Published in: Revista Canarias de Estudios Ingleses 65, November 2012, pp. 115-130.

APPRAISAL ANALYSIS AND THE COMPUTER

Mick O'Donnell
Universidad Autónoma de Madrid

Abstract: This paper will explore how software programs can help in the performance of linguistic studies using Appraisal Analysis, both in the maintenance of a coding scheme, annotation of texts using that scheme and statistical studies base don the annotation. Special emphasis will be placed on how to produce visualisations of appraisal-annotated data. Most of the discussion will be based around UAM CorpusTool, software for corpus annotation and exploration.

Keywords: appraisal analysis, corpus annotation.

Resumen: En este trabajo se explorará cómo puede ayudar los programas de software en la realización de estudios lingüísticos mediante el Análisis de Evaluación, tanto en el mantenimiento de un esquema de codificación, la anotación de los textos que utilizan ese sistema y los estudios estadísticos fundamentada en la anotación. Se hará especial hincapié en cómo producir visualizaciones de evaluación anotados los datos. La mayor parte de la discusión se basa en la UAM CorpusTool, software para la anotación de corpus y la exploración.

Palabras claves: análisis de appraisal, anotación de corpus.

1 Introduction

Since the 1990s, there has been growing interest in means of assessing the attitudes expressed in text. This has occurred under several names, including Appraisal Analysis (Martin and White 2005), Evaluation (Hunston and Thompson 2000), Sentiment Analysis (e.g., Das and Chen 2001) and Opinion Mining (e.g., Dini and Mazzini 2002).

To study the attitudes expressed in text, one can take an 'anecdotal' approach, just picking out the tokens expressing attitude, and building these into a story as to what the writer was doing. This approach however can be easily abused, ignoring tokens which do not support one's theory. Following this approach, one could produce two contradictory analyses of a text, each drawing on different sets of tokens within the text.

To avoid the problems of an anecdotal approach, one must be more systematic in the identification of the attitudinal tokens. All tokens which fall within the scope of your study should be identified, classified and counted. Only on the basis of such an approach can meaningful findings be reported.

Early studies of attitude used pen and paper: the analyst would read the text, perhaps circling each attitudinal token, perhaps using coloured highlighting pens to distinguish different types. When the analysis was complete, each type of appraisal could be counted, and some interpretation of the results then made.

This approach however leads to difficulties when one finds errors in the codings (it may not be possible to erase the highlighter to change the colour), or if one changes one's coding criteria after coding a few texts (leading to some texts being marked up using different criteria than others). The different statistics derived from the text may all need to be recalculated. It is also difficult to share one's analysis with others interested in the analysis.

All of this argues for using the computer for analysis. Digitally stored texts can be marked up, and this mark up can be changed at any point if the analyst finds an error, or changes their coding criteria. Additionally, analyses can be shared across the Internet.

The most basic approach to attitudinal analysis using a computer involves the use of a word processor or spreadsheet. Using a word processor (e.g., Microsoft Word or Open Office), one can play with text styles to indicate the appraisal category, for instance, colour coded for the major type, and bold vs. italic to indicate positive or negative appraisal. This approach is limited however where more than two or three dimensions need to be identified (e.g., type of appraisal, object of appraisal, strength, polarity, explicitness, etc.) or where the range of types identified requires more colours than can be readily distinguished by eye. Additionally, counting of tokens in each category still needs to be done by eye, which is time consuming, and these counts need to be updated whenever any token is re-coded.

An improvement over the word processor approach is to use a spreadsheet (e.g., Microsoft Excel). Here, as the analyst creates a spreadsheet. As the analyst discovers an appraisal token in a text, it is copied and pasted into the text column, and other columns can be used to record its categories across multiple dimensions. The analyst can now modify the coding of each token without problem, and the spreadsheet can be made to provide counts of each category.

This approach also has problems. Each token is taken out of its original context of occurrence within a text, and the reason it was coded as it was may not be clear out of that context. Additionally, the colour visualization of the word processor document is lost, so seeing how attitude patterns change through a text is not possible. Additionally, some statistics will be difficult to derive. For instance, one important statistic is the number of attitude tokens per 1000 words of text, and if the original text is not kept, this statistic cannot be derived.

The best solution today is to use text annotation software for marking up texts. This paper will outline how such software can be used for exploring attitude in text, from design of one's scheme of attitude features, to the actual annotation of text, performing statistical studies of attitude, and visualization of both patterns within a single text, and of patterns over a corpus of texts.

While the techniques here are applicable to any theory of attitude, for the rest of the paper I will demonstrate using Appraisal Analysis (Martin and White 2005). Of the three components of this approach, I will focus only on Attitude, ignoring Engagement and Graduation. It should be clear how the techniques demonstrated for Attitude could be applied to the other two components.

2 Annotation of Appraisal

Tools for text annotation usually provide an interface presenting the text, and the user either clicks at a location in the text, or selects the text to be tagged, and then selects the category or categories to assign to the selection.

During the 1990s, various tools for text annotation were introduced, but most of them came with a pre-loaded coding scheme, and consequently they could not be adapted to a description they were not designed for. Since none of these were designed for appraisal analysis, none were of use for the present task.

Because of the lack of adaptability of such tools, in the early 1990s I started development of a program called *Systemic Coder*, which allowed linguists to design their own annotation scheme, annotate texts, and then perform statistical studies on the annotations (AUTHOR REMOVED 1995). The current incarnation of the software is called *UAM CorpusTool* (AUTHOR REMOVED 2008). The program is free, and works under both Macintosh and Windows. It has been designed to be as user friendly as possible, so that even those not so technically talented can manage it.

This paper will thus explain how to use this software for the annotation and study of Attitude in text.

2.1 Getting Started

UAM CorpusTool can be downloaded from http://www.wagsoft.com/CorpusTool/. Follow the instructions to download and install it on your machine. This chapter assumes version 3.0 of the software.

Once installed, launch the software. You will be presented with a window offering an option to start a new project, and another to open an existing project. For this paper, press "Start New Project". The windows that follow will allow you to specify a name for the project and where to save it. Once completed, you will be presented with the main window of your project.

2.2 Adding a Layer for Attitude

Most corpus annotation software allows you to annotate texts at multiple layers. Each layer can be a rank (e.g., clause, group, or word), or layers might be on different linguistic stratum, e.g., one layer for the Register of the text, another for semantic units (e.g., turns, moves and exchanges in dialogue), another for grammatical units, and another for orthographic units (e.g., paragraph and sentence in writing), etc.

For the present demonstration, we will start with one layer, for Attitude. At the top of the screen, there is a button "Add Layer". Press this button, and create a new layer, calling it "Attitude", specifying "Annotate Segments", and on the next window, select "Plain text Segments". On the final window, where it asks which coding scheme to use, click on the "Copy Existing Scheme" button, and then click on the "…". This will bring up a list of coding schemes included with the software. From this list, choose "UAM-Attitude.xml". Then click "Next" and "Finalise" to finish creating this layer.

You should now have a box in the top part of the screen with information about your new layer. To view the coding scheme for attitude, click on the "Edit" button. A window showing the Attitude coding scheme will be presented.

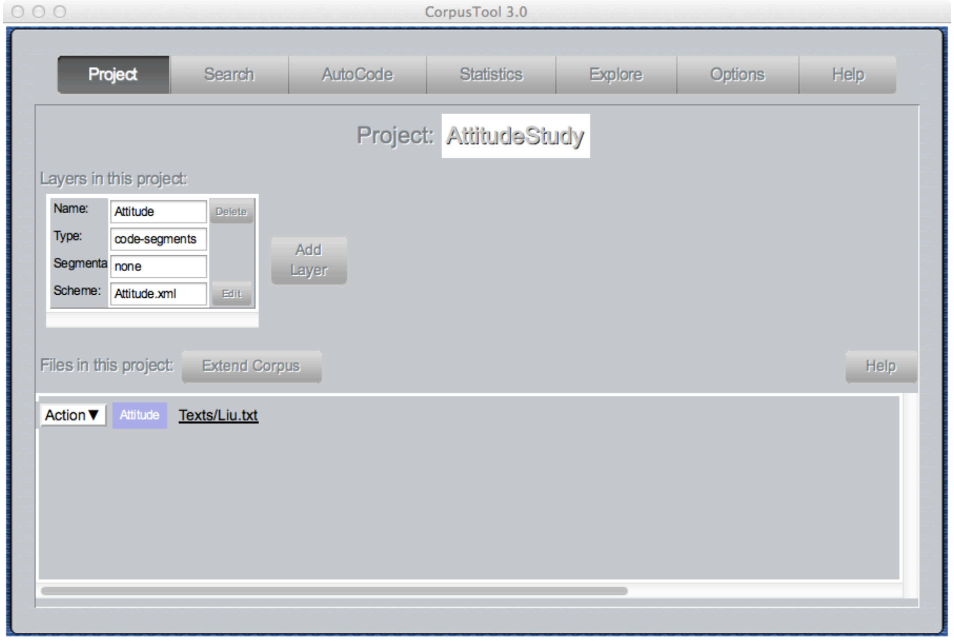
This particular scheme was developed by Lexie Don and Peter White and for use with UAM CorpusTool. This scheme can however be edited to suite your own understanding of Attitude: deleting systems, adding further systems, or changing the names of features. See the online help for details as to how to do this.

Note that many of the features in the scheme have glosses associated with them. These glosses will be visible while you are coding, and help you make the right coding decisions. You can edit these glosses in the scheme view to suite your own needs.

2.3 Adding Files to the Study

The next step requires adding a file to your project. At the bottom of the screen, you will see a button "Extend Corpus". Click this button to add a single text file to your project (see the manual for further directions).

Once you have added files to your project, each file will have a row of buttons next to it, one button for each annotation layer you have in your project. At this point, there should only be one button, for your Attitude layer. Fig. 1 shows how your project window should look at this point.



Project: AttitudeStudy

Layers in this project:

Name: Attitude
Type: code-segments
Segmenta none
Scheme: Attitude.xml

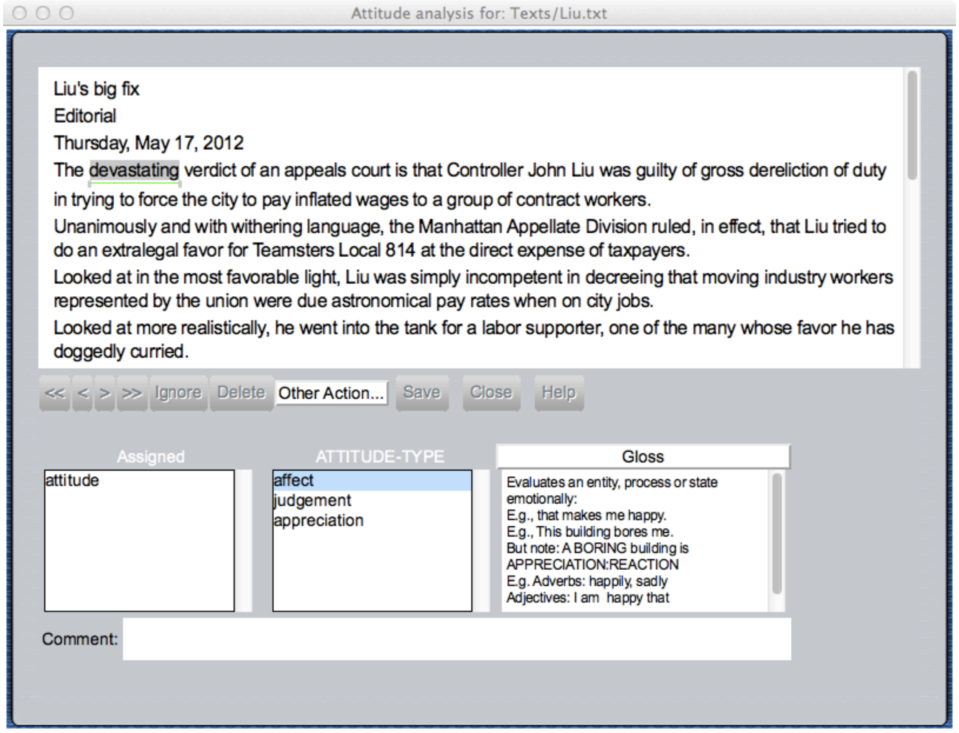
Files in this project:

Fig. 1: The Project Window

2.4 Coding Attitude tokens

Click on the Attitude button for one of your files. A window will appear, showing the text to be annotated. Read through the text, looking for words expressing attitude. When you find a word or phrase expressing attitude, select this text (click down at the beginning of the text and move the mouse to the end of the segment, then release).

The bottom of the screen will now show you three boxes. See Fig. 2. The first of these, labelled “Assigned”, shows the features so far assigned to this segment. Initially, this will only be the most general feature (leftmost on your coding scheme), *attitude*. In the middle box, you have a set of features to choose from, in this example: *affect*, *appreciation* and *judgement*. If you click once on one of these features, the gloss (if any) associated with this feature will be shown in the third box. You can double-click on a feature to ‘select’ it. It will then move to the ‘Assigned’ box, and a new choice will appear in the middle box. Double-click on a feature in the ‘Assigned’ box to undo that choice.


Attitude analysis for: Texts/Liu.txt
<< < > >> Ignore Delete Other Action... Save Close Help

Liu's big fix
Editorial
Thursday, May 17, 2012
The devastating verdict of an appeals court is that Controller John Liu was guilty of gross dereliction of duty in trying to force the city to pay inflated wages to a group of contract workers.
Unanimously and with withering language, the Manhattan Appellate Division ruled, in effect, that Liu tried to do an extralegal favor for Teamsters Local 814 at the direct expense of taxpayers.
Looked at in the most favorable light, Liu was simply incompetent in decreeing that moving industry workers represented by the union were due astronomical pay rates when on city jobs.
Looked at more realistically, he went into the tank for a labor supporter, one of the many whose favor he has doggedly curried.

Assigned	ATTITUDE-TYPE	Gloss
attitude	affect judgement appreciation	Evaluates an entity, process or state emotionally: E.g., that makes me happy. E.g., This building bores me. But note: A BORING building is APPRECIATION:REACTION E.g. Adverbs: happily, sadly Adjectives: I am happy that

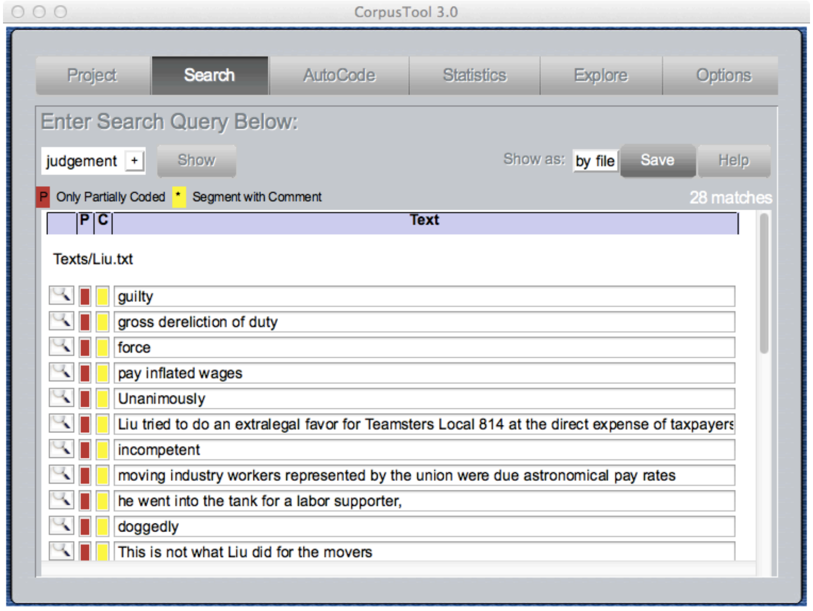
Comment:

Fig. 2: An Annotation Window

For your convenience, a table summarising the main Attitude categories is available under the "Help" menu on the main menu-bar: "View Appraisal Summary". Once you have coded some segments, press the "Save" button, then the "Close" button.

2.5 Basic Searching and Statistics

Once all documents are fully coded on the Attitude layer, we can perform various tasks. Firstly, we can retrieve all expressions of attitude of a particular type. For instance, we can retrieve all judgements by setting the search query to "judgement" and pressing the "Show" button. All judgement tokens will appear below. Search queries can be very complex, specifying to find all units within some unit, or to find those containing another unit, or a concordance string.



judgement +   Show as: by file

Texts/Liu.txt

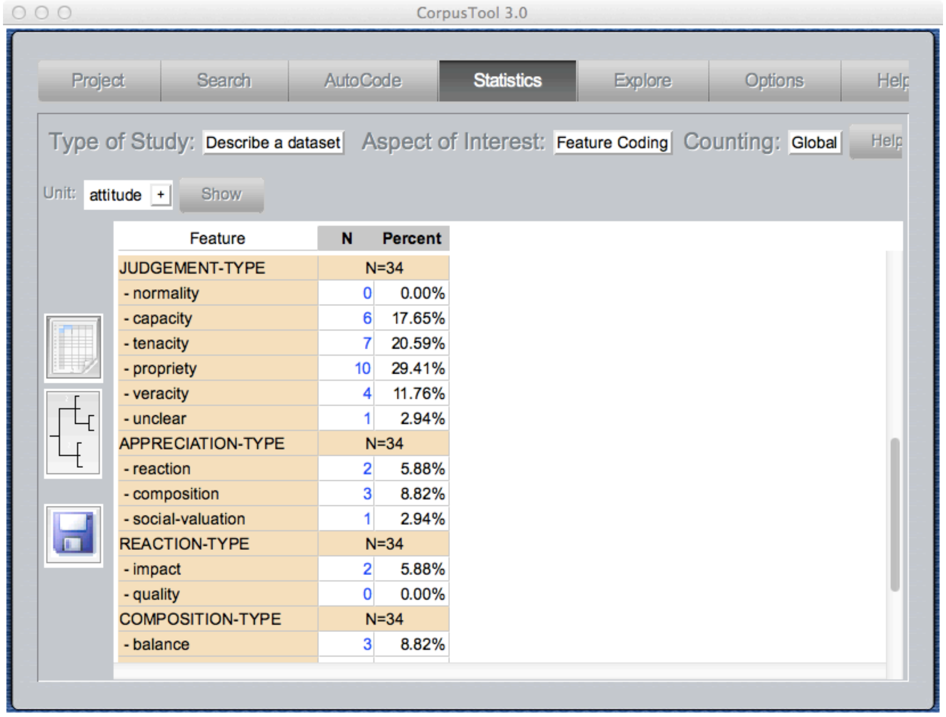
P	C	Text
		guilty
		gross dereliction of duty
		force
		pay inflated wages
		Unanimously
		Liu tried to do an extralegal favor for Teamsters Local 814 at the direct expense of taxpayers
		incompetent
		moving industry workers represented by the union were due astronomical pay rates
		he went into the tank for a labor supporter,
		doggedly
		This is not what Liu did for the movers

Fig. 3: The Search Window

The Statistics window is useful for getting descriptive statistics (statistics on a single text or corpus) as well as comparative statistics (statistical comparison of two or more datasets, indicating which differences between the sets are statistically significant). For instance, switching to the Statistics window, setting:

- **Type of Study**: Describe a dataset;
- **Aspect of Interest**: Feature Coding;
- **Unit**: attitude

Then press "Show". You will be presented with a table showing how much the texts in the corpus use each Attitude feature. One column presents the N (how often the feature was selected). Another column presents "Percent" (what percentage of all attitude have this feature). See Fig. 4.



Type of Study: Describe a dataset Aspect of Interest: Feature Coding Counting: Global

Unit: attitude

Feature	N	Percent
JUDGEMENT-TYPE	N=34	
- normality	0	0.00%
- capacity	6	17.65%
- tenacity	7	20.59%
- propriety	10	29.41%
- veracity	4	11.76%
- unclear	1	2.94%
APPRECIATION-TYPE	N=34	
- reaction	2	5.88%
- composition	3	8.82%
- social-valuation	1	2.94%
REACTION-TYPE	N=34	
- impact	2	5.88%
- quality	0	0.00%
COMPOSITION-TYPE	N=34	
- balance	3	8.82%

Fig. 4: The Statistics Window

Per 1000 words: This way of describing Attitude is useful to see what mix of the types of appraisal are used by the writer. However, it won't show whether the writer expresses a lot of attitude, or is relatively flat, expressing little attitude. To view this, change "Counting" to "per 1000 words", and press "Show". Now the table will show how often each attitude type appears in every 1000 words of text. For instance, the Liu text has 73.12 attitude tokens per 1000 words of text, and 60.22 of judgement tokens.

Describe each file: If you have a number of files coded, you can change "Type of Study" to "Describe each file" and press the Show button. You will be presented with one column for each text, and can compare the distribution of Attitude resources used by each writer.

2.6 Producing styled text output

In some publications on Appraisal, the researcher includes in their document a sample of the text being studied, using font size, bold, italic, underline, text colour, etc., to indicate different aspects of the analysis (e.g., colour for the type, bold vs. plain to indicate polarity, etc.).

You can produce this styling from your codings within *UAM CorpusTool*. Returning to the Project window, if you click on the filename of a file, a window will open presenting your text. At the bottom of a window is an area for adding text stylers. At first there will be just one, defaulting to *attitude*. Change this feature to *affect*. After the feature, there are a number of symbols, being: 'b' (bold), 'i' (italic), 'u' (underline), 18 (to change font size), 'A' (to change the text colour).

Use these widgets to change the style in which *affect* tokens are displayed. For instance, click on the 'A' and choose a blue colour. Then, click on the "Add" button to add a styler for *appreciation* (set to a red colour), and a third for *judgement* (use a green colour). Then perhaps add another for positive tokens (use bold).

You can save this diagram by pressing the "Save text" button. This will save it as HTML. If you then open this document up in MS Word or OpenOffice, you can save as a doc file, to use in your own papers.

3 Specifying the Appraiser: Heteroglossia

A single text may contain many voices. For instance, in "the Daily News reported Liu had falsely claimed to have worked as a child in a sweatshop", we have three voices:

1. The author of the article, who talks about the Daily News and what they did;
2. The Daily News, who report on what Liu said, and
3. Liu, who supposedly talked about his child labour.

In a sense, the writer of the article wrote the entire sentence, choosing the words to use, although hopefully corresponding to what these other voices did in fact say. With quoted speech however, the writer does not choose the wording for the other voices, although the writer is responsible for choosing which of the many utterances made by the other speaker are included in their article. What is important here is that the writer has chosen the set of attitudes that the other voices are expressing, and, to understand his characterization of these others, we need to explore the pattern of attitudes they express. The assignment of a particular set of attitudes to a person is part of the process of constructing the persona of that person in the text.

What we are talking about here is often discussed under the label 'heteroglossia'. It has relation to another aspect of Appraisal Analysis, called 'engagement':

> engagement is concerned with the ways in which resources such as projection, modality, polarity, concession and various comment adverbials position the speaker/writer with respect to the value position being advanced and with respect to potential responses to that value position – by quoting or reporting, acknowledging a possibility, denying, countering, affirming and so on. (Martin and White 2005 p36)

But note here that engagement is more concerned with the writer's position in regards to what the other voices say: distancing or supporting. In the current section, we are not concerned with the writer's position in respect to other voices, but rather with how these other voices differ in their attitudes (with the understanding that it is the writer who selects these attitudes for the voices).

Without exploring heteroglossia, we are at risk of confusing the writer's own attitudes with those of the other voices in the text. If we annotate the text in terms of who is 'saying' what, then we can explore the attitude sets of each voice in the text. The problem is: how do we annotate the text to separate these voices. There are two possibilities, explored in the sections below.

3.1 Annotating APPRAISER within Attitude

The simplest way is to add a new system to our Attitude network, in parallel with the attitude network, with options for “authorial-voice” or “other-voice”. Under “other-voice”, we could add a further system, and record the names of these other entities. See Fig. 5. Now, when one codes an attitude token, one is first asked about the type of attitude (and more delicate subtypes), and then one is asked *who* is doing this appraising.

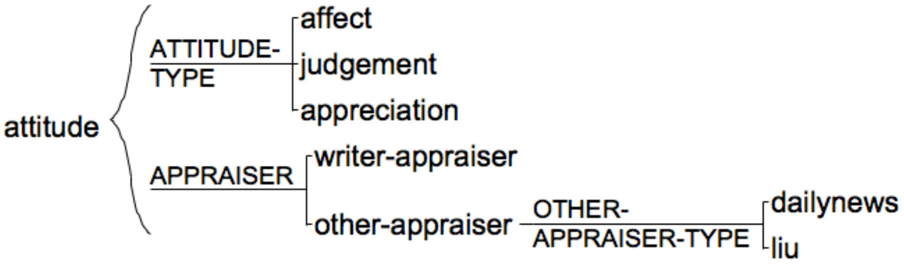


Fig. 5: Combining attitude and heteroglossic coding in one layer

If, during the coding of a text, you encounter a new voice, you do not need to return to the coding scheme to modify it. There is an option under the “Other Actions” menu to “Add new feature” to the current system.

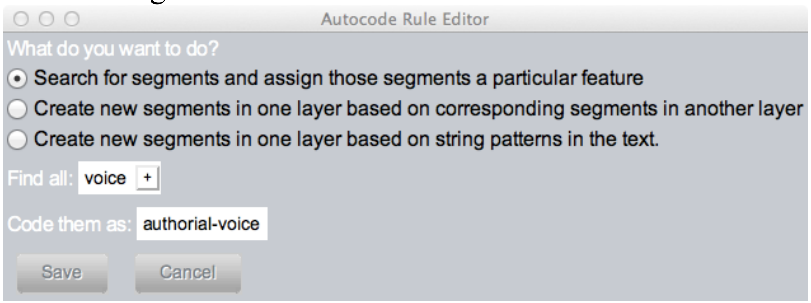
3.2 Adding an Appraiser layer

It is perhaps more theoretically correct to code the extent of each voice on a separate layer. We thus return to the Project window on the main interface, and add a new layer. This one we could call “Voice”. It should be specified as “Annotate Segments”. To save some time in annotation, specify “Automatic Segmentation” as “sentences”. This will mean that each sentence of each text in the corpus will be created as a ‘voice’ segment. Specify “Create new scheme” and finish.

You can then edit the scheme to create a network of the different voices in the text, e.g., *authorial-voice* vs. *other-voice*, and under *other-voice*, the different voices in the texts.

In the above example (with the Daily Mail and Liu), I was interested in specific voices in one particular text (or a small set of texts with recurring characters). In larger studies, one might be less interested in individuals but rather with classes of voices, e.g., government officials, union leaders, media reporters, etc. In this case, your “Voice” network could be a hierarchy of such roles.

To save some time, it would be useful to initially set all of the sentence segments to be *authorial-voice*. To do this, we switch to the Autocode pane. Click “Add” to create a new rule, and add a rule as in Fig. 6.



What do you want to do?

● Search for segments and assign those segments a particular feature

○ Create new segments in one layer based on corresponding segments in another layer

○ Create new segments in one layer based on string patterns in the text.

Find all: voice +

Code them as: authorial-voice

Fig. 6: Defaulting all voices to *authorial-voice*

Then click “Save” and then click the “Show” button, whereby the program will find all *voice* segments and display them. Then click “Code Selected” to code all of these as *authorial-voice*.

Now, we need to open the Voice annotation window for each text file in turn, and find places where the voice is not the authors. For such instances, move the segment boundary of the existing segment to just include the projecting part, and then add a new segment to cover the projection, e.g., the following sentence becomes two segments, the first coded as *authorial-voice*, the second coded as *other-voice*:

> The devastating verdict of an appeals court is that | Controller John Liu was guilty of gross dereliction of duty in trying to force the city to pay inflated wages to a group of contract workers.

Apart from quoted speech, it is not always an easy job to identify who is saying something. For instance, following the above sentence is another sentence evaluating John Liu's behaviour. It is not clear if it was something additional said by the court, or if it was something the writer is adding from their own knowledge. It may be worth adding an *unclear-source* option to your Appraiser scheme.

3.3 Searching and Statistics with Two Layers

Now we have two layers of analysis, we can demonstrate some of the power of search within *UAM CorpusTool*. The software allows you to specify a search making reference to multiple layers, for instance, to find all attitude tokens contained within particular segments on the Appraiser layer. Returning to the Search window, we can now retrieve all attitude tokens made by a particular appraiser. For instance, select "judgement" as the feature of interest. Then click on the small '+' icon next to it and select "in segment". The search query should now read: "judgement anywhere in attitude". Click on the word "attitude" and change this to one of the appraiser options. Then click show. You will be presented with all of the judgement tokens issued by that appraiser.

Layers can also be used on the Statistics interface. For instance, to view contrastively how different voices differ in their use of attitude resources, specify:

1. **Type of Study**: Compare two datasets
2. **Aspect of Interest**: Feature Coding
3. **Unit**: attitude
4. **Set 1**: authorial-voice
5. **Set 2**: other-voice (or one of the other voices)

Then press the "Show" button. The bottom window will show a table comparing the two voices in terms of how they use appraisal resources.

4 Exploring how different entities are appraised

It is often the case that a writer will evaluate one person, thing or activity positively, while evaluating other things negatively. The way they evaluate these entities may also vary, e.g., judging morally one person and evaluating aesthetically another. For instance, a news writer might cast Bill Clinton with negative moral propriety and veracity, while Nancy Clinton might be cast with a mix of negative happiness (affect), but positive tenacity (judgement). We as readers connect with a writer who expresses attitudes we share. Once connected, writers can start to introduce evaluations we might not otherwise agree with, shaping our opinions.

The same is also true for the other voices within the text: the writer can construct each other voice by making them positive to some entities, and negative to others.

By separating out the attitudes of a single voice (authorial or other) in regards to the target of the appraisal, otherwise murky data can clarify. In the case of the Clinton article above, if the analyst ignored the target, they would see the writer using quite a range of positive and negative attitude tokens. Once separated by voice and target, we can start to paint the picture of each voice's attitudes to different targets.

Exploring attitude according to target can be done in two ways:

1. **Naming individual targets**: for small studies of single text, or a set of texts based around the same people, we can add a system to our Attitude network with names of entities that are being evaluated. This system should be in parallel with the attitude system (click on *attitude* and select "Add System").
2. **Naming classes of targets**: for larger studies, over multiple texts, it may make more sense to use instead a hierarchy of classes of targets. For instance, if one is studying war reporting, one might include a hierarchy distinguishing ally vs. enemy vs. neutral participant, government vs. military vs. media worker vs. civilian, etc. Again, this should be parallel to the Attitude systems.

Once we have modified the Attitude network to include target, we can go back to our codings and add this category for each attitude token we have already coded. Open each Attitude window in turn, and go through each segment, adding only the features to code the target. If a token corresponds to a target not yet in the scheme, you can select "Add New Feature" from the "Other Actions" menu of the Coding window.

Once all tokens are coded, you can return to the Statistics window and perform studies to see which combination of attitude resources particular appraisers use to appraise each target. However, due to lack of space, this will not be demonstrated here.

4.1 Using the Explore Pane

The Explore pane offers various options for visualizing patterns within a text. If you have a layer using features "affect", "appreciation" and "judgement", then the program adds a special tab in the Explore window: "Explore Appraisal Data". Only some of these options will be explained here.

At the left, are a number of icons, which select different ways of visualizing your file (colour coding in the diagrams below may not show):

- **Text**: shows the text, and at the top choose between "Attitude Type" (affect, appreciation. Judgement), "Appraiser", "Appraised", "Polarity" and "Explicitness" (invoked vs. inscribed). As one clicks on each of these buttons, the text will be shown with segments color-coded according to the current category (e.g., if you select "Polarity" all positive segments are shown in one colour and all negative ones in another).
- **Table**: showing the counts of all features under the current category as a table.
- **Cloud**: presents a tag-cloud. Attitude features which are used more often in this text in comparison with the rest of the corpus are displayed more saliently. For instance, Fig. 7 shows a Cloud View for a text which is full of negative propriety judgement.

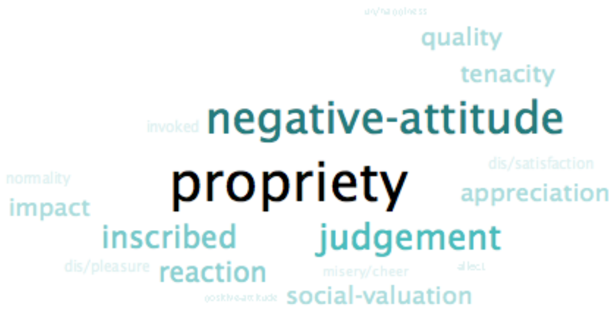


Fig. 7: Cloud View of a text

- **Stream**: A Stream View presents how the selections from a system change as the text progresses. For instance, Fig. 8 shows the Stream View with "Attitude Type" selected. Each colour represents a different attitude type: red: *affect*, green: *appreciation*, and blue: *judgement* (colours may not show in printed version). The diagram shows that *affect* (top) only comes into use about 2/3 of the way through the text, and at that point replaces the use of *judgement* (bottom) rather than *appreciation*.

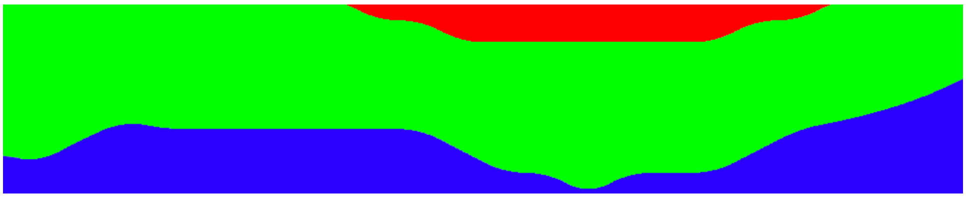


Fig. 8: Stream View of changing Appraiser through a text

- **Crosshair**: This diagram, showing how each voice uses Attitude types, was proposed by Peter White. The X-axis represents the degree of expression of 'impersonal' attitude (*affect* is taken as personal while *judgement* and *appreciation* are taken as impersonal, e.g., *I like it* vs. *It is likeable or He is fair).* The Y-axis distinguishes between appraisals on a mainly aesthetic basis (*affect* and *appreciation*) vs. appraisal on an ethical basis (*judgement*). For each text, an X and Y value in calculated on the basis of its overall usage of attitude resources, and it is plotted. You can also choose to show how each appraiser in the text positions himself in regards to each distinct target entity in the text, as has been done in Fig. 9. The dot near the centre represent Hugh Grant's publicist's view on Hugh Grant (a balanced mix of affect and judgement), while the dot high on the left reflects the author's position (judging morally and impersonally).

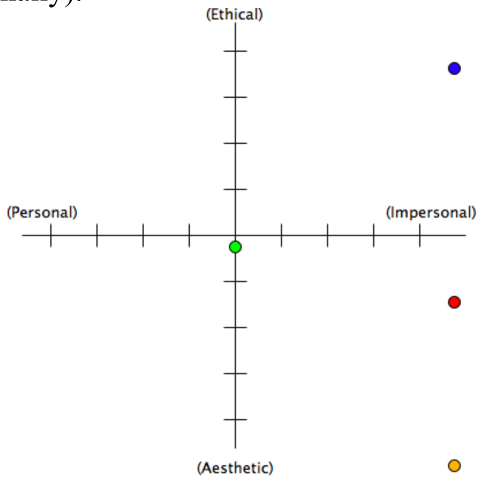


Fig. 9: Comparing two appraisers view of particular participants

5 Other Appraisal Studies

Above we have outlined some ways to study attitude using corpus software. There is a wide range of studies possible using Attitude coding. Below we outline some alternatives.

5.1 Explorations of a single text

- *Measuring the degree of attitude expression*: how frequently does the author and other voices appraise?
- *Measuring the Positivity of the writer*: are the attitude tokens of the voice on balance positive or negative?
- *Measuring degree of heteroglossia*: to what degree does the author open up the space to other authors?

- *Measuring ratio of inscribed vs. invoked attitude*: do more mature writers use more invoked attitude than novice ones?
- *Comparing the way different entities in the text are appraised*: e.g. does one character get judgement and another affect? Do males get appraised more via one type and females another? Does one class of target get overtly appraised more than another?

5.2 Comparison of two corpora

- *Comparison of two authors*: e.g., comparing the attitude resources used by Emily and Charlotte Brontë.
- *Comparison of two texts by one author*: e.g., comparing an early Shakespeare play with a later one.
- *Comparing two registers or genres*: e.g., comparing attitude expression in spoken vs. written versions of news, front page news vs. editorials, high-brow vs. low brow press, linguistic vs. computational linguistic articles, etc.
- *Comparing across social demographics*: comparing across gender, age, social class, sexual orientation, etc.
- *Comparing across cultures*: e.g., do Spanish authors use more attitude resources than English authors?
- *Comparing across proficiency levels*: e.g., do students change their usage of attitude resources as they progress in their mother tongue? In a foreign language?

5.3 Studying Attitude in relation to form

We can code each attitude token in regards to its syntactic form (is it an adjective or a verb, etc.). Additionally, we might code the attitude token in regards to its syntactic function (e.g., Subject, Complement, noun premodifier, etc.). The study would then explore whether information as to syntactic class and syntactic function can help resolve which attitude type the token expresses. For instance, 'good' expresses different attitudes in each of the following phrases: "a good man", "he is good", "he is not good", "I feel good", "a good measure".

UAM CorpusTool allows for automatic syntactic analysis of English text, which might be useful for this task (add a layer selecting Automatic Grammar Analysis). Alternatively, you might add extra systems to your Attitude network allowing you to manually classify each token by syntactic class and function.

6 Summary

This article has explored how to use corpus software to explore attitude patterns within a single text or a corpus of texts. It has focused mainly on the approach to attitude proposed by Martin and White (2005), and has demonstrated techniques in regards to *UAM CorpusTool*, since this software is free, relatively easy to use, and adaptable to the needs of the sentiment analyst.

Coding of attitude is time consuming, but using corpus software reduces the time needed, allows recoding of items. Also, cross-layer search allows retrieval of instances matching sometimes quite complex search queries. The use of search specifications in the Statistics interface allows the analyst to receive descriptive statistics of particular subsets of the data, or perhaps to contrastively compare two subsets.

The Explore interface of the software allows the analyst to view patterns within a text, both dynamically (how do attitude choices vary as a text progresses), but also in terms of how the different voices in a text vary in their 'rhetorical voice', in terms of the different attitude resources they use, and how they value the various entities mentioned in the text.

Development of the software is continuing, with new visualisations of appraisal patterns being introduced, both within a text, and across a corpus. Research into the automation of attitude tagging using large attitude dictionaries is currently underway, and may be incorporated into the software when more robust.

7 Works cited

Dini, L. and G. Mazzini. “Opinion classification through information extraction”. *Proceedings of the Conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields (Data Mining)*, pp. 299–310, 2002.

Hunston, Susan and Geoff Thompson, eds. *Evaluation in Text: Authorial Stance and the Construction of Discourse*. Oxford: Oxford University Press. 2000.

Martin, J.R. and P.R.R. White. *The Language of Evaluation, Appraisal in English*. Palgrave Macmillan, London & New York. 2005

O’Donnell, Michael. "From Corpus to Codings: Semi-Automating the Acquisition of Linguistic Features". *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford University, California, March 27-29, 1995.

O’Donnell, Michael. “Demonstration of the UAM CorpusTool for text and image annotation”. *Proceedings of the ACL-08:HLT Demo Session (Companion Volume)* 13--16. Columbus, Ohio, June 2008. Association for Computational Linguistics.