parameters and variables and then extracting a sample, or 'abstract model', from the corpus.
- **Analysis** consists of processes which generate and test hypotheses to search for interesting patterns in this model.

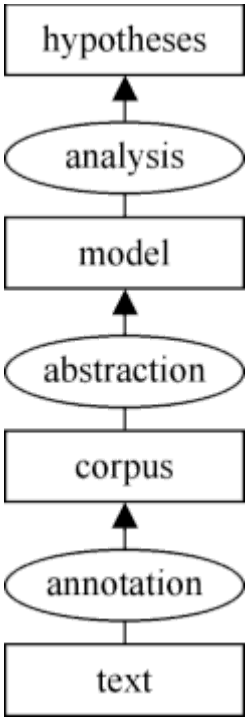

Figure 5. The 3A model of Corpus Linguistics.

As we have noted, each of these processes is **cyclic** and each higher level depends on the lower. Thus abstraction is based upon annotation. Unless a concept is represented formally and consistently in the corpus annotation, it is very difficult to apply a query to reliably retrieve cases of it.

We will return to abstraction and analysis later in this paper. Traditionally, annotation was performed centrally by a corpus builder (or team), whereas later stages of abstraction and analysis were carried out by individual linguists. This is not a necessary distinction, although it is a useful one.

Annotation is, ideally, a **public** endeavour. We annotate to make our data useful to ourselves and to colleagues. Consequently we should not narrowly anticipate the possible uses to which our data may be put, but aim for maximum re-use. Ideally, abstraction and analysis should be shared publicly also.

4. Formal and informal annotation

So far we have taken the term 'annotation' itself somewhat for granted. We must distinguish between formal and informal annotation.

4.1 Formal corpus annotation

Formal corpus annotation can include header information and other sentential markup (sentence subdivision, speaker turns), etc. It can also, as we have seen, consist of the serial identification of linguistic phenomena within a text: nouns and verbs, phrases, discourse moves, stems and suffixes. It might also include alignment markup mapping to translated texts, for example.

The aim of a particular annotation programme is usually to **apply a scheme completely** to the text. A complete analysis clearly increases the potential for re-use over a selective analysis (e.g., identify all nouns or modal verbs but little else, skeleton parsing versus a more detailed analysis, etc.).

A formal scheme can be elaborated and applied for any linguistic level, from prosodic and morphological to speech act and discoursal levels. As well as assigning values to a point or range of terms in the corpus, schemes often collectively form a structural description, such as a grammatical hierarchy.

These formal annotation schemes fulfil a number of robust criteria:

1. **Internal coherence within single terms** — the terms and their relationship to one another can be defined according to a typology. This typology could include
    - Discrete non-overlapping alternatives (a word, *w*, may be a noun or verb but not both) or, alternatively, may permit **gradience** (*w* may be considered to have multiple classifications).
    - Hierarchical relationships (e.g., subtypes of transitivity).
    - A commitment to full enumeration to ensure that all possible terms are listed.
    - For certain types of annotation (e.g., pitch, duration) a numeric typology may be appropriate (this is currently unusual).
2. **Internal structural coherence between terms** — the relationship between different related terms may also be defined formally, e.g. percolation of features, the derivation of transitivity from VP structure, etc.
3. **External consistency** — the *same term(s)* should be employed to capture the *same phenomenon* in the corpus consistently. This is a requirement for retrievability.

Internal rules permit deductive reasoning. The statement "if *x* is a noun, then *x* is not a verb" holds for a part-of-speech (POS) scheme where gradience is not permitted.

Similarly, the statement "if a verb *v* is ditransitive, then *v* is also transitive" holds in a scheme where 'ditransitive' verbs are a subtype of 'transitive' verbs. Full enumeration means that the complete set of possible outcomes are known, so that one can say "if *n* is a noun, and *n* is not singular, then *n* must be plural". Such rules can be exploited in retrieval.

General 'engineering principles' also hold. Any representational system should aim at simplicity as far as possible and complexity as far as necessary.

In one important sense, simplicity is the key to transparency. A scheme that makes fewer distinctions (**N**, **V**, etc.) is simpler than one with a greater number (**N(prop,sing)**, **N(prop, plu)**, etc.). Conversely, some increased complexity may be required to capture certain phenomena, e.g. proper versus common nouns.

Formal constraints (such as the rules above) tend to aid simplicity by specifying how terms interrelate.

As a general meta-principle, simple schemes are easier to apply than complex ones and external consistency tends to be easier to maintain.

These constraints should not outweigh genuine linguistic requirements. **Gradience**, in the form of multiple tags or probabilistic category assignments, may be required in some schemes (see, e.g. Denison, this volume). However, what may seem as a simple amendment to POS-tagging can prove complex once parsing is considered, typically leading to multiple tree analyses.

One particular source of ambiguity is **ellipsis**. There are two common ways of expressing this — to insert a notional element into the lexical stream or to flag the absence of an element at a higher level.

Examples of this phenomenon include verbless clauses, e.g.

> *you all right Gavin?* [ICE-GB S1A-026 #234]

which could be handled by inserting a notional *Are* at the beginning of the sentence.

However, let us consider what this would entail. We can take it as read that complex annotation schemes must cope with necessary ambiguity. Schemes which introduce assumed elements into the lexical stream must clearly distinguish between these elements and the words actually uttered, or we will end up describing (with the top-down linguists), an idealised language. Our query systems must also be able to make this distinction.

4.2 Informal corpus annotation

Necessarily, all formal annotation systems represent a simplification of a complex reality. They sacrifice certain kinds of expressivity that may be desirable. Yet even a background comment may be of value to researchers.

The gap between a rigorous formal system and no annotation at all is filled by what we might term 'informal' corpus annotation. Examples of informal annotation include **explanatory comments** to users, **extra-corpus text**, background on participants that may be difficult to formalise, and **additional layers** (e.g. video, audio).

Informal annotation may be **difficult to search** and **internally inconsistent**. However there are two reasons to include informal annotation in a corpus.

1. It can provide a greater insight into the context of an utterance.

Audio material may not be searchable but hearing the recording of the speakers can bring a transcription 'to life'.

2. It may be used as a reference point to permit the elaboration of more formal annotation in the future.

   In ICE-GB dialogues, **overlapping text** markup was introduced (Nelson, Wallis & Aarts 2002: 13, 96). This annotation indicates where speakers talk independently, signal their desire to speak, interrupt each other, or engage in completing one another's sentences. However this annotation is not currently easily searchable.

For the remainder of this paper we will concentrate on formal annotation, and its primary purpose — to reliably identify linguistic events in a corpus.

5. Retrieval: identifying linguistic events in a corpus

Fundamental to the exploitation of any corpus is the question of the reliable **retrieval** of cases of a particular linguistic phenomenon or 'event'. A linguistic event may be further subclassified, e.g. by grammatical features, lexical realisation, or lexical or grammatical context.

The researcher retrieves cases by defining a **query** on the corpus. The retrieval process *matches* the query to a tree in the corpus, as shown in Figure 6, returning a list of matching cases.

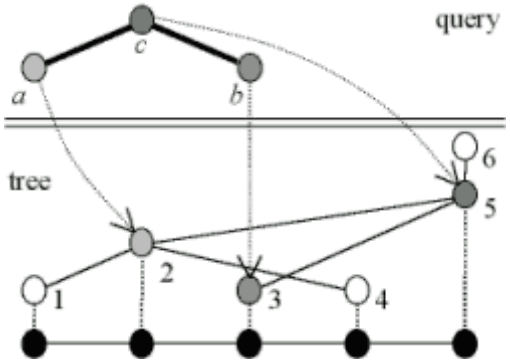

Figure 6. Matching a query to a tree so that <*a*, *b*, *c*> = <2, 3, 5> (from Wallis, forthcoming).

Desiderata:

> *An ideal query system must have sufficient delicacy to capture linguistic events at precisely the necessary level of linguistic generality for research aims.*

One example of delicacy would be the reliable retrieval of a clause and its components (see Figure 10 below). Without the ability to identify each type of clause, one cannot examine all permutations and carry out an experiment. However sophisticated the POS-tagging, reliable clause identification requires a parsing process.

Secondly, whatever the level of detail in our annotation, our research questions will inevitably be more general than the corpus encoding. Thus, for example, Fuzzy Tree Fragments are generic tree structures from which information is deliberately 'thrown away' during abstraction from the corpus.

We refer to this principle as **linguistic adequacy** below. However (as the subtitle of the paper indicates), you can only get out of a corpus what you (or others) put in. In other words, the quality and sophistication of the annotation determines the set of possible queries.

In the next three sections we will examine how this principle might work for a range of grammatical annotation schemes, from plain text to treebanks. Although the discussion centres on grammar, the principles outlined below apply to other levels of annotation, both sequential (cf. POS-tagging) and structured (cf. parsing).

5.1 Retrieval from plain text corpora

In a plain text corpus, queries can consist of individual elements, lexical items (*fishing*), wild cards (*fish**), or even combinations of wildcards (**ing* ∧ ¬*thing*), depending on the query system. Queries can also consist of multiple elements, forming a continuous or discontinuous sequence of terms, (e.g. **ing party*).

Annotation determines retrieval possibilities. To take a trivial example, without word classification, it is impossible to distinguish in a query between *fishing* as a verb or noun.

5.2 Retrieval from POS-tagged corpora

Queries can exploit wordclass annotation by (a) classifying lexical items ('*fishing*+<**V**>' = *fishing* as a verb) and (b) including wordclass tags in strings ('<**PRON**> *was fishing*' = pronoun followed by *was fishing*). It is also possible to simply retrieve all verbs in a given text.

Some query systems can also apply logic and set theory to wordclass tags. Thus (for example) ICECUP 3.1 (Nelson, Wallis & Aarts 2002) lets you specify a tag as '<**PRON ∨ N**>' (pronoun or noun), or '<**V({ed, past})**>' (an *-ed* participle or past verb form). Combining wild cards with lexical items permits queries such as '(**ing* ∧ ¬*thing*)+<**N**>' (a noun ending in **ing*, but excluding *thing*). [3]

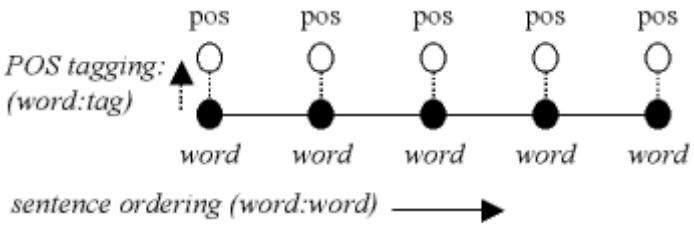

Figure 7. The structure of a POS-tagged sentence (after Wallis, forthcoming).

5.3 Retrieval from parsed corpora

Parsed corpora, or treebanks, contain a structural analysis (in the form of a tree) for each sentence in the corpus. Different parsing schemes contain different constraints, and may be based on diverse theoretical traditions. However there is greater structural agreement between diverse schemes than is often supposed (Wallis, forthcoming). In other words, what constituent terms *mean* may differ markedly, but the rules governing the *structure* are much more consistent.

Without prejudice to grammatical debates regarding meaning, parsing schemes can be summarised as falling under either dependency or phrase structure approaches.

A dependency grammar has one term for each word in the sentence, and these terms, or nodes, are linked together in a branching sequence of dependencies (typically ending at a final head). Dependencies between terms may be labeled with a function, *f*. An example structure is shown below (top).

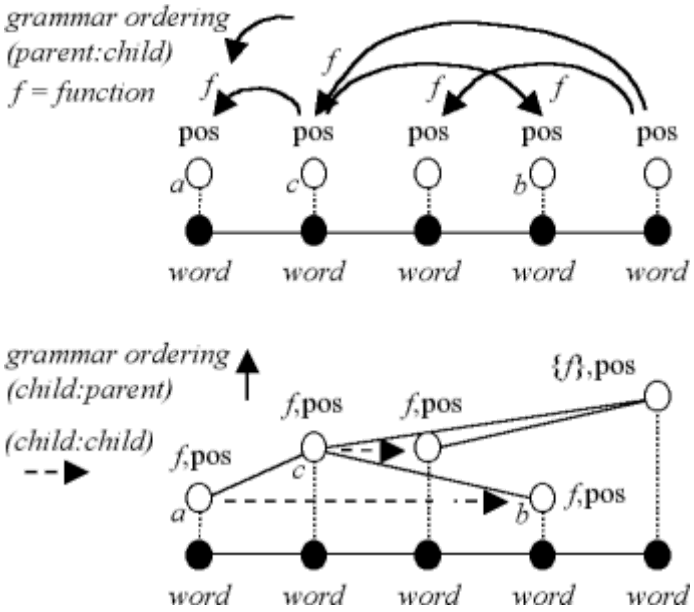

Figure 8. Topology of a dependency or constraint grammar (from Wallis, forthcoming) — alternative visualisations.

The same set of terms can be redrawn (lower) as an explicit tree, and the function may be associated with the node from which the arc originates. [4] Sibling (child:child) order is more explicit in this visualisation. We can see that *a* and *b* are children of *c* and *a* precedes *b*.

Compare this to the equivalent phrase structure illustrated by Figure 9 below. In a phrase structure grammar, each node brackets either a *word* (in which case the term is a part of speech tag) or a set of *nodes* (in which case the term is a phrase or clause). The first obvious property is that the number of nodes has increased. Phrase structure trees have more nodes than dependency grammars (in ICE-GB, nearly 1.8 nodes per word).

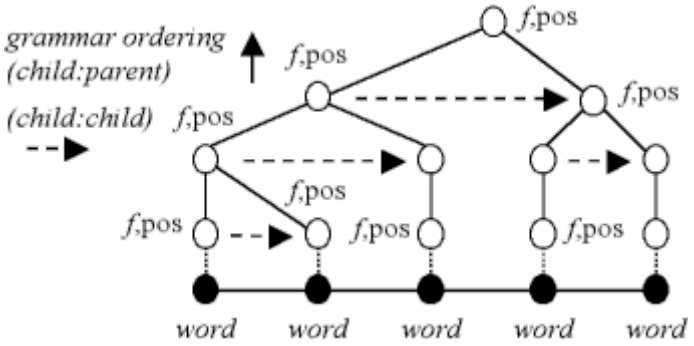

Figure 9. Topology of a phrase structure grammar (from Wallis, forthcoming).

Another difference is that typically, phrase structure grammars are strongly ordered, so that 'crossing links' are not allowed and sentence order determines phrasal ordering.

Despite these structural permutations, however, when it comes to constructing queries on a parsed corpus, similar principles apply regardless of the parsing scheme. The only real difference is the relaxation of certain constraints (e.g. word order implying node order).

6. Grammatical query systems

Query systems in parsed corpora are typically diagrammatic or 'model-based', i.e. they consist of patterns of nodes held to be true together. The alternative logic-based approach has not proved particularly successful (Wallis, forthcoming), not least because users have found real difficulties in expressing queries in this way. Logic is more expressive than a simple model-based approach. However the trade-off is a much less clear query system.

Fuzzy Tree Fragments (FTFs: Wallis & Nelson 2000; Nelson, Wallis & Aarts 2002) are a model-based system used by ICECUP with a phrase structure grammar. The idea is that the query is better understood as a wild card pattern (similar to those we have seen thus far for POS-tagged corpora) than as a formal logical system.

In ICECUP 3.1 (Nelson, Wallis & Aarts 2002), FTFs incorporate logic within nodes by a series of gradual steps. The aim is to avoid making the system unnecessarily complicated but providing the option of including logic where required.

How may we select one query system over another? Wallis & Nelson (2000) proposed the following criteria, in descending order of importance.

- **Linguistic adequacy** — can you express all the queries you need to carry out linguistic research, i.e. is the system sufficient in practice?
- **Transparency** — does a complex query make sense and can it be readily communicated? Logical expressions with two or more tree nodes can be difficult to comprehend. Model based systems such as FTFs are much more intuitive.
- **Expressivity** — can you express all theoretically possible queries? Logical systems are more formally expressive but it is doubtful that this expressivity is useful in practice.
- **Efficiency** — how quickly does the query function in practice?

Model-based systems like FTFs make a correspondence between query and corpus easy to see, as Figure 10 illustrates. ICECUP identifies matching cases in the tree structure as a series of shaded nodes. Since an FTF can match the same tree more than once, this can be very useful to distinguish between cases and to focus attention on each distinct case.

An FTF consists of an incomplete group of tree nodes with a further innovation: links between nodes and edges of the structure can be flexibly constrained. This means, for example, that you can select whether two nodes must be immediately adjacent in the corpus, one eventually follows the other, or that they are unordered. Some links may be removed so that the arrangement is barely constrained at all. Thus the FTF in Figure 10 does not specify the order of words on the right hand side. Links are indicated by the coloured 'cool spots' controlling lines and arrows.

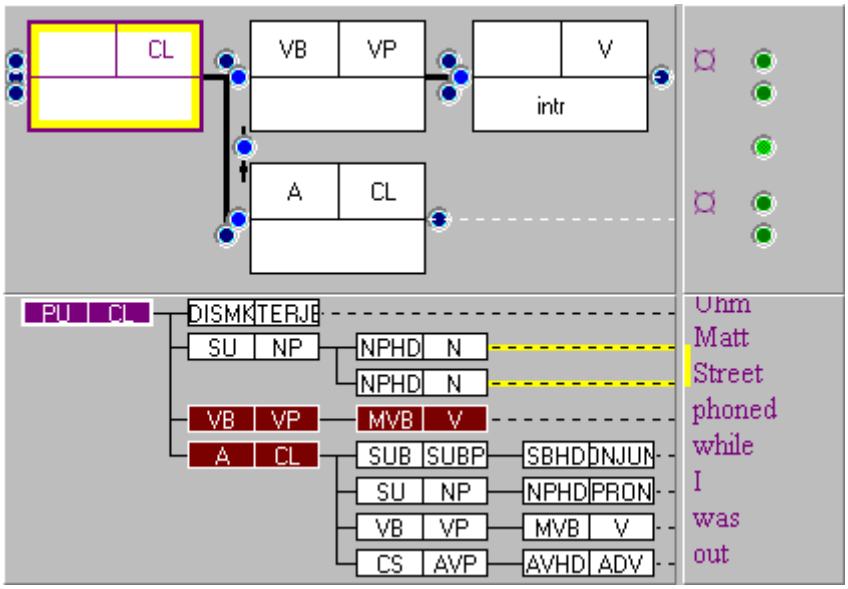

Figure 10. A Fuzzy Tree Fragment and a matching case in the ICE-GB corpus. [5]

These 'looser' queries are valuable for finding a large sample of cases to further refine. The idea is that over a series of queries the researcher can make their query more specific, identifying linguistic events in the corpus with greater accuracy.

Query systems, therefore, are also to be thought of as part of a cyclic process on top of the annotated corpus. This is **the abstraction cycle**.

7. The abstraction cycle

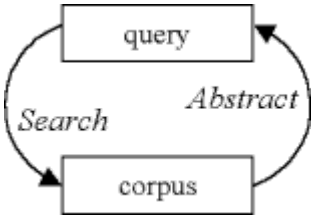

Figure 11. A simple abstraction cycle.

This cycle might look something like this. The query is an abstract pattern which we want to use to identify a particular linguistic event in the corpus. If this event is simply annotated, we should be able to construct a query immediately. Note that we don't just want to locate a single example. We want to find the *definitive set* of all cases.

However as corpus annotation systems become more complex we encounter a Catch-22. To carry out the optimal query we need to know how the grammar is realised in practice across the corpus. But the only way to learn the grammar at this depth is to immerse ourselves in the corpus.

This apparent conundrum is solved once we appreciate that the process can be cyclic. We don't have to get our queries right first time, provided that we may approximate towards an optimal solution.

Some tools, including ICECUP, offer a further trick. A user can select part of a corpus tree and then an algorithm can be invoked to selectively discard information in order to make the tree more abstract and turn it into an FTF. This query will match the source case. The more information that is removed, the more general the query will become and the more analogous cases it is likely to find.

This leaves us with a final problem. What if the grammar is inconsistently applied, or a distinction in the corpus is not one that a researcher agrees with? In other words, *what if the annotation is misleading?*

A single query capturing all cases may not be possible.

Sometimes more than one query is required. ICECUP permits queries to be combined using propositional logic, so you can say 'find me cases like *A* or *B*', where *A* and *B* are separate queries.

Very occasionally, if a phenomenon is not represented in the annotation, the only way to reliably retrieve it may be to **refine** the annotation. In this way the abstraction cycle may prime the annotation cycle by identifying new markup for inclusion or areas where the annotation scheme should be refined.

An abstraction cycle using FTFs can be conceived of as shown below. There is a cyclic relationship between the FTFs and matching cases in the corpus.

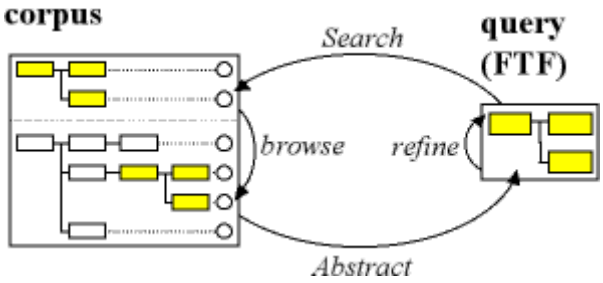

Figure 12. An abstraction cycle with FTFs.

The view of corpus research we have discussed so far has largely been through the selective use of individual queries, driven by research concerns (i.e., top down). However querying a corpus should be just the beginning. Once cases are discovered, what do we do with them?

Note that this is not the only way to proceed. Abstraction need not operate through a single focused query.

Statistical work with a corpus, such as abstracting a lexicon with frequencies, finding the top ten most common words, etc., entails many separate queries (in some cases running into the thousands). To carry this out we require an algorithm that processes a corpus from the bottom, up.

However, firstly, it is difficult for non-computer scientists to construct or modify algorithms. Secondly, it is difficult to integrate relatively low-level statistics (of individual words, etc.) with more general theoretical concerns. (So what, if the most common word in the English language is *the*?)

Some integration of bottom-up approaches with exploratory corpus linguistics is certainly possible and valuable. Tools such as ICECUP 3.1 contain a lexicon wholly derived from the corpus, containing tens of thousands of precalculated terms, each one of which is a query. Others (e.g., Stubbs & Barth 2003) have proposed *n*-grams and similar methods for abstracting a series of patterns (i.e., queries) from a corpus.

The most important point, however, is that researchers have a wide range of different goals. Our corpus methodologies and tools must be sufficiently flexible to support them.

8. Experimental Corpus Linguistics

8.1 Towards an Experimental approach to Corpus Linguistics

Traditional scientific research methods are based around the concept of an **experimental sample** (Wallis 2003). This sample is randomly taken from a wider population of all possible cases. Hypotheses about the world in general may be evaluated against the sample to see if they hold up. Formal statistical methods are used to test hypotheses and select between them. Significance tests allow an experimenter to state with confidence that the patterns they perceive in their sample are no mere accident but are likely to reoccur.

So how do we go about defining a sample from a corpus? The answer is to use queries.

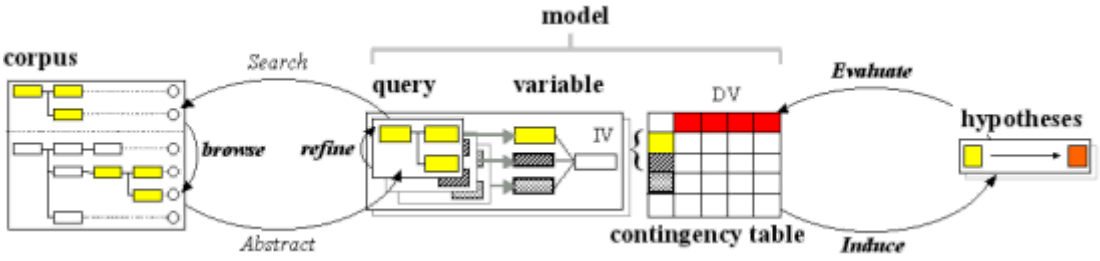

Figure 13. From abstraction to analysis 1. FTF queries define variables to construct a contingency table.

The simplest experimental model consists of two variables, an independent variable (**IV**) and a dependent variable (**DV**). The idea is to try to find out if the value of a dependent variable will change if the independent variable changes (i.e. **IV** → **DV**). The *null hypothesis* is that no significant change occurs.

Let us consider a simple example. Suppose we wish to find out if, in certain clauses, the transitivity of the main verb increases the likelihood that the verb phrase will be followed by an adverbial clause.

Cases in our sample are clauses containing a verb (and therefore a verb phrase). Some of these clauses include an adverbial clause following the verb phrase. We can define FTFs to elaborate

each distinct combination. The process of defining FTFs and obtaining values is described in detail online or in Nelson, Wallis & Aarts (2002, Chapter 9.8.3).

Table 1. Contingency table from ICE-GB (after Nelson, Wallis & Aarts 2002).

CL[VP-V(?trans) ?A,CL]		dependent variable (**adverbial clause**)		
		DV = *T*	**DV** = *F*	**TOTAL**
independent variable (**transitivity of verb**)	intr	2,766	26,848	29,614
	cop	2,159	27,371	29,350
	montr	4,661	54,752	59,413
	cxtr	285	3,628	3,913
	ditr	118	1,574	1,692
	dimontr	19	241	260
	trans	130	2,401	2,531
	0	2	46	48
	TOTAL	10,140	116,681	126,821

In this case the sample is the set of all clauses which conform to the chosen pattern (i.e., clauses containing a verb). We define two discrete variables, the independent variable for transitivity and the dependent variable capturing the presence of the adverbial clause. We then define the values for both variables. Each value of the variable contains a single query that identifies the particular subset of cases.

By applying these queries together one can subdivide the sample along both axes (**DV** and **IV**), generating a table of totals as shown below. A simple chi-square (χ^2) statistic is applied to this data to calculate whether it is likely that the mood feature affects how the direct object is realised.

A statistically significant result means that the observed phenomenon in the sample is likely to reoccur in the population from which the sample is drawn — i.e. in English spoken by comparable individuals similar in time and place to those from which the corpus is taken.

Note that just because a dependency of this type is observed to occur it does not necessarily mean that changes in transitivity 'cause' the adverbial clause to appear or disappear. Both phenomena could derive from a deeper 'root cause'. This is a theoretical question which may not be easily answered by experimentation.

A further problem is that cases may overlap one another. For example, the adverbial clause itself may be a case. The two cases are unlikely to be completely independent, although experimental statistics is based on this assumption. Dealing with this problem, **case interaction**, is beyond the scope of the current paper (this is discussed in depth in Wallis, forthcoming, and elsewhere on the web). Perhaps the most practical observation for us to make here is that experimental results based on a large number of cases spread across a corpus are rather less vulnerable to this kind of criticism than those based on small datasets from a small number of texts.

8.2 Experimenting with multiple explanations

Wallis & Nelson (2001) described a complex experiment using the parsed ICE-GB corpus involving several variables at once. Such an experiment permits us to evaluate numerous competing hypotheses which might explain the same trend or outcome. However, such a procedure is difficult with current tools.

Even the simple act of visualising multi-variable data like this in a contingency table is problematic (each variable is another 'dimension'). Instead, it is often more useful to list the sample as a series of datapoints, case-by-case and line-by-line.

A new project at the Survey of English Usage is concerned with the development of a software platform to enable linguists to carry out research with corpora. The new ICECUP IV software allows users to formally define new variables based on corpus queries and logic, and displays an abstracted sample line-by-line. This **experimental dataset** is shown below.

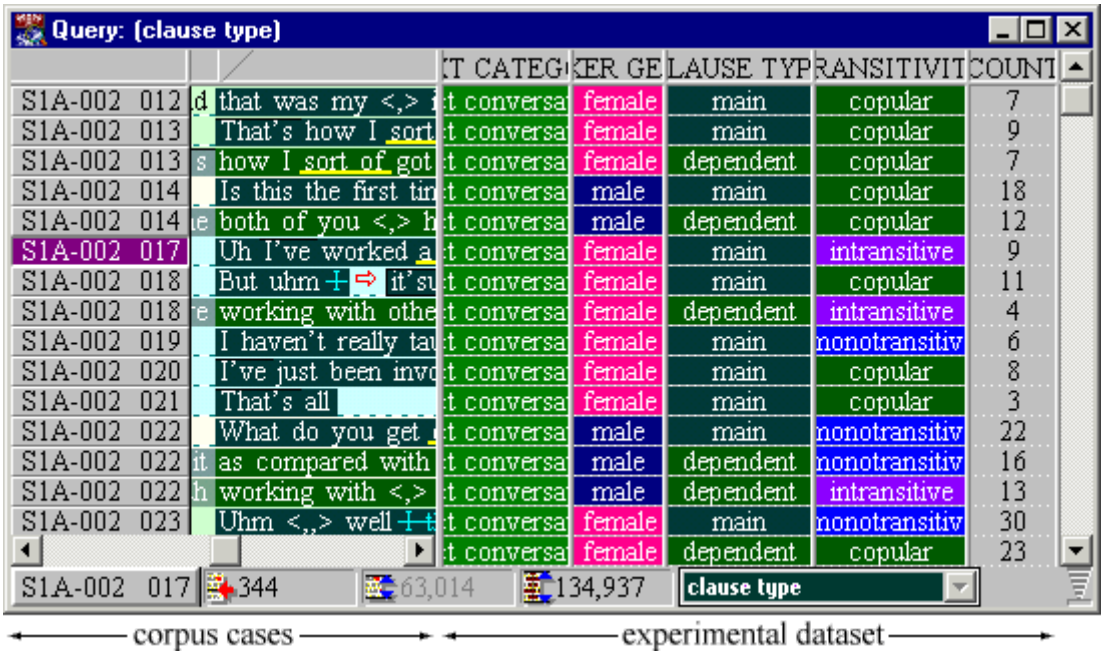

Figure 14. A sample of clauses from ICE-GB classified by **clause type** = {main, dependent}. Cases in the corpus (left) are visible alongside the experimental dataset (right) — a 'spreadsheet' containing datapoints abstracted from the corpus.

The experimental dataset is defined by the same kind of experimental model we saw before. It includes a **case definition** (in this case, clauses) and a series of variable definitions. One of the variables is defined as the dependent variable (**DV**), which we aim to predict from the other (independent) variables (in this case, clause type).

There are three different kinds of variable in this example. These are as follows:

- **sociolinguistic** variables imported from the corpus map (**text category**, **speaker gender**),
- **discrete grammatical** variables (**clause type** = {main, dependent}, **transitivity** = {copular, intransitive, monotransitive...}), and
- a **numerical grammatical** variable (**count** = number of words).

Note that in order to abstract the dataset, grammatical variables are applied separately **to each distinct matching case** in the corpus (i.e. this algebra works case-by-case, not sentence-by-sentence). To see what this means, consider S1A-002 #14 above, which contains two matching clause cases (one main, one dependent). 18 words are found under the main clause (the entire sentence, *Is this the first time both of you...*), while 12 are under the dependent clause *both of you...*. [6]

A trivial hypothesis might be that main clauses are likely to be heavier than (contain more words than) dependent clauses. This hypothesis can be tested with either the Student *t* (Ozón 2004) or Mann-Whitney *U* test. Perhaps a more interesting question might be whether transitivity varies between main and dependent clauses. In this case results could be evaluated with a chi-square (χ^2) test (as in the univariate example above).

Other hypotheses can examine the interaction of variables in predicting an outcome. An algorithm can **search** the space of possible hypotheses, looking for significant correlations and reporting them. The resulting analysis would be a list of independent hypotheses giving possible alternative explanations for a particular linguistic choice. This multivariate approach is summarised in Figure 15.

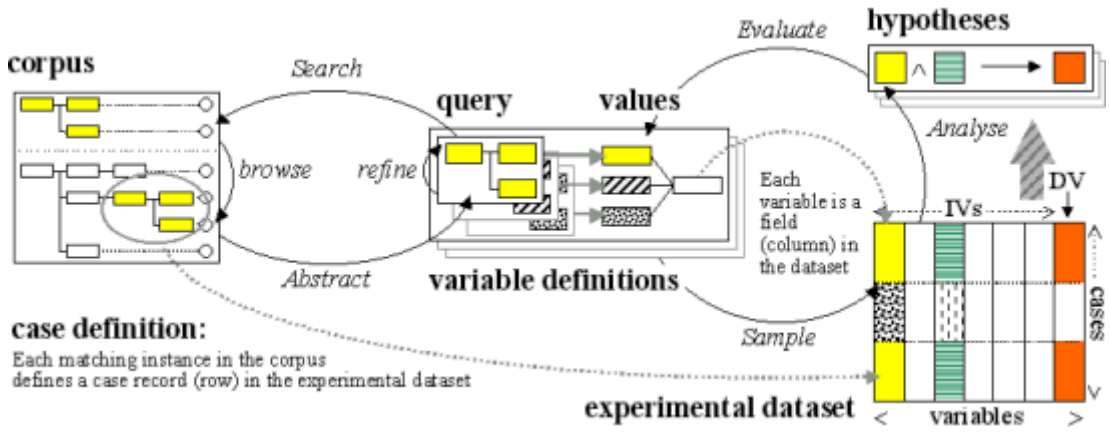

Figure 15. From abstraction to analysis 2. FTF queries define variables to abstract a sample for multivariate analysis.

8.3 Formalising the experimental process

Experimental corpus linguistics consists of applying proven scientific experimental methods to a sample abstracted from the corpus. **Abstraction** and **analysis** can be integrated into a single experimental cycle, as shown below.

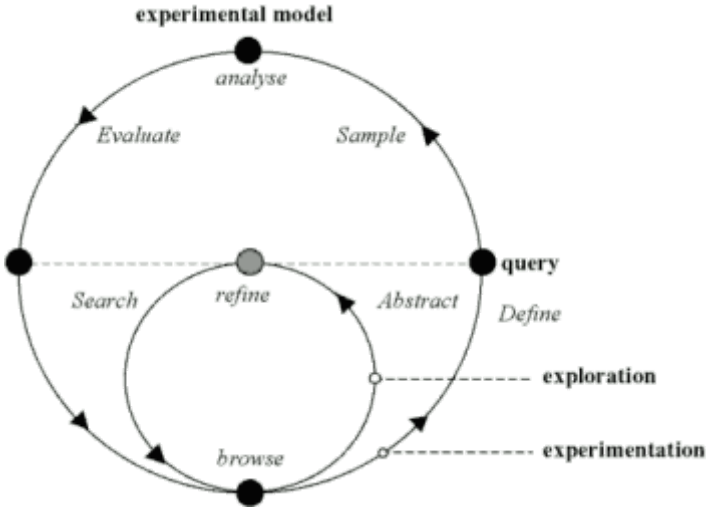

Figure 16. The exploration and experimentation cycles.

This experimental cycle may be usefully considered as four subprocesses: **definition**, **sampling**, **analysis** and **evaluation**. By comparison, our existing tools are really exploratory tools.

1. **Definition** constitutes specifying the research question (e.g. *What determines the choice between V+IO+DO and V+DO+PP in the case of ditransitives?* or *What determines the choice between finite and nonfinite relatives in English?*), as well as specifying variables and queries based on a case definition.
2. **Sampling** consists of enumerating the set of matching cases and classifying them by the variables. The experimental sample can be a random subsample of these matching cases or the entire set of cases if they are infrequent. Where cases are taken from the same text they may be evaluated for case interaction.
3. **Analysis** involves carrying out statistical tests on the sample, obtaining sets of results and refuting or confirming hypotheses. Above we summarised how discrete alternatives can be evaluated with χ^2 tests on totals calculated by ICECUP. Ozón (2004) carried out *t* tests on ordinal, manually-scored, corpus examples.
4. **Evaluation** of experimental results has a dual aspect — verifying results against the corpus and considering results in terms of their theoretical implications.

a. **Intrinsic: Relating results to the corpus**. A researcher needs to know that her observations reflect a real linguistic phenomenon in the data, rather than mistakes in the analysis, an artefact of the grammar, variable definitions or sampling. More positively, examining unexplained cases is often productive, prompting new research questions.

b. **Extrinsic: Evaluating the theoretical implications of the results**, by relating/contributing results to the literature. This includes the following issues and claims:

   i. **Theoretical commensurability** (results relate to reported phenomena).

   ii. **Causality and circularity** (results are neither correlational artefacts nor circular).

   iii. **Inherent simplicity** (hypotheses are expressed in the most simple and fundamental way).

It should be apparent that **definition**, **sampling** and **analysis** may be characterised formally. These are relatively routine procedures and are amenable to automatic processing once an experiment has been defined.

Intrinsic evaluation, process 4a above, is illustrated by the returning arm of the cycle in Figure 16. As we have seen, every case in the experimental sample is identified by a query and, for each case, every variable is instantiated by one or more queries. Every query is, in turn, based on the annotation.

ICECUP IV is an interactive environment where every variable, sample case and hypothesis cross-references (and may be related back to) the corpus. The first type of evaluation — relating results to the corpus — can be achieved by ensuring that a sample is never separated from the corpus from which it is derived (Figure 14).

The new version of ICECUP will allow a linguist to: (1) express research questions and variables, (2) construct and explore an experimental sample dataset, (3) apply and visualise the results of statistical analysis, and (4) trace the source of the results in the corpus.

9. Conclusions

We can only get out of a corpus what we put in.

Annotation decisions determine the types of research questions that we can ask.

Let us consider the experimental model in Figure 14. Each datapoint in the sample derives from a single concordance line. Each concordance line represents a **case** 'of something' in the corpus. Since each case is defined by a query (an FTF), the kinds of questions we can ask ultimately depends on our ability to accurately retrieve cases from the corpus (**retrieval**). Ultimately, what can be retrieved is determined by the annotation.

Applied to a parsed corpus, this methodology permits us to carry out experiments which evaluate naturally occurring speech and writing — to see 'what makes English tick'. More specifically, it permits us to examine the interaction of different grammatical linguistic events within and around a particular linguistic event defining a 'case' in our experimental model.

Figure 14 clearly demonstrates the value of a parse analysis in the process of reliably abstracting an experimental dataset:

- Since the corpus is parsed, we can focus queries, and thus cases, on grammatical constituents (here, clauses). By anchoring all queries **on the same clause**, variables can be instantiated and all possible outcomes elaborated, in a grammatically meaningful way.
- Without parsing, queries would have to rely on collocations and patterns of lexically adjacent items (words or tags). It may be possible to extract and subclassify small phrases by a

tangential process, selectively constructing word+tag pattern queries. However, reliable clause extraction is unfeasible.

- Clause structure implies **embedding** (i.e., clauses within clauses). This raises a further potential problem — how may we deal with the fact that cases in our sample are not strictly independent from one another, but rather, include cases from the same text which partially interact? This **case interaction** issue is discussed in some detail in Wallis (forthcoming). It is also commented on in Nelson, Wallis & Aarts (2002) and on our website.

The experimental vision is a wide and highly extensible one (Wallis, forthcoming). **Increased annotation broadens the range of possible research questions.**

For example:

- Additional layers of annotation would permit researchers to examine the interaction between, e.g., morphology and grammar.

  To see this, consider a parsed corpus where each lexical item was morphologically decomposed in the form of a tree. Such a corpus would make it possible to test morphological agreement rules. If the *International Corpus of English* was so analysed it would be possible to see whether varieties of English worldwide does break morphological agreement, whether 'British English' actually always complies, and if there are other regional rules.

- A parallel treebank would permit research contrasting two parsing schemes.

  Consider a corpus parsed with two or more parsing schemes. Currently there exists only micro corpora like the *Amalgam Multi-Parsed Corpus* (van Zaanen, Roberts & Atwell 2004). However, internationally, the last decade has seen the rise of parsed corpora with a corresponding rise in the diversity of schemes. How is it possible to choose between representations?

  1. We can study whether particular phenomena (linguistic events) are **retrievable** in one scheme but not in another.
  2. We can attempt to discover if there is a **1:1 mapping** between linguistic events, e.g. a clause in a phrase structure grammar and a particular set of named relations in a constraint grammar.

What should we conclude if an annotation scheme is shown to be deficient in some way? The answer is that it may be amended or even rejected. Experiments in grammar are always partially dependent on the grammar. As we have seen, they rely on interpretive steps — annotation and abstraction. Moreover, grounds for preferring one scheme over another may depend on an external criterion — ease of retrieval from a corpus — as well as internal criteria (theoretical consistency, etc.).

The epistemological fact that an experiment depends on a particular set of annotation choices does not constitute grounds for rejecting either annotation or the scientific method. In point of fact, all experimental science rests on an edifice of assumptions. There is no such a thing as an assumption-free science. Our experiments draw out the implications of annotation and abstraction. It is for us to **evaluate** the results. One consequence of such evaluation may be indeed to lead to challenge to basic assumptions and raise criticisms of the annotation scheme.

The main safeguard against theoretical circularity is, therefore, entirely practical: collective criticism and debate.

Experimentation cannot be a wholly automatic process, even if certain subprocesses can be automated. We have to question our own hypotheses and ask —*what do these results really mean?*

Notes

[1] Geoffrey Leech observed (2003) that corpus linguists often make the (possibly unwarranted) theoretical assumption that lexical words are the principal units of language. Those of the data-driven school have little alternative.

[2] This second example demonstrates that describing a process as cyclic does not require that the activity on both arms of the cycle be equal. Rather it requires that the returning arm of the cycle should be *possible*. If primary text (or audio) is lost, further selection may not be feasible.

[3] This example shows that the absence of a particular annotation does not necessarily rule out effective queries on it. In this case, a 'morphological' suffix query (*-ing* nouns) can be constructed using a wild card plus exception(s) and a part of speech constraint. The success of this tangential approach is entirely dependent on the lexical consistency of the retrieved phenomenon in modern English.

[4] Where circularity is permitted (typically in compound tags) elements can be treated as siblings.

[5] This FTF searches for clauses (CL) containing a verb phrase (VB,VP) with an intransitive verb (V(intr)) which is followed by an adverbial clause (A,CL).

In the example, the entire sentence clause is matched. *Uhm* is an interjection discourse marker (DISMK,INTERJEC), *Matt Street*, a compound noun treated as the subject NP (SU,NP). The FTF matches the intransitive verb *phoned* followed by the adverbial clause *while I was out.*

The example also illustrates how the ICE grammar spells out phrases. The adverbial clause consists of

- the conjunction *while* treated as a subordinator (SUB,SUBP-SUBHD,CONJUNC),
- *I* is the personal singular pronoun subject (SU,NP-NPHD,PRON(pers,sing)),
- *was* is the copular past verb (VB,VP-MVB,V(cop,past)), and
- *out* is treated as an adverbial subject complement (CS,AVP-AVHD,ADV).

For more information see The TOSCA/ICE grammar.

[6] This example also illustrates why we need software support to abstract an experimental dataset, beyond that of exploring individual queries. Identifying interactions between discrete grammatical patterns is not trivial with the exploration software, ICECUP 3.1. Extracting values of numeric variables is an even more problematic process. These variables function by counting the number of times an ambiguous FTF matches the corpus.

Sources

Cobuild Concordance and Collocations Sampler, http://www.collins.co.uk/Corpus/CorpusSearch.aspx [archive.org]

DCPSE corpus, http://www.ucl.ac.uk/english-usage/projects/dcpse/*

Fuzzy Tree Fragments (FTFs), http://www.ucl.ac.uk/english-usage/resources/ftfs/
...and experiments in grammar, http://www.ucl.ac.uk/english-usage/resources/ftfs/experiment.htm

- FTF FAQs: Isn't the analysis in the corpus simply one interpretation out of many? http://www.ucl.ac.uk/english-usage/resources/ftfs/faqs.htm#science
- How FTFs match trees, http://www.ucl.ac.uk/english-usage/resources/ftfs/matching.htm
- Links and edges in FTFs, http://www.ucl.ac.uk/english-usage/resources/ftfs/links.htm
- Performing experiments using FTFs: A feature and a constituent, http://www.ucl.ac.uk/english-usage/resources/ftfs/experiment4.htm#example3

- Performing experiments using FTFs: What do we do if cases overlap one another? http://www.ucl.ac.uk/english-usage/resources/ftfs/experiment4.htm#prob_overlap

ICECUP software, http://www.ucl.ac.uk/english-usage/resources/icecup/*

- ICECUP 3.1 Drag & Drop Logic, http://www.ucl.ac.uk/english-usage/resources/icecup/dnd.htm
- ICECUP 3.1 Lexicon & Grammaticon, http://www.ucl.ac.uk/english-usage/resources/icecup/lex.htm
- ICECUP 3.1 Sound Playback, http://www.ucl.ac.uk/english-usage/resources/icecup/audio.htm

ICE-GB corpus, http://www.ucl.ac.uk/english-usage/projects/ice-gb/*

International Corpus of English (ICE), http://www.ucl.ac.uk/english-usage/ice/

Next Generation Tools project, http://www.ucl.ac.uk/english-usage/projects/next-gen/

The TOSCA/ICE grammar, http://www.ucl.ac.uk/english-usage/resources/grammar/index.htm

* ICECUP 3.1 and sample corpora may be downloaded by following these links.

References

Aarts, Bas. 2001. "Corpus linguistics, Chomsky and Fuzzy Tree Fragments". *Corpus Linguistics and Linguistic Theory*, ed. by Christian Mair & Marianne Hundt, 1-13. Amsterdam: Rodopi.

Chomsky, Noam. 1995. *The Minimalist Program*. Massachusetts: MIT Press.

Denison, David. 2007. "Playing tag with category boundaries". *Annotating Variation and Change*, ed. by Anneli Meurman-Solin & Arja Nurmi. (*Studies in Variation Contacts and Change in English* 1). Helsinki: VARIENG. http://www.helsinki.fi/varieng/series/volumes/01/denison/

Leech, Geoffrey. 2003. Contribution to plenary debate at ICAME 22, Guernsey, April 2003.

Nelson, Gerald, Sean Wallis & Bas Aarts. 2002. *Exploring Natural Language: Working with the British Component of the International Corpus of English*. (Varieties of English around the World series). Amsterdam: John Benjamins.

Ozón, Gabriel. 2004. "Ditransitive alternation: A weighty account? A corpus-based study using ICECUP". Presentation at *ICAME 2004*, Verona, 19-23 May 2004.

Sinclair, John. 1992. "The automatic analysis of corpora". *Directions in Corpus Linguistics*, ed. by Jan Svartvik, 379-397. (*Proceedings of Nobel Symposium* 82). Berlin: Mouton de Gruyter.

Stubbs, Michael & Isabel Barth. 2003. "Using recurrent phrases as text-type discriminators: A quantitative method and some findings". *Functions of Language* 10(1): 65-108.

van Zaanen, Menno, Andrew Roberts & Eric Atwell. 2004. "A multilingual parallel parsed corpus as gold standard for grammatical inference evaluation". *Proceedings of the Workshop on the Amazing Utility of Parallel and Comparable Corpora*, LREC 2004, Lisbon, Portugal. http://eprints.whiterose.ac.uk/81661/

Wallis, S. A. 2003. "Scientific experiments in parsed corpora: An overview". *Extending the Scope of Corpus-based Research: New Applications, New Challenges*, ed. by Sylviane Granger & Stephanie Petch-Tyson, 12-23. (*Language and Computers* 48). Amsterdam: Rodopi.

Wallis, S. A. forthcoming. "Searching Treebanks". Chapter 36 in *Corpus Linguistics*, ed. by Anke Lüdeling and Merja Kytö. (*Handbooks of Linguistics and Communication Science, HSK series*).

Berlin: Mouton de Gruyter. http://www.ucl.ac.uk/english-usage/staff/sean/papers/36chap2.pdf

Wallis, S. A. & Gerald Nelson. 2000. "Exploiting fuzzy tree fragments in the investigation of parsed corpora". *Literary and Linguistic Computing* 15(3): 339-361.

Wallis, S. A. & G. Nelson. 2001. "Knowledge discovery in grammatically analysed corpora". *Data Mining and Knowledge Discovery* 5(4): 305-336.

Studies in Variation, Contacts and Change in English 1: Annotating Variation and Change

Last updated 2007-12-19 by Arja Nurmi
Links updated 2016-11-03 by Linda Ravindrarajan