

Native Or Not? Golden Rules In Open Ended Coding

Contributed by Language Connect
Tuesday, 08 February 2011
Last Updated Wednesday, 16 July 2014

There is an uptick of interest among the web analysis community into how market researchers manage open ended coding to capture consumer comments and feedback. The research coding industry has some structural challenges and new software which has recently appeared on the scene may address some of these. In this post we outline some of the golden rules of coding, how to achieve quality and the most cost effective outcome.

Open ended coding refers to the task of analysing relatively unstructured feedback and comments made by people who have been interviewed about their opinions or behaviour. It is probably best described as a being both an art and a science.

In the past, coding has been a specialist part of the of market research toolkit and one which is now drawing an increasing amount of attention due to the volume of comments generated by social media, online surveys and other initiatives in co-creation, interaction and engagement.

One of the principal reasons for the interest is because analytical companies working on behalf of global retailers and online platforms such as Amazon, Twitter and eBay, are taking serious note of the skills and tools which let you analyse large scale consumer feedback quickly.

The software tools which handle open ended coding are maturing and, although semantic coding and contextual phrase analysers are still in their infancy, the research sector is vulnerable to disruptive technology from mainstream IT companies. A second tier of wishful thinking by global corporates such as Apple - the ability to translate consumer feedback on the fly into any language, looks to remain several decades away. The current state of acceptance is low, researchers and marketers do not accept that Google Translate and other machine translation programmes will be able to code customer comments which can be used in a meaningful and commercial way until multilingual context based and semantic translations become a reality.

In many research agencies, there is often a shortage of skilled coders to handle high volumes of work. The shortage of skilled practitioners has not been helped by poor access to training and the research coding arm of the industry is largely unregulated for quality and consistency. Many research execs and clients don't really understand how coding happens or how to get the best from the technique or its practitioners.

And this is just in the English language! Just imagine the challenges trying to manage open ended coding from customers in thirty languages!

Certainly, good project specification will go a long way towards solving some of these problems, so here are some golden rules in coding.

In language coding is by far the most efficient way of handling multilingual feedback.

Before starting work, think carefully about how to process your responses from more than one language. If you take time to specify the process carefully, you may find that you can save a good sum of money or at the very least, avoid the risk of costs spiraling at the end of the project.

Researchers often wait for all the open ended comments to be delivered at the end of fieldwork, then back translate all the comments into a single language (often English) and code the responses in a single language. The cost of back translating the comments is often relatively high and the method often incurs other losses caused by 'bottlenecking' - trying to process all the verbatim feedback in a rush before the client debrief. Also, it's not unusual that nuances of feedback are lost in back translation unless a highly skilled translator is used (which they unfortunately do not tend to be in this type of work). An advantage, however, is that a central client can read and understand all the responses in a single language.

A popular alternative is to code all the responses in their local language, thus cutting out the time consuming and expensive translation stage. Generally speaking, you can expect to save up to 80% of verbatim processing time and 40-50% of your coding processing costs by removing a back translation stage.

If your client wishes to see feedback in English (for example), the best practice workaround is to code all the responses in their native language and then back translate a small, representative sample into English. This method produces large scale cost savings and allows the client to read and understand a representative sample of replies.

Calculate the number of responses accurately.

The first thing to do is count the number of open ended questions and the likely volume of responses. Do not guess!

The response rate per question will have a big impact on your costs, so if your survey is likely to deliver a large volume of feedback, consider setting a maximum quota or code only a sample of the total responses.

There are three main question types and it's important to distinguish between them since they impact your marginal costs.

Long verbatim responses where the respondent is asked for their opinion about a brand or a service are, fairly obviously, expensive and time consuming to process. It's important to be clear how long you expect the responses to be and how many different points you anticipate the respondent might make per comment, which will need coding separately. Some customer satisfaction surveys get individual responses of several paragraphs, for example! A more common scenario from the packaged goods or services sector, allows for 2-3 sentence replies that might contain 3 to 5 different codes in each response.

Single word comments, typically a brand or product name with a high degree of predictability. "When you think of banks, what is the first bank name that comes to mind?"

'Other/Specify' questions, often single word or short answers. "Can you tell me one word that you think describes this product?" These catch-all questions are useful for picking up non precoded responses. Don't overlook them, as research analysts find them very useful for identifying emerging trends.

Use a unified code frame across multiple countries and questions.

A code frame is a numerical list of all possible responses from an open ended question. Generally the responses are put into a pre-prepared list and sometimes a short pilot test is run to gauge what the likely responses will be.

It is easiest to use what's called a "unified (or global) codeframe" where the same codes are used across all segments, countries and markets. This is best practice and prevents the possibility of duplicating answers – be very careful indeed about using separate brand code lists across different countries. However, some types of work may indeed need individual frames, such as where market and regulatory conditions differ greatly.

Since a significant element of the cost for coding is in the set-up of the code frame response grid for each question, consider designing a code frame structure which might form the basis of a frame across several questions.

Sometimes, new response codes are added as the survey progresses. If you wish to do this you will need to specify how many mentions an item needs to have before adding a dedicated response code. The process of adding new codes into a pre-existing codeframe is called 'back coding'. Back coding may be done when fieldwork and coding is in progress and although this is often quick and cost effective, it is arguably less accurate since earlier responses will not be coded against the later codes. Alternatively, back coding can be carried out at the very end of the coding process.

An increasingly popular approach is to group codes into thematic areas. You might consider designing your survey to ask respondents to self theme their comment themes. This works well in helping keep costs down when designing employee surveys, for example, but works less well for new product or concept testing where a consistent approach is required.

Self theming works by firstly asking the respondent for their unprompted open ended comment. Once they have answered, show the respondent a thematic list and ask them to select which theme they think fits their response best.

A more detailed approach is for the researcher to group comments into Nets in a code frame. Nets are top level category codes made up of individual codes that have a parent/child structure. Nets can refer to a likability factor, a common grouping pattern is Good, Bad and Average for example. Alternatively, Nets can be structured by the type of product or service. A shampoo survey may have responses netted by different shampoo brands, scents or formulations. It's a wise idea to allocate the codes from the ground up, and combine responses into Nets at the end of the process. Approaching coding purely from the top down means that you will not be able to unpick the Nets for subsequent analysis.

So, where does open ended coding go from here?

It's already a reality that coding software can automate lists of brand and single word answers fairly easily, enabling global marketers and media planners to feed market share calculations into CRM systems. However, the programming and automation of open ended customer comments is much more tricky, and still requires a human coder to check and code for context, tone and meaning using local expressions and native language. It's unlikely that software coding programmes are going to replace the man-machine interface anytime soon.

About Language Connect

As the market leader in the provision of language services to the market research industry, Language Connect delivers full-service translation, localisation, interpreting and verbatim coding services to a wide range of clients. With offices in the UK, Germany and Australia, our dedicated staff of account managers, project managers, localisation and technical

support specialists who are complemented by a network of over 5,000 subject specialists, work with clients on a local level to address specific language and cultural needs on a global scale.

Maggie Little +44 207 940 8110

mlittle@languageconnect.net

www.languageconnect.net

February 2011