CITATION

Creative Commons License: Attribution-NoDerivatives 4.0 International (CC BYND 4.0)
QUESTION

In your literature review of content analysis, you discuss the trade-offs between human coding and computer coding in terms of reliability, validity, and costs. Thinking specifically about data policies and management documents, what dimensions do you see that would be well-suited for computer analysis and what dimensions would require human coding? What processes would you use to ensure reliability and validity? How would you resolve disagreements between computer coding and human coding?

RESPONSE

There are five primary dimensions for one to consider when deciding whether or not to use computer coding or human coding generally with regards to an analysis of data policies and management documents. Most of these dimensions may be decided when setting up the research study in the first place.

1. What does one intend to measure? Is one examining latent or manifest content? Both?
2. Is this a qualitative Content Analysis? A Quantitative Content analysis? Both?
3. What is the unit of analysis? Is it the entire document or interview? Is it the sentence? The word?
4. What types of policies is one examining and analyzing? Standards? Best practices? Local policies (e.g., the policies of the archive itself, and its governing department and/or institution)? Policies implemented at the machine-level, in the code?
5. What is the goal of the analysis? Is it to create a typology of policies? To classify policies by domain? To classify policies by the issue(s) that drove their creation?

Each of these dimensions will drive the choice of human versus computer coding, or the decision that either one would be an appropriate choice.

Policy implementation tends to be both top-down and bottom-up; it is both explicit and implicit. It is a bi-directional process. An example of a top-down implementation would be an archive administrator of a government digital data archive, who provides
public access to the data, who must comply with federal or national and/or state/province laws; archival standards and best practices; international standards (such as the OAIS Reference Model); and the archive’s own policies, which are determined by the archive staff, and the department and institution, if the latter are applicable. Thus, any written policies would be considered explicit, and any “understood” policies would be considered implicit.

A competent technical administrator of the same archive should implement certain policies based on computer science best practices, even without the written policies. Those policies should include regular backups of the data to at least two off-site locations; the testing of the backups to make sure they are actually backing up the data; fixity/integrity checks of the data upon ingest and regularly thereafter to ensure the data has not been corrupted or changed; virus scans upon ingest; the use of unique identifiers; the migration of hardware and software; and, a method for accessing all stored files, among others. The technical administrator may implement these policies because of written, explicit policies, or simply because it is implicitly understood that this is what is done.

The policies themselves may be thus classified as:

1. Standards (“explicit”). For data policy and management documents, these include international standards such as the “OAIS Reference Model” and the “Audit & Certification of Trusted Digital Repositories”. One can assume that the wording within these documents has been rigorously examined to avoid ambiguity, although some ambiguity may exist. Standards are thus explicit, written policies.

2. Best Practices (“explicit or implicit”). For data policy and management documents, these would encompass two types of Best Practices. The first
are the best practice policies of the domain that created the data, such as Social Science, Hydrology, or Physics, for example. Some domains have no explicit written data policies; it is a kind of Wild West from an Information and Library Science (ILS) perspective. Other domains, such as Social Science, have very explicit policies.

The second are the Best Practices of the domain that is stewarding the data. In some cases, the domain that created the data is also the domain that is stewarding the data for the indefinite long term. In other cases, the domain that is stewarding the data is a separate domain, such as ILS. If the domain stewarding the data is ILS, then policies tend to be explicit.

Best Practices are thus either implicit (not written) or explicit (written) and usually vary by domain. If written, the wording may or may not be rigorous in order to avoid ambiguity.

3. Implicit. These policies are policies implemented knowingly or unknowingly by the managerial and technical administrators of an archive. An example of this is the regular migration of the Social Science data and supporting software and hardware by Odum Institute employees from the 1960s to the late 2000s. The Odum Institute Data Archive currently has a stated preservation policy, but from the early 1960s to the late 2000s, the archive did not. The Data Archive administrators had an implicit goal of preserving the data and ensuring access to it over the long-term, but they did not know they had a preservation policy. The preservation policy was not explicitly stated in any documentation until sometime around 2008.

Thus, implicit policies are, well, “implicit”. The wording may or may not be unambiguous.

4. Machine code (“explicit”, but based on both “explicit” and “implicit” policies). These maybe scripts or programs that implement human-created policies at the machine-level. As previously stated, these policies may be implicit or explicit, and may conform to national, local or other standards and Best Practices.

Machine code policies are explicit in that they are implemented in code; the source of the policy itself may be written (explicit) or unwritten (implicit). The policy implemented by the code should be unambiguous, but in some instances may not be.

5. Local policies (“explicit or implicit”). Local policies are the policies of the archive itself, its department and/or institution, and any relevant federal, state or county rules and regulations (within the USA). Local policies may or may not contradict Best Practices and International Standards. For example, archivists in the UK had to forgo certain requirements of the OAIS Reference Model in order to comply with UK guidelines for providing access to handicapped persons.
Thus, it may be more appropriate to use computer coding for one kind of policy, and human coding for another. I have outlined the following dimensions, based on the above. The following assumes that the corpus to be analyzed is small enough to be analyzed by one or two human coders. If the corpus is too large, either more humans need to work on it, or computer coding must be utilized.

<table>
<thead>
<tr>
<th></th>
<th>Computer Coding</th>
<th>Human Coding</th>
<th>Manifest ((M)), Latent ((L)), or Both ((B))?</th>
<th>Reason/Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standards</strong></td>
<td>X</td>
<td></td>
<td>B</td>
<td>Wording should be unambiguous enough to be computer coded.</td>
</tr>
<tr>
<td><strong>Best Practices</strong></td>
<td>X</td>
<td>X</td>
<td>B</td>
<td>Case-by-case; wording may or may not be unambiguous. If ambiguous, human-code. If not, computer code.</td>
</tr>
<tr>
<td><strong>Implicit</strong></td>
<td></td>
<td>X</td>
<td>B</td>
<td>Wording is likely to be ambiguous and must be gathered via an interview or survey.</td>
</tr>
<tr>
<td><strong>Machine Code</strong></td>
<td>X</td>
<td></td>
<td>M</td>
<td>Wording is likely to be unambiguous.</td>
</tr>
<tr>
<td><strong>Local Policies</strong></td>
<td>X</td>
<td>X</td>
<td>B</td>
<td>Case-by-case; wording may or may not be unambiguous. If ambiguous, human-code. If not, computer code.</td>
</tr>
</tbody>
</table>
All of the above would require cross-checking and cross-validation. For example, if one wants to analyze domain Best Practices that are written, a small subset coded between two humans may determine whether or not the language is ambiguous. If the wording is reasonably unambiguous, then once agreement is reached between the two humans, then a cross-check between the human coding and the computer software may determine whether or not the Best Practices may be better analyzed by a human or a computer.

Another factor in the decision to use human or computer coding – not mentioned in the chart above – is the unit of analysis. If one simply wished to perform a purely quantitative Content Analysis consisting of a straight word count, then computer coding is the best choice, regardless of the quality of the source material. If the unit of analysis is the sentence, the paragraph, or the document itself, then the implication is that the analysis also includes an interest in latent material. In that case, a computer or a human may analyze the policy documents, based on the matrix above.

If the goal of the analysis is simply to analyze policies in order to classify them by type of policy (“security”, for example) and domain (“Physics archives use these types of policies”), a straight up word count may be enough data to provide the answer to that question. In that instance, a computer coded analysis should be sufficient. If one is also interested in what issues the implemented
policies represent, for example, then a qualitative analysis is a better choice. This is likely to involve both a quantitative analysis of word-count, and human coding of the chosen corpus that then uses computer software to analyze the coded data.

In order to ensure reliability and validity, one should explicitly state which existing policy documents (including computer code) were used, from whom they were obtained, when they were obtained, what version was examined, and any observed anomalies. If the policies are generated via an interview or survey, that must also be stated. In the event that human coding is used, then at least two human coders should code the same material, and then the results must be compared. The two or more coders must come to agreement on their results to ensure reliability.

In order to ensure validity, then one must be sure to design the study so that it measures what it is supposed to measure. If one is interested in determining which issues drive the creation of policies, then one should not design the study to measure the number of times “the” is used in a particular policy document. The study must gather data that is relevant to the question, not spurious. Again, inter-coder reliability should help ameliorate any unintentional study design that would affect validity. With regards to Content Analysis, however it is generally better to go for high validity over high reliability.
Another way to ensure validity is to use human coding over computer coding. For example, since computers cannot always determining meaning, if one is examining latent content, if the corpus is small enough, it may be better to use human coders. A computer analysis will certainly plow through a far larger corpus than a human or group of humans will be able to, but if the coding is not set up correctly, or the computer cannot determine meaning, then the results may be spurious. Whereas, one or more humans analyzing the same corpus may come up with valid and important results, because a human is generally better at assessing the implied meanings within the text, especially in terms of what is not stated.

If and when human coding does not match well with computer coding, there are several avenues to address the problem. The first is to compare the coding instructions for the software versus the code book for the humans, in order to determine whether or not there are any discrepancies. Another is to determine if this software has any particular bugs. One could make some small adjustments to the configuration of the software to test if and how those changes affect the results. If the human training and the software configuration are in sync, and the software does not contain any bugs, then one can try installing the software on another machine as a completely fresh install, re-use the configuration, and run the analysis again. It could be that the second instance of the software may provide different results, as the first instance my have some unknown bugs. One can also check to ensure that the human coders and the
machine are actually examining the same corpus. A human may “eyeball” the corpus the computer just examined to determine whether or not the computer is even ingesting the material correctly. If the results still don’t pair up, then the researcher will have to make a judgment call as to which results to use.

In conclusion, there are multiple dimensions to consider when conducting a Content Analysis using data policy and management documents. As well, reliability and validity must be considered, as must inter-coder reliability, and human-computer inter-coding reliability. There are trade-offs for all of these.