TESL-EJ Teaching English as a Second or Foreign Language

March 2006
Volume 9, Number 4

Contents | TESL-EJ Top

QQ+Concordance: An Analysis Tool For Text Research [1]

Roland Sussex
The University of Queensland
Brisbane, Australia
<sussex@uq.edu.au>

Abstract

This paper presents a software+hardware package for the annotation and analysis of texts, using a combination of common word-processors or text editors and free concordancing software. This package, called "QQ+Concordance," uses in-line tags--tags which are inserted in the linear stream of text--to capture formal, pragmatic, syntactic, semantic, stylistic, contextual and other aspects of texts. The KWIC analyses (key word in context) of QQ- tags support analyses of co-occurrences and patterns of tags with tags, tags with text, and text with text.

QQ+Concordance provides a viable learning framework for students of language. The process of developing, refining and analyzing tags and their patterns, allows students of text analysis to familiarize themselves with the processes of analysis in a nuts-and-bolts way, and to use the output to construct arguments and theories about the content and implications of the components of texts.

1. Introduction

Students and researchers of language need convenient, efficient, transparent and inexpensive means of tagging and analyzing texts so that they can undertake qualitative and quantitative analysis and interpretation. The literature provides numerous approaches to tagging and annotation (see 2.1 below). But there has been less work on the computer-aided extraction of patterns from texts (see 2.2 below; McEnery & Wilson, 1996; Sinclair, 1991). The present proposal presents a

formalism and a combined software solution for machine-aided text analysis. It is based on the observation that concordancing software is able to extract from machine readable text not only words and phrases of raw data, but also tags which have been inserted into the raw text to identify different kinds of linguistic, pragmatic and other evidence.

We propose a simplified markup system and show how to combine it with a text concordancer to provide an uncomplicated, powerful and inexpensive computer-based tool for language researchers and students. The tags in this approach begin with "QQ-," hence the name "QQ notation." This "QQ+Concordance" approach is not as powerful as some proprietary software systems like NUD*IST or ATLAS.TI (Barry, 1998), or the text-based HIAT (Ehlich & Rehbein, 1976), or Shoebox and its successor LinguaLinks (Kretzschmar, 2001). But it is nevertheless capable of supporting serious academic work of substantial dimensions at a modest cost. It also has significant applications to teaching and learning about text analysis, and to language learning and teaching, both in the study of language, and in the study of techniques for exploring language.

2. Annotation, Markup, Tags

2.1 Tagging and annotation

There are numerous systems for text markup, or adding tags to stretches of text of varying lengths for later analysis. Some systems involve tags which are added to a stream of text--the "in-line" tags--for grammatical analysis, like those in grammatical tagging formalisms like CLAWS. Here tags have the form

(1) <w TAG>word

where "w" indicates that the tag deals with a word-class specification. So

(2) <NN1>book

marks book as being a singular common noun. Automatic text analysis programs like CLAWS, which search and tag words and phrase structures for categorical membership like N (noun), NP (noun phrase) and so on, often use similar annotations to the systems for manual annotation.

An extended application of this kind of approach is used for markup in conversation, with two or more participants undertaking conversations which may overlap or interrupt each other. Most of these formalisms, one way or another, go back to the celebrated Sacks, Schegloff and Jefferson (1974) model for conversation markup. Allwright and Bailey (1991, Appendices) list a series of systems designed for the analysis of written and spoken language for linguistic and applied linguistic research. These include an extended list of tags for the analysis of conversation and discourse, and some page-based layout formalisms. For instance:

(3) A: And then Bill, you know, said "No"
B: [Ye:ah] [Why?]

to indicate overlapping talk. In principle, this kind of formalism can be used to label grammatical phenomena like subjects and objects, or heads and modifiers; pragmatic phenomena like statements, greetings, orders and threats, or turn-taking in conversation analysis; and in fact any feature of language forms, functions or content that one wishes to signal as part of the analysis of a text.

In the wider computational literature there are systems like XML and SGML, which use syntax more like that of HTML, the language used to insert tags for Web browsers. There are currently controversies about the usefulness of such formalisms for databases. Although such systems are extensible--hence the "X" in XML for "extensible markup language"--they have at least two principal difficulties for language researchers and students of language analysis:

(a) they are simply not very easy for novices to use. There is a steep learning curve, and there are rules about nesting tags which have to be respected, including the nesting of tags;

(b) their structured nature can be problematic. XML works best with a structured system of tags, where an item has a specific set of known properties. But grounded theory, and bottom-up text exploration, involve the progressive refinement of tags where the structure is being explored as the tags are invented and tested (Glaser, 1995).

There are also two barriers to the implementation of these tagging systems for the purposes addressed in the present paper. First is a problem of orthography. Analysis conventions like those of XML can work well for marking up text for visual / manual inspection and analysis. They can also function effectively for further analysis ***within*** the software packages for which they were designed. But there can be problems if we use the tagged text with other software. Here the tag delimiters which mark the barrier between tags and regular text in these formalisms, like

(4) / . . . /
{ . . . }
< . . .>

are not always recognized as delimiters, and the software may simply ignore the "/" and record the material inside the "/ . . . /" sequence as a piece of regular text (see below), thus obliterating the difference between raw text material and tags. In such a case,

(5a) /question/

would be taken as the word "question" (e.g., "Don't question my judgement") rather

than as a tag, as in

(5b) Don't question my judgement QQQuestion.

And second, there is a Catch-22 from the point of view of the qualitative language researcher. Here the need is to explore the text, to develop ideas about the various components of the text in terms of categories like spelling, morphology, syntax, pragmatics, discourse analysis, conversation analysis, sociolinguistics, cultural linguistics, interpersonal dynamics and other factors. The tagset undergoes a process of constant refining as the analyst works through the text, developing taxonomies and theories, revisiting existing tags and creating working hypotheses. This process of feedback and recursive refinement is fundamental to qualitative analysis, both grounded theories (Glaser, 1995) and more recent approaches to structured qualitative analysis (Miles & Huberman, 1994; Silverman, 1993). This heuristic process needs to be ergonomically and conceptually transparent and simple. And most importantly, it must allow the progressive, bottom-up, inductive development of ideas and theories about the structure and content of texts.

We therefore need to look at alternatives to in-line tagging, especially proprietary software designed for the analysis of qualitative textual data.

2.2 The computer-aided extraction of patterns from texts

There are several proprietary software packages for text and conversation annotation, description and analysis.

HIAT (Ehlich & Rehbein, 1976) is a German software system for discourse analysis (the name stands for *Halbinterpretative Arbeitstranskriptionen*, or "semi-interpretive working transcriptions"). It works under DOS, and provides vertically linked lines for the representation of text and associated commentary, explanations and annotations. Shoebox is a somewhat similar product from the Summer Institute of Linguistics (www.sil.org). Like HIAT, it allows multiple lines of text, linked vertically like a musical score, where each horizontal bar is simultaneous with all those in the same vertical plane. The different lines of text may consist of, for instance, elements like

(6) raw text data
gloss (morpheme-by-morpheme translation)
translation
pragmatic coding (e.g. ORDER, STATEMENT)
stylistic, cultural, communicative, conversational and other vectors

The record can be searched on any line: one can search, say, for examples of ORDER in the pragmatic coding line, and the search will report all occurrences of ORDER, together with all vertically linked lines, so that one can see, for instance, whether ORDER is always realized by verbal expressions like *order, command* and

so on. Shoebox has been widely used, but it is limited in its cross-platform portability (the latest versions run only on Windows), and there is a significant learning curve as one learns to drive and exploit it. Expert users achieve efficient and excellent results. Learners find that the software is not transparent, and until they become competent in its use, the software can get in the way of their thinking about the texts themselves. In other words, what we need is a transparent and very easily accessible and usable interface.

More extensive facilities for qualitative analysis and text exploration and theorizing are provided by NUD*IST and ATLAS.TI, which can also handle multimedia. They both allow the annotation and linking of individual formal items in texts of any size (morphemes, words, phrases, etc.). In addition, they allow users to build networks of associations and trial variables. One can check through a range of colour terms in the poetry of Shelley, say, to see how often they are correlated with names of plants. These annotations are mostly manually inserted (a search-and-replace can semiautomatically insert tags, provided that the anchor keywords are known). The software can then display networks, hierarchical structures like trees, and other configurations of data.

However, NUD*IST and ATLAS.TI are not cheap. Their individual commercial/educational prices are USD 540/USD290, respectively. Companies and institutions able to afford the software have the benefit of full-featured and extensive tools for multimedia exploration and theorizing (see Barry (1998) for a comparison of the two systems). There is also a tradeoff between power and ease of use. While both systems are relatively transparent, given the complexity of the features that they offer, there is a significant startup cost in terms of training and familiarization, until the software starts to feel like a genuinely transparent aid to thinking about the structure and content of the text.

The aims of the QQ notation are more modest, and more geared to use with fewer resources and less time available to learn how to use a large software package. The QQ notation requires two software functions:

- the ability to insert material into a text. This can be handled by standard word-processing systems like Microsoft Word, Open Office, or Mellel; or by text editors like BBEdit;
- the ability to extract items from texts and to list them together for inspection and analysis. This is the domain of concordancing software.

QQ+Concordance, then, aims to access established, familiar computer and texting skills to help researchers and language students think critically and creatively about texts and how they work.

3. Annotation + Concordancing

3.1 An alternative: concordances of tags

Concordancers are software packages that extract occurrences of items in text corpora in three main ways (examples are from the sample text at the end of this paper):

(A) lists of words arranged by frequency, for example:

(7)

the	11
I	7
you	6
s	4
to	4
a	3

Even this small list has some relevant properties of relative order and content: The lack of *of* and *and*, for instance, suggests a lack of modifying phrases and shorter (non-conjoined) constructions. We also find some differences we might wish to explore if we compare this distribution with the top six from the 775,160 word ACE corpus of Australian English:

(8)

the	50,754
of	25,692
and	21,877
to	19,594
a	18,302
in	16,030

(B) lists of words arranged alphabetically, sometimes with frequency counts attached for comparison (using the same example as in (A)):

(9)

a	18,302
and	21,877
in	16,030
of	25,692
the	50,754
to	19,594

(C) KWIC, or "key word in context," where a keyword or headword is presented in the context of its usage. The keyword is typically presented in the middle of the screen. In this output for the keyword "you" the line numbers of the original text are given at the left-hand margin:

(10)

4	: Fred/Hi/Good to hear from	you.	I was beginning to worry that
4	I was beginning to worry that	you	might be sick or something
5	all right. Let me know what	you	think. /But the formulae--
7	. Have you got a copy and can	you	send me one? Don't bother if
7	the article by Short that	you	mentioned last week. Have you
7	you mentioned last week. Have	you	got a copy and can you send

(The text needs to be displayed in a fixed-width font like Courier, to ensure vertical alignment.) This shows *you* in statements and questions; after prepositions (*from*) and in subordinate clauses after *that;* in indirect questions like *what you think;* and so on.

With some concordancers it is possible to specify a number of words to be displayed on either side of the headword, or to allow the software to look for natural breaks like punctuation. Some concordancers allow the user to specify a keyword and another word or phrase used within a given number of words, such as, *love* used within 5 words of *marriage:*

(11)	the institution of	marriage	is accompanied by love
	love leading to	marriage	before the age of 20

Alternatively, one can specify a keyword which is NOT used within x words of another word of phrase.

These combined resources provide fast and reliable means of extracting sequences of language from texts. They have typically been used to investigate large corpora like the language of the Bible, major authors like Shakespeare, or more recently the study, and comparative study, of the language properties of authors or genres. There have also been major uses of concordancers in forensic linguistics and in authorship investigations, as in research on St Paul's gospels, or on the Madison "Federalist" papers in the USA (Collins et al., 2004). There is a vigorous and growing literature on the use of concordancers in computational and general linguistics, especially in journals like *Computers and the Humanities* and *Literary and Linguistic Computing*.

Concordancers are typically used to research natural running text. But they can be used to extract any sequence of characters from a text or corpus, provided that the concordancing software recognizes the characters involved: in working on French corpora, say, the software must distinguish accented letters like *e, é* and *è*, or *c* and ç. Standard concordancers typically work only on linear text: on text in a single stream, line by line. They cannot easily extract information from multiple parallel lines, like a musical score or like the data representations in systems like HIAT and Shoebox (above), and in software packages like NUD*IST and ATLAS.TI, which can

build networked and hierarchical structures between text items (above).

But if concordancers can extract material from running linear text, they can also extract material which has been inserted into the text in the form of comments, notations or tags--labels to identify various items, categories, properties or functions of the text itself.

For instance, say we want to investigate abbreviations. Abbreviations are linguistically important in many kinds of ways: in technical texts and in informal interpersonal communication, to name only two. And they may have multiple different forms like the Australian *arvo, arvie, arv, aftie, arv* and *sarvo* for "afternoon" (Sussex, in preparation). The task is then to extract the abbreviations from the text and to examine their forms, properties, context and usage. One approach is to acquire or assemble a list of abbreviations, and then to use a concordancer, or even a simple search function in a word-processor, to check for the presence of these items in the text. This is laborious and error-prone, since the list of abbreviations may not be complete, and it may miss unfamiliar or even nonce (i.e., occurring once only) forms in the text under investigation.

An alternative is to find a way of manually inspecting and tagging the various abbreviations in a way which will keep the tags distinct from the raw text material, and then to run the concordancing software to extract the tags for abbreviations, together with the raw text examples to which they are attached. This is the key idea behind QQ+Concordance.

3.2. The QQ notation

The QQ notation is a relatively simple answer to the need for a transparent and effective means of tagging phenomena in texts in a way that interfaces directly with standard computer concordancing software. The QQ notation is based on a set of specific criteria:

- unique name (see 3.2.1)
- usable in in-line text (see 3.2.2)
- delimited (see 3.2.3)
- scope (see 3.2.4)
- length, convenience and transparency (see 3.2.5)
- portability (see 3.2.6)

3.2.1 Unique name

The name of the tag must be unique--it must not be the same as the name of another tag--and it must not be confused with regular lexical items in corpora. This the rationale for starting QQ notation tags with the sequence "QQ-," which is not found in any language that I have worked on. In principle "xx-" or "zz-" would have done equally well. "QQ-" is favoured for mnemonic reasons: It implies a question

about usage and interpretation. "QQorder" is then different from order: The former is a tag which refers to a speech act, and the second is the actual word "order." In the following piece of marked-up text the first "order" is a lexical piece of text, and "QQOrder" is a tag:

(12) "Don't do that. I order you not to do that!" QQOrder

The capitalization of ""QQ" and "Order" is discussed below.

3.2.2 Usable in in-line text

The tag must be usable in-line. That is, it must be possible to insert it into a linear line of text, as in example (1). The QQ+Concordance approach is therefore unlike the multi-line HIAT and Shoebox (see 2.2). Nonetheless, one of the aims of the QQ notation is to emulate as much of the functionality of HIAT and Shoebox as possible in a simpler linear software environment. The in-line tags can increase the original text in length by factors which can approach 100%, depending on the density and delicacy of the tags.

3.2.3 Delimiting the tags

Tags need to be graphically distinguished from regular text. If this were not the case, a tag "order" could be confused with a regular lexical item "order." Grammatical taggers commonly use slashes, backslashes, square brackets or braces, usually in pairs around the total tag. HTML uses "<"and ">," with "/" to mark the end of a tag, so that the statement

(13) the *large* antichinus

will switch on italic mode before *large* and turn it off after *large*. The rationale is that the slashes (etc.) are not otherwise used in the regular running text, and so there is no chance of a mis-analysis based on software mistaking a tag for a genuine word.

The QQ notation makes a simplifying assumption which derives from the fact that some concordancing software is not able to count slashes, braces or brackets as parts of words. This means that

(14) /noun/

may be interpreted as a regular lexical item "noun" between punctuation, and not as a tag denoting a grammatical category. On the other hand, no word in English (or in any other language that I am familiar with) begins with the sequence "QQ-." So

(15) QQNoun

will be unambiguously interpreted as a single tag for the noun category, and will be

extracted by a concordancer as "qqnoun".[2]

A tag in QQ-notation is a single word with no spaces or punctuation, and consists only of alphanumeric characters. This follows the usage of the Simple Concordance Program; other concordancing software may allow different characters. Permissible examples in the QQ notation therefore include:

(16) qqExclam (= exclamation)
qqNameInvite (= an invitation to someone to call you by a particular name)
qqPragOrder (= pragmatics: an order or command)

By convention, new words or morphemes inside a tag start with a capital letter, but that convention can be freely changed by users. Concordancers may ignore case differences, or it may be possible to set them to recognize case.

Tags are separated from preceding text by a space. They may be followed by space (if in the middle of a sentence) or by punctuation which belongs to the original text:

(17) "Ow! qqExclam she said.
You can call me "Fred" qqNameInvite.

3.2.4 The scope of tags

Tags, by definition in the QQ formalism, refer to the word or words to their left:

(18) Hi! qqGreeting

One issue not yet resolved in the QQ notation is the question of scope: how much of the preceding text is covered by the tag? This question is still being investigated. In principle it could be solved by

(19) qqStartChallenge . . . qqEndChallenge

which recalls the HTML

(20) . . .

3.2.5 Length, convenience and transparency

Tag names can be of any length. In addition to the "qq-" prefix they can contain in principle any number of meaningful words or parts of words, often abbreviated for convenience:

(21) QQErrorSpelling spelling error
QQErrSpell spelling error

Maximum transparency is achieved with no abbreviating of the names, and maximum clarity of grammatical/lexical relations between the items.

3.2.6 Portability of tags and tagged text

An important feature of the QQ notation's design is that it is possible to edit QQ tags into the working copy of the original text with simple text editors or word-processors. Files are saved as text-only, since some concordancing packages cannot handle the kinds of formatting codes that are standard in word-processing packages like Microsoft Word. Such files are portable across platforms, specifically between UNIX, Apple's operating systems and Windows.

In addition, and more significantly, QQ tags are easily read and handled by a wide range of concordancing packages, many of them available via the Web free of charge. This is not the case with Shoebox--though Shoebox does contain a (limited) concordancer of its own. Many concordancers are now able to handle non-standard character sets and accented letters, though they may still have problems with digraphs like "ll" or "ch," which in some languages are treated as units and not as mere sequences of letters. These units may have a place in the alphabetical sequence which is not the same as that which would hold if they were treated as sequences of characters: in English LL is L+L, whereas in Spanish LL is a letter which is ordered after all the L entries; similar arguments apply to CH in languages like Slovak, where it is listed after H. Digraphs also skew letter-counts if one is counting the occurrence of different letters in a text. These issues can be handled by special counting of letters and sequences.

3.3 QQ+Concordance

Once we have a tagged text we need to extract the tags into a concordance (frequency, ABC, or KWIC listing) for investigation. There are many concordancing packages available for different computing platforms and operating systems. They include freeware, shareware and commercial products. The software chosen for the purposes of this paper is Simple Concordance Program, a piece of free software written by Alan Reed. It runs on both Windows and Macintosh platforms, and is available from several web sites (references at the end of this paper). Simple Concordance Program runs under Windows, and on Macintosh computers under OS X. Other concordancers would suit as well, though many run only on one platform, and some are relatively expensive (see "WWW Sources" below for current information).

3.3 Using the QQ notation with the Simple Concordance Program

At the end of this paper we append a sample raw text file and the same file marked up with QQ tags. These can be used in their current form for testing the Simple Concordance Program. For this demonstration, cut the Sample Text from the end of

this paper, omitting "vvv" and "^^^." Paste it into a new document. If you are using a text editor, just save it. If you are using a word processor like Microsoft Word, select "Save As ..." and select "Text Only" from the Format options. Give the file the name "tryscp.txt" and place it in the same directory as the software which drives the Simple Concordance Program.

Now proceed as follows:

1. Create two backup copies of the raw data. Store the original in a secure place separate from the original raw data file, make it "Read Only" or lock it with software to prevent inadvertent modification.
2. Use version #2 for concordance searches of the raw untagged data. It is wise to restrict access to it by making it "Read Only" to prevent inadvertent changes. It is often useful to have two windows open on the screen, one with the raw text, and one for the concordancer, so that larger sequences of raw text can be identified and extracted if necessary: the concordancer cannot extract text sequences longer than 100 characters as a context of the target word or phrase.
3. Working on version #3, insert tags as appropriate (see e.g. the sample file below).
4. Save the file as "text only," which will strip all formatting from the file (e.g. layouts, formats, font specifications, bold, italic and so on).
5. Open the Simple Concordance Program.[3]

   From the File menu select New.

   In the window called "New scp Project" use the dialogue box in the right upper quadrant of the screen select the folder which contains the .txt files that you want to analyze. Click on the folder that contains the Simple Concordance Program software and the file tryscp.txt that you have created.

   A list of .txt files appears in the dialogue box in the left upper quadrant. Select (highlight) tryscp.txt. Below them is a button "Add selected files to the project." Click the button.

6. Click OK (top right of screen)

   If you want to save the project, give it a name with the .scp suffix in the Title line:

   MyProject.scp

   Click Save. Otherwise click No.

   Click OK.

   You are now at the analysis window.

7. To create a concordance,
    a. select "Ascending Alphabetic Order" (top centre of the screen);
    b. Select "Concordance" from "Concordance ... Word List ... Statistics" (centre screen);
    c. click on the KeyWords button at the left-hand margin of the screen just above "Concordance:";
    d. click on "Concordance" from the two buttons "Index ...Concordance" (centre screen);
    e. click on "Kwic" under "Statistics."

    Simple Concordance Program will build and display the result. You can now experiment with different options. To select the width of text to be displayed with the key words, use the Tools menu.

3.4 Investigation using Simple Concordance

The output of this initial concordance for the word from looks like this:

(22)

from			
1		From:	Fred QQAuthor//Hi QQGreet
3	//Hi QQGreet/ Good to hear	from	you. I was beginning to worry
15	collect the kids QQColloq	from	school QQSynEllipsis--gotta
21	QQDiminutive///(reply)//	From:	Jean QQAuthor//Freddy QQDiminutiv

The line numbers in the original text are in the left column. The search word "from" is centred, and up to 100 characters of context is included where relevant on both sides of the search word. We will not devote further attention here to interpreting concordance outputs. For this, there are ample guides in the literature (Hunston, 2002; McEnery & Wilson, 1996; Sinclair, 1991; Wichmann et al., 1997). We will concentrate instead on aspects of Simple Concordance Program and its use which interact with the extraction and interpretation of the QQ tags.

QQ+Concordance allows us to check the consistency of the tag list. To extract a list of tags, first use Simple Concordance Program to build a word-list, sorting the word-list alphabetically. Then scroll down to words beginning in "QQ-," which also displays the line numbers. From this we can cut-and-paste to extract the following:

(23)

QQAbbrev	10,14,21,22,23
QQAuthor	1,19
QQBodylang	26
QQColloq	13,13
QQDiminutive	17,20

QQExclam	8
QQGreet	2
QQImper	24
QQSignature	17,27
QQSynellipsis	13,14,22
QQValediction	16

At this point the list can be scanned for inconsistencies, and the output list can be used to correct incorrect or inconsistent tags. More importantly, the interim list can be used for the recursive refinement of tag by revising the set of tags the better to fit them to the data and the goals of the analysis. It is possible to split tags which are covering too much ground in an indiscriminate way, or to create sub-tags:

(24) QQBodylang > QQBodylangOcular, QQBodylangGesture . . .

or to merge tags which overlap; we can also consider metatags (3.4.4). This process is covered in detail in source like Miles & Huberman (1994).

We can then investigate correspondences between text and text, tags and text, and tags and tags, with a view to investigating metatags and hierarchies (3.4.4), networks (3.4.5) and statistical correspondences (3.4.6). Unlike the case of NUD*IST and ATLAS.TI, QQ+Concordance requires the user to do this manually. The merit of QQ+Concordance is that it harvests the data in convenient and usable form.

3.4.1 Text < > text patterns and correspondences

Patterns and correspondences between items in the raw text can be investigated by running a concordance on the raw (untagged) text file. This is often the first step in developing hypotheses about patterns in the text. For instance, a search of the King James Bible for the key word "yellow" finds only four hits. Three refer to "hair," for instance:

(25) if the scall spread not, and there be in it no yellow hair, and the scall be not in sight deeper than the skin (Leviticus 13:32).

and one to the metal gold:

(26) Though ye have lien among the pots, yet shall ye be as the wings of a dove covered with silver, and her feathers with yellow gold (Psalms 68:13).

There are 66 occurrences of "golden," only one of which refers to colour rather than the metal:

(27) And I answered again, and said unto him, What be these two olive branches which through the two golden pipes empty the golden oil out of themselves

(Zechariah 4:12)?

We may then ask how colour references in the Bible relate to primary colour terms, and how they relate to metaphors or similes ("like the colour of x"): there are 320 hits for "silver." Except in rather small texts, however, it is likely that a concordance of the raw text will be suggestive rather than conclusive about patterns and correspondences: the 66 occurrences of "golden" are already numerous enough to be inconvenient for manual inspection. The next step is to insert the appropriate tags for colour terms and to use the concordancer to extract the tags for analysis.

3.4.2 Tag < > text patterns and correspondences

The QQ+Concordance instrument allows us to ask in which circumstances a particular form of words is used. It is also possible to quantify these occurrences, and to conduct investigations on the frequency and cross-correlation of these statistical results.

The check of tag < > text correspondences can be used to relate specific expressions to pragmatic, semantic, formal and other factors. In the sample text included at the end of this paper, a concordance check on abbreviations with the QQ code reveals (with line numbers):

(28)

qqabbrev

10	still working on it. BTW	QQAbbrev,	I can't find the article
14	- gotta QQColloq	QQAbbrev	rush QQSynEllipsis.
21	Freddy QQDiminutive, THNX	QQAbbrev	for the update. Short
22	. And see his ref	QQAbbrev	to the paper by Kruszewski
23	by Kruszewski with info	QQAbbrev	about handling formulae

Even in this simple example, we can see that abbreviations (QQAbbrev) occur with acronyms (*BTW*) and chat-type shortenings (*THNX* = thanks), with two more common abbreviations used within professional communities (*ref, info*), and with two other tags: QQColloq and QQDiminutive (3.4.3). This information linking tags and text suggests a variety of further lines of investigation: What other abbreviation types are used in a larger text? Are they professional or not? There are numerous possibilities. We can test the co-occurrence of specific words or phrases with tags denoting speech acts, or of their links to agents (Silverman, 1993). Or we can link speech events to speakers: for instance, the well-known reluctance in some Confucian-based cultures of East Asia, including Japan, China, Korea, Vietnam and Cambodia, to say "no" directly to an interlocutor. Conversely, we can ask, for a particular circumstance, what different forms of words are used, and how often. Consider the concordance output for the tag QQSignature:

(29)

qqsignature

17	Cheers QQValediction Fred	QQSignature	QQDiminutive(reply
27	out. *hug* QQBodyLang/Jean	QQSignature	

Fred uses a formal valediction; Jean does not, as do many regular and fast users of email. The conventions of email discourse (Cherny, 1999; Niesten & Sussex, In press) can be richly investigated with QQ+Concordance.

3.4.3 Tag < > tag correspondences

Co-occurrences and frequencies of tags are a potentially powerful tool for investigating the structure and dynamics of texts. Simple Concordance Program does not allow searches for a tag within a specified word-length of another tag, as some more advanced concordancers can do. But it does allow us to search for a tag and to inspect which other tags occur in the same line, or in close proximity: the latter may require us to have a second window open on the screen with a display of the original raw text, ideally with line numbers to help in correlating the output of the concordancer with the original text. The example above in 3.4.2 shows QQAbbrev co-occurring with QQColloq and QQDiminutive. It also shows different patterns of syntactic ellipsis (QQSynEllipsis). Here we have set the context to 100 characters on either side to capture the wider context:

(30)

qqsynellipsis

13	Must collect the kids QQColloq from school QQSynEllipsis--gotta QQColloq QQAbbrev rush QQSynEllipsis
14	QQSynEllipsis--gotta QQColloq QQAbbrev rush QQSynEllipsis. More later Cheers QQValediction Fred
22	update. Short article's attached in PDF format QQSynEllipsis. And see his ref QQAbbrev to the paper

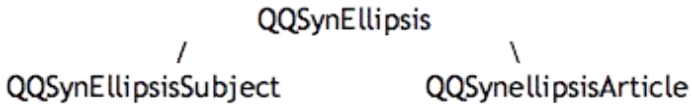
Here two colloquial tags in line 13 alone, and another in line 14, suggest a correlation which is worth exploring more widely, as does the co-occurrence with QQAbbrev in line 22.

3.4.4 Metatags and hierarchies?

Hierarchies involve the inclusion of tags within other tags. Theory-driven text analysis tends to start with high-level categories and then to investigate the structures beneath. Inductive, bottom-up grounded approaches tend to start with lower-level tags, though they can build both hyper- (higher-level) and hypo- (lower-

level) tags. It would be possible, for instance, to break QQSynEllipsis down into

(31)



(relating to lines 13-14 and line 22, respectively). Such structures are built through the recursive refinement and structuring of tags as the tagging process proceeds (Miles & Huberman 1994, chapters 4-5). Unlike NUD*IST and ATLAS.TI, such structures have to be built manually outside the QQ+Concordance framework, which offers no way of graphing or otherwise representing them visually. You use QQ+Concordance to extract the data and numbers, and feed them into a standard graphing package like Excel.

3.4.5 Networks

A major feature of qualitative analysis, and qualitative analysis software, is the building of networks of the kinds most frequently encountered in relational databases. Probably the most extensive and extensively realized semantic network is Wordnet (http://wordnet.princeton.edu), a very large and multilingual lexical database organized in intersecting networks. It provides both hierarchical links like ako ("a kind of") and meronyms (A is a part of B, a component of B, a material of which B is composed); and flat links like synonyms and antonyms.

The QQ notation also allows the building of such networks. The QQ tag names can reflect network structure by sharing common names or parts of names; or the network can be built independently outside the QQ+Concordance framework. For instance, QQAbbrev and QQColloq may be part of a network of informal language use. Some tags may belong to more than one network: A QQUSA tag could be part of an ethnic network, a sporting network, a political network, and so on.

3.4.6 Statistics

Concordance packages differ in the volume and variety of statistical information which they present. Most offer word counts, so that at least raw frequencies, and ranked raw frequencies, can be automatically extracted. Simple Concordance Program also generates cumulative statistics, so that it is possible to see, for instance, what portion of a text is consumed by words which occur only once, and so on. More elaborate outcomes, like standard deviations and even averages, however, have to be calculated from the raw figures that Simple Concordance Program provides. It is possible--for instance, by editing a file to contain only the utterances of a single speaker--to arrive at quite sophisticated analyses of different aspects of the use of language by individual speakers.

4. Limitations And Problems With The QQ Notation

The QQ notation and the analysis routines which we have proposed here are fairly modest in scope. They do allow the progressive and recursive development of patterns of tags, or metatags, to capture different kinds of phenomena (formal, pragmatic, social, cultural, etc.) in textual datasets. And outside the computer environment provided by the word-processor / text editor + concordancer, it is possible to build complex hierarchies and networks of related phenomena, based on rich qualitative and quantitative raw data, in research projects of very substantial size and scope.

The software configuration described here, because of the limitations of Simple Concordance Package, does not readily allow

a. the building of graphic trees to map patterns of tags and their concepts
b. searching for tag x in the context of tag y (e.g., search for tag x within 4 words of tag y, search for tag x where tag y does NOT occur within 10 words)

Such features are available with either more advanced concordancers, or with more fully featured software like NUD*IST or ATLAS.TI.

On the other hand, there is a great deal that can be done with the QQ notation in the current framework. It does allow

a. the recursive refinement of a structured system of tags
b. the qualitative investigation of tag correlations
c. the qualitative investigation of the relations of tags to raw text
d. the quantitative investigation of tags, together with the quantitative investigation of aspects of raw text

More complex quantitative analysis can be pursued with more advanced statistical tools (Burrows, 1987), using data derived from QQ+Concordance.

In addition, the QQ notation can be used for language teaching. For example, it could be used to encourage students to investigate the structure of texts and to explore ideas about the differences between texts. Giving students a tagged text and inviting them to use the concordancer to explore its structure and characteristics is valuable. Having students start this process from the beginning by tagging an untagged text, and developing and refining a tagset, is a more advanced task for the development and recursive refinement of descriptors to characterize chunks of consecutive language.

There are several drawbacks to the use of in-line notations:

(A) If the tag interrupts a phrase, it is no longer possible to search for that phrase as a linear sequence, whether in a text editor, a word-processor or a concordancing program. For instance,

(32) the girl with the green qqColourAdj eyes

cannot be searched as "green eyes." However, the phrase "green eyes" can be searched in the original raw data file. The best solution here is to have two windows open on the screen, one with the output of the concordancer and one with the raw text, with the lines numbered. This allows relatively quick recovery of fuller text content and contexts.

(B) in its present form the QQ notation is not suitable for working with conversational transcripts where overlaps, concurrent speech and interruptions are common. The current format of QQ notation works best with a single linear speech sequence. It is possible to imagine extensions of the QQ notation which would incorporate simultaneous speech by marking overlaps, pauses and so on with QQ tags. But whether this would be as perspicuous as more standard notations is not so easy to judge.

5. Conclusion

The QQ+Concordance system is relatively easy to master and use, and is relatively cheap. Most users will already have a word processor capable of storing files in Text-only format, and Simple Concordance Program is available as a free download. Since Simple Concordance Program also runs on both Windows and Macintosh platforms, it is as portable as any non-UNIX mainstream software package currently available.

QQ+Concordance is best with written texts and with monologues or single-voice texts, or with conversations like emails with clearly delineated turns. In the form presented here, QQ+Concordance refers to "language material to the left of the tag." However, it would not be difficult to modify it to deal with spans of multiple words, with paired start / end tags like

(33) qqOrderStart . . . qqOrderEnd

which would syntactically parallel the kind of formalism offered by HTML

(34) <tag> . . . </tag>

QQ+Concordance could possibly be adapted to handle multi-voice conversations with interruptions and overlaps, like the Jefferson formalism (above). But in-line formalisms are inherently less elegant and less visually transparent than layouts like Jefferson's, and it remains to be seen whether the issues of simultaneity, overlap, and interruption can be adequately handled by QQ+Concordance. On the other hand, QQ+Concordance can certainly handle pauses, hesitation, and similar phenomena by simple QQ-tags like

(35) QQPause05ms

for a pause of 5 milliseconds. The only requirement is that the concordancing

package must be able to accept numerals as characters which can be included in words.

The relative power of QQ+Concordance vis-a-vis multi-line formats like HIAT and Shoebox needs to be investigated in more detail. QQ+Concordance cannot conveniently handle parallel raw text and glosses, as can Shoebox, though one could imagine something like

(36) houses qqGlossRootHouse QQGlossAffixs

for the Shoebox

(37) house

house +-s

On the other hand, HIAT is now no longer available in a current operating system, since it is restricted to DOS. And Shoebox, apart from having an outdated Macintosh implementation, is not easy to drive. For some less elaborated analyses, it is arguable that QQ+Concordance will be faster and as perspicuous.

When compared to ATLAS.TI and NUD*IST it is clear that QQ+Concordance belongs to a different and lesser level of intellectual tools. It lacks the explicit organization and visualizations of features and hierarchies of these two more advanced formalisms, and its theory-building and testing capacities are partly transported outside the software for manual development by the researcher. On the other hand, both ATLAS.TI and NUD*IST are expensive, and they have significant learning curve costs before significant work can be done in analysis and interpretation. All three formalisms require the manual allocation of tags, though all three can handle some semi-automatic tag allocation (see above). It is easier, cheaper and quicker to develop both preliminary heuristics and a significant level of analytical understanding with QQ+Concordance. The process of developing a preliminary hypothesis, inserting trial tags, checking with the concordancer, and then revising the tags, is an important component of understanding how texts work and how to go about accessing what they have to tell us.

This accessibility means that QQ+Concordance also has significant potential for teaching purposes. While learning to use this formalism is not trivial, language students, in both first and second languages, can use QQ+Concordance to explore the structure, content and interpretation of texts, including their own writing. The ability of concordance packages to report on raw text occurrences, frequencies and contexts has already been shown to be of major value in language learning and teaching (Wichmann et al., 1997). This is further enhanced by the ability of both standard word processors and Simple Concordance Program to handle a variety of fonts and writing systems. The further capacity to extract parallel information about tags opens an avenue of investigation which has not been as easily accessible to either teachers or students of languages.

A Sample File

The following text can be used to test the QQ notation in its operation with a concordancer like Simple Concordance Program.

Raw data

The raw data are:

(a) message

Writer: Fred

Hi

Good to hear from you. I was beginning to worry that you might be sick or something.

I've nearly finished the first draft of our paper. Your idea about presenting the model first hasn't worked, so I have revised the order and put the data first. I hope that's all right. Let me know what you think.

But the formulae--AAARGH--are a terrible problem for the software. I'm still working on it.

BTW, I can't find the article by Short that you mentioned last week. Have you got a copy and can you send me one? Don't bother if it's too much trouble.

Must collect the kids from school--gotta rush.

More later

Cheers

Fred

(b) reply

Writer: Jean

Freddy,

THNX for the update. Short article's attached in PDF format. And see his ref to the paper by Kruszewski with info about handling formulae in word-processors.

Do send a draft of the new paper--I'd like to see how it's working out.

hug

Jean

The tagged corpus is presented below. The QQ tags used are taken from a list of tags which have proved useful in different pieces of my own research. Other users might formulate the tags differently, or use different tags for different purposes. The abbreviated labels in the tags are designed to save space, but one can just as easily use full words. The meanings of the abbreviated tags are:

QQAbbrev	abbreviation
QQAuthor	author
QQColloq	colloquialism / colloquial language
QQDiminutive	diminutive
QQExclam	exclamation
QQGreet	greeting
QQImper	imperative
QQSignature	signature
QQSynEllipsis	syntactic ellipsis
QQValediction	valediction

To use the tagged corpus to test the QQ + concordance formalism, select the text material between the marks vvv and ^^^ and copy it into a new word processing or text editing file. Then proceed as detailed in 3.3 above, "Using the QQ notation with Simple Concordance."

vvv

> From: Fred QQAuthor
>
> Hi QQGreet
>
> Good to hear from you. I was beginning to worry that you might be sick or something.
>
> I've nearly finished the first draft of our paper. Your idea about presenting the model first hasn't worked, so I have revised the order and put the data first. I hope that's all right. Let me know what you think.
>
> But the formulae--AAARGH QQExclam--are a terrible problem for the software. I'm still working on it.
>
> BTW QQAbbrev, I can't find the article by Short that you mentioned last week. Have you got a copy and can you send me one? Don't bother if it's too much trouble.

Must collect the kids QQColloq from school QQSynEllipsis--gotta QQColloq QQAbbrev rush QQSynEllipsis.

More later

Cheers QQValediction

Fred QQSignature QQDiminutive

From: Jean QQAuthor

Freddy QQDiminutive,

THNX QQAbbrev for the update. Short article's attached in PDF format QQSynEllipsis. And see his ref QQAbbrev to the paper by Kruszewski with info QQAbbrev about handling formulae in word-processors.

Do send a draft of the new paper QQImper--I'd like to see how it's working out.

hug QQBodyLang

Jean QQSignature

^^^

Bibliography

Allwright, D. & Bailey, K. M. (1991). *Focus on the language classroom: An introduction to classroom research for language teachers*. New York: Cambridge University Press.

Barry, C.A. (1998). Choosing qualitative data analysis software: Atlas/ti and Nudist compared. *Sociological Research Online, 3*(3). Retrieved 28 October, 2005 from http://www.socresonline.org.uk/socresonline/3/3/4.html.

Burrows, J. (1987). *Computation into criticism: A study of Jane Austen's novels and an experiment in method*. Oxford: Clarendon Press.

Cherny, L. (1999). *Conversation and community. Chat in a virtual world*. Stanford, CA: CSLI Publications.

Collins, J., Kaufer, D., Vlachos, P., Butler, B. & Ishizaki, S. (2004). Detecting collaborations in text: Comparing the authors' rhetorical language choices in the Federalist Papers. *Computers and the Humanities* 38(1), 15-36.

Ehlich, K. & Rehbein, J. (1976). Semi-interpretative working transcriptions (HIAT). *Linguistische Berichte* 45, 21-41.

Glaser, B. G., (Ed.). (1995). *Grounded theory 1984-1994*. Volumes 1-2. Mill Valley, CA: Sociology Press.

Hunston, S. (2002). *Corpora in applied linguistics*. Cambridge: Cambridge University Press.

King James Bible. (1611). [online]. Available: http://www.gutenberg.org/etext/10.

Kretzschmar, W.A. (2001). Literary dialect analysis with computer assistance: An introduction. *Language and Literature*, 10(2), 99-110.

McEnery, T. & Wilson, A. (1996). *Corpus linguistics*. Edinburgh: Edinburgh University Press.

Miles, M. B. & Huberman, A. M. 1984. *Qualitative data analysis: A sourcebook of new methods*. London: Sage.

Niesten, R. & Sussex, R. (In press). Negotiated meaning and ludicity in Internet chat. To be published in L. van Waes, C. Neuwirth, & M. Leitjen (Eds.), *Writing and digital media*. Oxford: Elsevier.

Sacks, H., Schegloff, E. A., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language 50*, (4), 696-735.

Silverman, D. (2005). *Doing qualitative research: A practical handbook*. London: Sage Publications.

Sinclair, J. (1991). *Corpus, concordance, collocation*. Oxford: Oxford University Press.

Sussex, R. (in preparation). *A Dictionary of Australian diminutives*.

Wichmann, A., Fligelstone, S., McEnery, T., & Knowles, G. (Eds.). (1997). *Teaching and language corpora*. London: Longman.

Notes

[1] The QQ+Concordance tools were initially developed for research by two groups: (a) H. Kim, R. Sussex and K.A. Yu, Intercultural communication on the Internet: A case study of Koreans and Australians, a research project funded by the Korea Research Foundation Grant (KRF-2004-042-A00073), whose support is gratefully acknowledged; and (b) members of the Research Foundry, a research group in language, society, culture and technology in the School of Languages and Comparative Cultural Studies at the University of Queensland.

[2] There are some special technical and discipline-specific uses of "QQ": to designate the class of all rational numbers, and in some transcriptions of Chinese

characters. These exceptions, however, are marginal enough not to be a problem for the QQ+Concordance package in regular use.

[3] If the message "Cannot initialize the help system, aborting" appears, ignore it and click OK: it seems to have no effect on the ability of the Simple Concordance Program to function properly.

WWW Sources

BNC (British National Corpus):

http://www.natcorp.ox.ac.uk/sara/index.html

http://homepage.mac.com/bncweb/home.html

CLAWS:

http://www.comp.lancs.ac.uk/ucrel/bnc2/bnc2guide.htm

http://www.comp.lancs.ac.uk/computing/research/ucrel/claws/

Concordancers:

http://www.nsknet.or.jp/~peterr-s/concordancing/specs.html

HIAT:

http://www.daf.uni-muenchen.de/HIAT/HIAT.HTM

Jefferson's transcription formalism:

http://www.sscnet.ucla.edu/soc/faculty/schegloff/TranscriptionProject/

NUD*IST:

http://www.qsrinternational.com/

Simple Concordance Program:

http://web.bham.ac.uk/a.reed/textworld/scp/

http://www.textworld.com/scp

About the author

Roland Sussex, is Professor of Applied Language Studies in the School of Languages and Comparative Cultural Studies at the University of Queensland in Brisbane, Australia. sussex@uq.edu.au

Editor's Note: The HTML version contains no page numbers. Please use the PDF version of this article for citations.