Classifying Open Occupation Descriptions in the Current Population Survey

Fred Conrad

This is really a study about coder error. I think in some ways it follows Jim's talk about inference and validity.

I guess the first point shouldn't be a surprise to anyone given that we are all gathered here – open ended responses must be classified (coded) in order to be turned into quantitative data. This is subject to human error. It seems to me that it is subject to human error for two reasons. One because coding is similar to the way that people do every day classifications and the other because coding is different from every day classifications.

Looking first at why it is different from every day classification – coding lacks the flexibility and nuance that is part of making every day categorizations. For example, in every day classifications instances belong to a category to a matter of degree. There are graded memberships so an ostrich or a penguin are worse examples of birds than robins are. That is fine, they are all birds. We can accommodate graded membership to warrant less typical instances. Sometimes we can grow new categories when the existing category doesn't quite work for an atypical instance. There is a study by Kunda and Oleson in which they present people a description of Steve, the introverted lawyer. The idea is that any category for lawyers just doesn't have room for an introvert in it. The authors gave some participants additional information that allowed them to create an introverted lawyer category and found that for these participants the general lawyer category remained in-tact because they grew a new subcategory to handle these atypical categories.

The point is, coders can't do either of these things. They can't assign a description to a category to a matter of degree – it is either in or out – and they can't create new categories or subcategories. They can't add new digits to the classification scheme.

On the other hand, everyday classification and coding are similar and therefore coders are subject to the same kind of errors that we are. For example, consider Tversky and Kahneman's representative heuristic and their conjunction fallacy in which people are basically blinded by superficial similarities so that if the description *sounds* like a category member then people put it in the category, even if there is something that is more predictive like base rate. It turns out that this description sounds like an engineer but 80% of the instances in this experiment are lawyers there is a good chance the entity being described is a lawyer sounding like an engineer. People just don't seem to consider that. So coders can easily make the same kinds of errors that we do on an every day basis.

Let me just comment – I made this point that people can't develop new categories – coders can't develop new categories. But survey researchers who develop coding systems can and I just want to plug a paper by Jon & Neil Malhotra. It presents a very interesting method for developing new categories – in their case new biotechnology jobs. It seems like there may be some reason to discuss this as we go on.

Mick Couper and I investigated some of the ways coders might make errors at the Current Population Survey coding occupation questions. They ask two questions, "What is your occupation?" and "What kind of work do you do?" that is what are your usual activities or duties at this job?" sort of job title and duties? Typically the answers to both questions are coded by coders who sit in Jeffersonville, IN not in the basement as someone earlier describe them. The coding team in Jeffersonville also codes occupation for the decennial census and they do other things besides coding. There are 12-15 coders working fulltime. They assign descriptions based on the answers to the two questions to something similar to the Standard Occupational Classification system, which has about 500 categories that are defined by 3 digits. We have heard about these in the workshop – the first digit is the highest level and each subsequent digit refines the classification.

There are really 3 parts to our study. The first was kind of exploratory. We took advantage of the fact that the Census Bureau double codes. They have two coders classifying descriptions from about 10% of the respondents in the current population survey. We had a double-coded data set from about a year's worth of data consisting of about 32,000 judgments. We just wanted to look at a couple of things. One thing was the descriptions themselves if there were characteristics of the descriptions that might have affected the agreement between the coders, that is, pairs of coders.

Second, we decided that because we did not have control over the descriptions themselves we could not ask all the questions we wanted to so we designed an experiment in which we seeded the coding production stream – the responses the coders saw – respondents' open job descriptions that we borrowed from the actual descriptions provided by respondents in the prior study. So we could then look at how different characteristics of the descriptions might have interacted to affect reliability.

Third, we noticed that neither of these gave us really any insight into coder's thinking. When there was disagreement what were they thinking? Why would they reach different conclusions? We conducted a very small qualitative experiment or observational study in which we asked respondents/coders to think out loud as they classified about 100 descriptions. Looking at the first of these – this is the large data set that was double coded -- about 10% are double coded each month; at least they were in the late '90's. We have about a year's data from 3/97-2/98.
The first thing we observed was that there was about 15% disagreement between the coders, which struck us as high. We wondered whether these are these superficial or substantive differences. The vast majority (78% of the disagreements) were in the first two digits. This means that pairs of coders were placing the same respondents' job descriptions into different high level categories. (We'll look at some examples soon.) So this is not rounding error. They are not disagreeing about whether running with your dog is exercise, or pet care; this is about drivers vs. paramedics or even more different than that.

One thing to note that I think echoes a comment that has come up a few times in the workshop is that agreement or reliability itself does not guarantee accuracy. It is possible that two coders could agree and both could be wrong. But if there is disagreement it does mean that at least one of them is wrong. It is not a perfect measure of validity but it is useful.

We next looked to see if the disagreement might have been just due to typing error and there was no evidence of that. Simple transposition error, for example typing 234 vs. 243 accounted for less than 1% of the disagreements. There were some (about 7%)

discrepancies of more one digit like 123 vs. 223 but these may well have been intentional. These are fundamentally different categories.

So what are the other findings? One that came as a surprise to us is that the length of the description affected reliability in the opposite direction from what we would expect: It turned out that when descriptions were longer, coders disagreed with each other more than when they were shorter. They also referred to supervisors, which is kind of a non-response version of disagreement, a coder felt he or she could not classify the description. So, longer descriptions were harder for them to code. We expected the opposite – namely that more information should result in ambiguities and, therefore, disagreement.

The difference is small but quite significant. When coders agreed with each other, the description was about ½ word shorter, than when they disagreed. This is consistent with an earlier project that Mick and I have done when asking both of these occupation questions led to lower disagreement instead of just asking one of them. So, more information seems to lead to lower agreement.

What this might suggest from a practical point is that asking for additional information is not always a good thing. By collecting more information that is not the right information it may actually make matters worse. There is some suggestion that coders actually recognize this. In an old study by Jim Esposito and David Canter they asked CPS coders to read the descriptions and judge if interviewers should have probed for more information. The coders rarely asked for more information. The only circumstances in which they asked for more information were when they felt they could not make a decision based on what they had; if they could make a decision, whether or not it was necessarily the right decision, they were content with the descriptions that they had.

Besides length of terms in the description it turns out that some words that respondents use are just inherently more codeable or lead to higher reliability than others. Words that lead to high disagreement tend to be more abstract, like “administrative,” “services,” “research,” “assist,” “maintenance” and so on. Compare these to words that lead to less disagreement. They are more concrete like “waitress,” “registered,” “guard,” “electrician,” “secretary” or “accountant.” So concreteness seems to be relevant as well.

So we were wondering how these characteristics of open-ended descriptions might interact with each other. As I said before, we didn’t have sufficient control over the descriptions to explore this systematically so we conducted a follow-up experiment where we seeded the production stream with descriptions that we varied systematically on a number of dimensions. I will just talk about length of the description and concreteness (or the difficulty) of the actual words. Length – 1, 2 or 3+ words -- and the primary occupational term was either *easy* or *hard* based on these criteria. What we found was that the impact of length was really only observed when the descriptions contained easy words. Basically if the coders could reach a decision on the basis of these words because they were easy, more words got in the way. But when the words were hard, additional words didn’t make things worse.

Then, in the third part of the study, we looked closely at 4 coders. We had them think out loud as they coded about 100 descriptions that we chose to be difficult. From the previous experiment we picked descriptions that were known to cause disagreement –what would they do when they were confronted with cases that their colleagues had other opinions about? These descriptions were relatively long and there was high disagreement about their codes in the experiment.

The clearest result was that coders used what we call *special purpose rules* that were not documented anywhere. These concerned superficial characteristics of the descriptions as opposed to being more theoretically grounded. They didn’t use the definition of these jobs (which would have been more theoretical). I will show you some examples in a second. The

coders weren't really aware of the origins of these rules either; they were just in the culture. But they were all *familiar* with them.

Whether all of the coders used the rules systematically is another question. It could very well be that these rules are doubly bad because they distract the coders from the content of the codes but then the rules also weren't applied systematically. The point is, they seem to be widely used. This has been observed in other coding organizations. Rules of this kind are probably used because they increase reliability but they have nothing to do with accuracy or validity, or at least increasing these scores does not seem to be the motivation. Coders are rewarded for reliable judgments.

Another study that observed something very similar was by Jean Martin and her colleagues in the '90's. They compared what they called *office coding* to *field coding* (in which interviewers coded the responses they collected) and they have the usual coding staff code in the office. They found more correlated coder variance like Patrick was discussing yesterday among the office coders – the regular coders – as opposed to the interviewers. That could be due to the fact that the office coders are sitting in the same physical environment so they can develop these special purpose rules, but the error was correlated – different errors for different coders; this could well be the case because the office coders didn't apply the rules systematically. So there is a double whammy in using these special purpose rules.

There is another relevant study conducted by Hak and Burndt in the Netherlands. They applied a conversation-analytic approach in which they looked closely at the interactions between the coders and observed the creation of one of these special purpose rules. The study does not concern occupation coding or survey research. The coding category in this example is something called *value*. One person says that they think if the word "being able to" occurs in the answer "we can almost always choose VALUE" as a code "if it is possible to read the answer in that way." So the coders decided that that was how they were going to go about answering when there isn't a specific documented rule.

One of the rules that we observed was designed to deal with situations where the respondent had to describe two occupations instead of just one. In this case the respondent was employed by Domino's Pizza in the pizza industry. He or she described their duties as a "cook" and a "driver." And when asked the duties question the respondent reported that he or she delivers and does cooking. So the rule that the Jeffersonville coders used to deal with this situation was that when two different occupations are described like *cook* and *driver* – match the duties and code according to the first duty listed. So this person is a cook and a driver but gets coded as a driver.

In the next example, this person is employed at Newberry's which is a variety store. Their job is a cashier and they do stocking. So when asked for duties they said stocking and cashier. So this person would be classified as *stocking* as a way of resolving the fact that they listed two occupations. This could introduce coding error if it is wrong, that is, if the second duty isn't the person's primary job. It could be that they really have two jobs or it could be that the third job they listed was their primary job but they might have done listed things in a different order to avoid being redundant. They are basically being asked the same question twice – "What is your job?" and then "What do you do?" If their job describes their duties, to avoid saying exactly the same thing twice they may just switch the order around. That is common in survey response. So this particular rule may lead to error because the order in which the duties are reported has no particular bearing on the truth.

Another rule that the coders adopted concerned particular words: if a particular word occurs, assign it to this code. This was specific to *secretaries*. If the word "secretary" appears in the occupation, don't look at anything else; code it as *secretary* which is 313. That might lead to

the right answer sometimes but because the coders don't look at anything else it could very well lead to the wrong answer.

Another one of these rules – this one might actually be helpful some of the time – had to do with when there is a modifier, in particular when the word "assistant" appears. The rule is that if "assistant" is in front of the occupation word, ignore it. If "assistant" comes after the word, "assistant" is the key word. In this case this person was an assistant teacher in special ed. It leads to coding for a special ed teacher. They dropped assistant and classified him or her as a teacher in special ed. But in this case the person is a teacher's assistant so they get classified as a teacher's aide, their job is being an assistant to a teacher, not being a teacher at with a rank of "assistant." There may be a difference between the meaning of *assistant* when it is in front or behind the primary occupation word but there is an arbitrary feel to this.

I'm actually done presenting the study. I think this is a record for me, being done early.

So to sum up, characteristics of open responses in this study seem to affect data quality, in particular, inter-coder agreement. As we saw longer descriptions reduce agreement. Although this may depend on particular words as some words are inherently more difficult to code than others, at least in this domain.

Coders developed special purpose rules for classifying potentially ambiguous cases – like 2 jobs; clearly this may improve reliability if the rules are applied systematically but it could harm accuracy or validity because they have no particular bearing on the truth – they are not informed by theory or definitions of jobs. If they are not applied consistently, even reliability will be harmed.

For us the implication is more *accurate* coding to the extent that we can define that, not just more *reliable* coding should be a priority because, again, these rules may increase reliability but not validity. How can this be done? Clearer coding logic can be conveyed to the interviewers but all in all I think it will require better job descriptions, definitions, and nuances conveyed to interviewers through training, and *automated support* in the interview.

So the interviewers essentially transcribe answers– at least this is how how the CPS works – and as the interviewer types verbatim what the respondent says, they can be flagged by the systemn when there is a potential ambiguity. Coders can be trained to code terms, known from studies to be difficult, in a theoretically sound way as opposed to coding *secretary* as always 313.,if these special purpose rules can be brought out of the closet and can be documented and re-written on the basis of the category definitions and relevant theory. So if accuracy is a priority then reliability will follow because if multiple coders are getting it right then they will also be recording the right code.

Formatted: Font: (Default) Arial, 10 pt