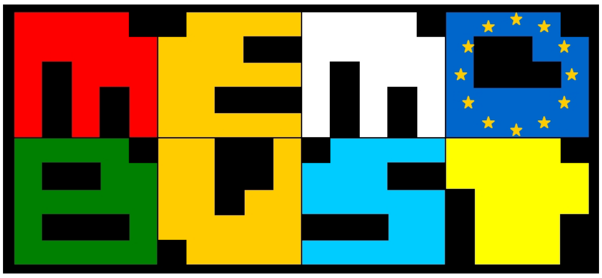

This module is part of the

Memobust Handbook

on Methodology of Modern Business Statistics

26 March 2014

Method: Automatic Coding Based on Semantic Networks

Contents

General section

1. Summary

For a number of variables in questionnaires, one wants the answer in closed form, e.g., "city"; this is a relatively simple classifying task. Sometimes this task is much harder, e.g., when trying to get a code for occupation. One approach is to ask an open question ("what is your occupation") and then try and code this text at the statistical office. For the sake of efficiency, that coding process will start by an automatic step.

In some cases, no previously coded material is available in electronic form. The starting point then consists of the data to be coded and a classification with a textual description per code. In this situation, we can either build the informative base transforming the classification manual so as to be 'processable' by a computerised system and ensuring pre-coded descriptions or one must try and code open text answers based on the texts themselves and the associated semantics, to enable the approach from the module "Coding – Automatic Coding Based on Pre-coded Datasets".

Although an informative base can be constructed based on expert knowledge, pre-coded answers may also be added to the informative base to enhance the coding rate. This makes the distinction with the module "Coding – Automatic Coding Based on Pre-coded Datasets" less strict. The main distinction between the latter module and this one is the amount of manual work to construct an informative base: the methods in the other module are based on machine-learning requiring much less manual work. As described in the module "Coding – How to Build the Informative Base", the informative base can contain:

- the classification manual descriptions, transformed so as to be 'processable' by a computerised system;
- pre-coded descriptions collected in previous surveys;
- different kinds of synonymous, hypernyms and hyponyms.

There are general systems (ACTR, now G-CODE (Wenzowski, 1988) and Cascot (Cascot)) that use the elements above to code text in a number of steps, like pre-processing the text, replacing words and finally assign a code. Alternatively, most of these steps can be combined into a so-called semantic network (Hacking and Janssen-Jansen, 2009). In the following section we will describe the "spreading activation" search method in the semantic network in more detail as an example; at certain points we will describe the link with the "processing approach" in the ACTR tool.

2. General description of the method

Coding methods based on semantic networks (in its simplest form a search table) have in common that they are based on a number of relations between words or combinations of words; there is also a relation between combinations of words and classification codes. The most common relations are hypernyms, e.g., an apple is a kind of fruit; this kind of relationship allows the coding system to reduce the variation of words before performing the final coding step. Other relationships are synonym and hyponym (described in the next subsection). To code words are linked to classification codes (e.g., "carpenter" & "building site" → code 12345, "carpenter" & "factory" → code 12344, …) or there is a more advanced algorithm (e.g., Cascot[1]) that derives codes from a pre-processed description.

Here we describe an algorithm to use all of the semantic information in a single semantic network called 'Spreading activation'. This method is described in more detail below.

2.1 Spreading activation

For the coding of SBI codes (the Dutch version of the NACE codes), a technique called 'spreading activation' is used, where coding is performed based on a semantic network (which may have been created manually). This is a directed graph, also called a digraph, where the nodes represent words, and where the edges or directed edges (or arcs) indicate relationships between words (the exact relationship is stated by listing that next to an arc). For example:

- greenhouse vegetables $\xrightarrow{\textit{hypernym}}$ tomato: greenhouse vegetables include tomatoes.
- tomato $\xrightarrow{\textit{hyponym}}$ greenhouse vegetables: tomatoes are a kind of greenhouse vegetables.
- Agatha $\xrightarrow{\textit{synonym}}$ potato: for the classification, the potato varieties like 'Agatha', 'Anya', 'Fingerling','Jersey Royal', 'Kerr's pink', etc. are not important, and if they do occur in a description they can be considered synonyms to 'potato' which can be used instead.
- sale_of_childrens_clothing $\xrightarrow{\textit{Code}}$ 12345, because the description 'sale of children's clothing' unambiguously leads to the code '12345'.

These relationships[2] form a semantic network, of which a small part is shown for illustrative purposes in Figure 1.

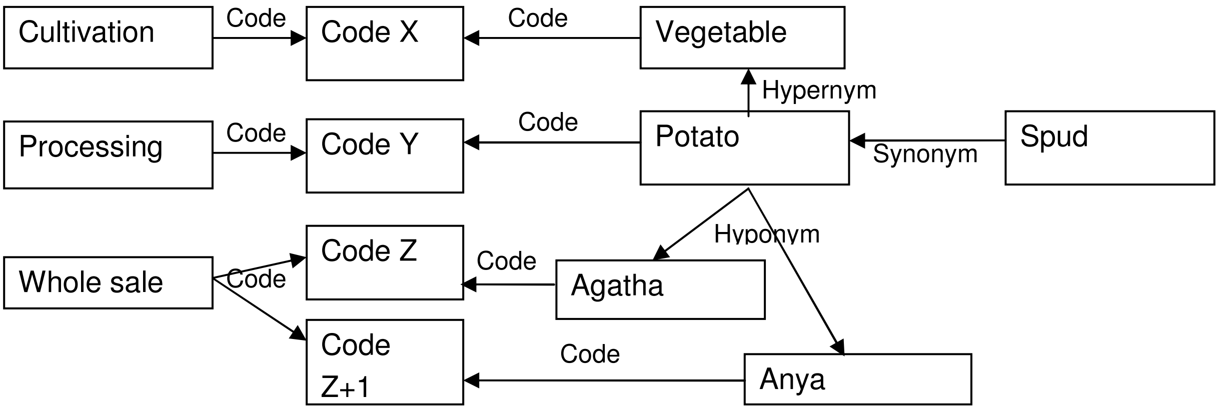

Figure 1. A fragment from a semantic network, as used in the coding of the SBI (Agatha and Anya are two varieties of potato).

---

[1] Unfortunately the actual coding/scoring algorithm is not described.

[2] Synonym, hyponym and hypernyms more or less correspond to the ACTR preprocessing step of replacing words. The ACTR step *remove words* simply corresponds to non-existing nodes in the semantic network. The code relationship corresponds to assigning certain codes a score, given the input description (the *weighting step* in ACTR).

Consider the description ‘cultivation of spud’ will give code X a score 2, whereas the other codes will get a score of 1 at the most: the word ‘spud’ leads to code X[3], through ‘potato’ and ‘vegetable’); the word ‘cultivation’ directly leads to this code. Hence, using hypernyms allows the description for code X (‘cultivation of vegetables’) to be found, even though ‘spud’ is too specific.

2.2 Algorithm

Text In this network, we see interrelated words, due to certain semantic relationships. The words in a semantic network are also called nodes, for which an associated tag is ‘activation’; this serves to quantify the extent to which a word correlates with the terms from the search string. The binary or other relationships that exist between the nodes can (formally) be recorded in an adjacency matrix.[4]

In brief, the algorithm amounts to the following:

1. Let the set of nodes be denoted as $\{n_1,...n_m\}$. Call the activation values during iteration round k of the nodes $A_k = (a_{k1},...,a_{km})$. Call the adjacency matrix $P = (p_{ij})$. This means:
    - $p_{ij} = 1$ if there is a link between two nodes n_i and n_j,
    - $p_{ij} = 0$ if that is not the case.
2. Next: for each word stated in the description:
    a. Set the activity of the node linked with word l from the description to 1: $a_{1l} := 1$.
    b. After this, all nodes that can be reached by an arrow from ‘activated’ nodes are also activated by means of the following relationship:

    $$a_{k+1,i} = \sum_{i,j} a_{k,j} \cdot p_{i,j}$$

    This must only be done for nodes not yet visited. In addition, there is a special restriction for the *hypernym* and *hyponym* relationships (the *parents* and *children*): if a path has already run along a *hypernym relationship*, then it may not run along any other *hyponym* relationships, and vice versa.[5]
3. The ‘expansion’ of the activity stops because all paths ultimately ‘collide’ on a code node, or because there are no more unvisited nodes near a node. The codes then contain an activity as described in 2a and 2b. All codes with an activity > 0, in order of activity, form the result of the search operation. In order to increase its effectiveness, one can only select those codes that have the same score as the top scoring node, e.g., if there is only 1 node with score 2 and 5 other having a score 1, the search method results in exactly 1 code, which is the desired for an automatic coding method.

---

[3] Actually, in this example, it leads to all codes.

[4] The actual implementation is likely to be different from the description given here, as this would be very inefficient. It is only for the sake of the explanation of the algorithm that an adjacency matrix is used.

[5] If this restriction were not present, then all the nodes in the classification tree would be visited, and this is not intended. We only want parents, grandparents, etc., and the ‘subtree’ of a classification node to be visited.

For more details, see Hacking and Janssen-Jansen (2009). For another application, see Berger et al. (2004). For a discussion of the use of semantic networks in coding, see Willenborg (2012).

3. Preparatory phase

In order to start coding with the approach as described in the previous section, one needs a so-called “informative base” (see “Coding – How to Build the Informative Base”). Such a base must be constructed “by hand” by experts of the classification. It is a process of trail-and-error: changes to the network may enhance the accuracy of some codes and, at the same time, decrease the accuracy of others. In order to keep the overall accuracy sufficiently large, one must use a test set to detect the changes in coding after changes in the informative base:

- descriptions that get coded correctly due to alterations or additions;
- descriptions that are no longer coded correctly.

These changes (especially the latter one) serve as a good feedback.

On the internet a number of general semantic networks can be found, such as “WordNet” or “OpenCyc”. These networks contain many concepts and relationships as described earlier. When using such network as a basis much work still remains as most classifications are rather domain-specific compared to these general networks.

4. Examples – not tool specific

5. Examples – tool specific

The Spreading Activation method has been applied to the classification of the SBI code. The provisional results are as follows: 80% correctly coded codes, and in 15% of the cases, multiple codes. For more details, see Hacking and Janssen-Jansen (2009). Some practical numbers: the total number of nodes was approximately 5200, the number of relationships was approximately 17200, and the search time in the implementation was around 0.23 sec on a 1.5 GHz machine.

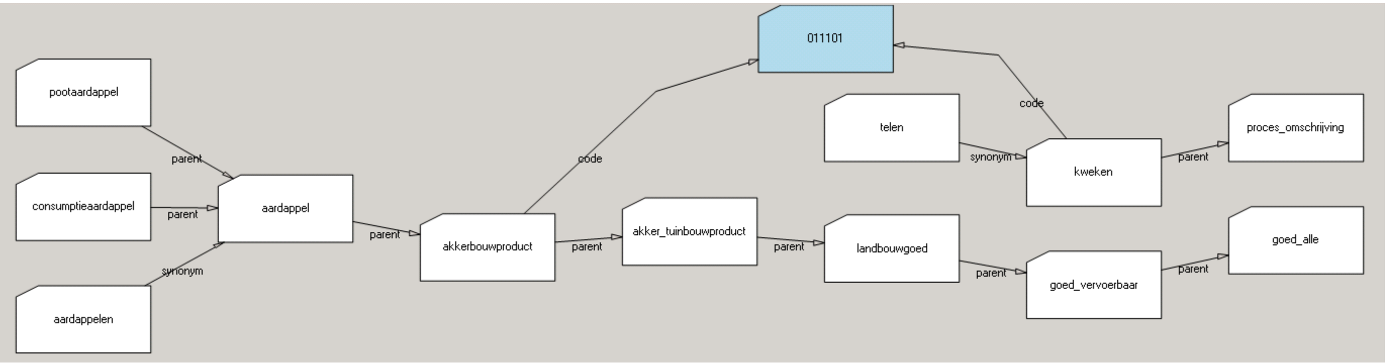

Figure 2. A screenshot that shows a small part of the semantic network (in Dutch) that was visited after the search string ‘telen van aardappelen’(‘cultivation of potatoes’) was provided to the spreading activation algorithm.

In Figure 2, a screenshot is shown of the proof-of-concept that was used for the coding of SBI. This shows a part of the semantic network that is ‘visited’ after providing the search string ‘telen van aardappelen’ (‘cultivation of potatoes’) Note that ‘aardappelen’ (‘potatoes’) (via the classification) leads to ‘akkerbouwproduct’(‘agricultural product’); combined with ‘telen’ (‘cultivation’) this leads to code 011101 having a score of 2; all other codes (not shown) that were visited received a score of 1.

The software described here implementing spreading activation (currently used for the coding of economic activity) can be used for other kinds of classifications, by creating a different semantic network files. Also, by translating the terms in the semantic network files into another language, one could have a system for other national statistical institutes[6]. The spreading activation program is currently being upgraded to a web version and we intend to offer a web interface or web page to various parties that need to code economic activity, especially the chambers of commerce.

For the ACTR and the Cascot tool, see Wenzowski (1988) and Cascot, respectively.

6. Glossary

For definitions of terms used in this module, please refer to the separate “Glossary” provided as part of the handbook.

7. References

Berger, H., Dittenbach, M., and Merkl, D. (2004), An accommodation recommender system based on associative networks. In: Frew, A. J. (ed.), *Proceedings of the 11th International Conference on Information Technologies in Tourism (ENTER 2004), Cairo, Egypt, January 26-28, 2004*, Springer-Verlag, 216–227.

Cascot (a program to semi-automatically classify descriptions): www2.warwick.ac.uk/fac/soc/ier/software/cascot/.

D’Orazio, M. and Macchia, S. (2002), A system to monitor the quality of automated coding of textual answers to open questions. *RESEARCH IN OFFICIAL STATISTICS (ROS)*, N.2 2002.

Hacking, W. J. G. and Janssen-Jansen, S. (2009), The coding of economic activity based on spreading activation. Report, Statistics Netherlands, Heerlen.

Hacking, W. and Willenborg, L. (2012), *Coding – interpreting short descriptions using a classification.* Contribution to the CBS Methods Series, Statistics Netherlands, The Hague and Heerlen.

Willenborg, L. C. R. J. (2012), Semantic networks for automatic coding. Report, Statistics Netherlands, The Hague.

Wenzowski, M. J. (1988), ACTR – A Generalised Automated Coding System. *Survey Methodology* **14**, 299–308.

---

[6] There is a complication however: national versions of the NACE classification are allowed to add a 5th digit to some NACE codes; this country-specific part needs to be redone.

Specific section

8. Purpose of the method

Automatic coding takes place just after the data have been collected, in most cases data from interviews. These interviews contain a few fields that are input for the coding step, e.g., “production of wooden crates” serves as input for the coding of economic activity. For simple classifications, e.g., “nation of birth”, a closed question can suffice in the interview; for more complex classifications such as education or occupation, a closed question will probably lead to long lists of possible code descriptions and the quality of the response will decrease rapidly. For that reason, it is often more logical to use an open question in the interview and code this answer at the statistical office; also, for reasons of efficiency, it is better to start coding automatically followed by a manual or a computer-assisted coding step (texts not coded automatically can be analysed by expert coders manually or with the computer support).

The method described here can be used to do the automatic coding step.

9. Recommended use of the method

1. Recommendations on the use of the different methods for coding (automatic or assisted) have been given in the module “Coding – Different Coding Strategies”: the decision about which is the most suitable coding approach to be adopted in a survey depends on different correlated factors. If it has been decided to use automatic coding, one needs a training set of coded descriptions that is available in electronic form, and a correct code (after verification) that is assigned to each description.

10. Possible disadvantages of the method

1. The initial construction of the information base requires a lot of work.

11. Variants of the method

1.

12. Input data

1. During the coding phase, the input is quite simple: a textual description, in most cases no more than 10 words. During the construction of the “coding machine” the input consists of the expertise from the classification experts.

13. Logical preconditions

1. Missing values
    1.
2. Erroneous values
    1.
3. Other quality related preconditions

    1.

4. Other types of preconditions

    1. The input text to be coded should not be too large; in general, this will result in many possible codes for this input text by the method.

14. Tuning parameters

1. In general, all practical automatic coding algorithms need a *score cut-off value*, to make a selection which descriptions are coded and which need manual or assisted coding. In our experience this parameter is rather robust.

15. Recommended use of the individual variants of the method

1.

16. Output data

1. Per description the following is derived by the method:

    - a score;
    - a classification code.

17. Properties of the output data

1.

18. Unit of input data suitable for the method

Incremental processing

19. User interaction - not tool specific

1. None

20. Logging indicators

1. A number of things may be logged during coding operations: for each coded text all (intermediate) results can be stored for further analysis. This logging can be used when analysing the coding results for a given test set; for each text that was coded correctly before and incorrectly coded now, one can look at logging associated with that text.

21. Quality indicators of the output data

1. The indicators described in the module “Coding – Measuring Coding Quality” (coding rate and precision rate) can be used to quantify the quality of the method. Quality testing is a continuous process when using semantic networks. During the development of the network, each alteration (or addition or removal) meant to enhance the classification towards code A may deteriorate the classification towards code B. For that reason one needs to check and record the codes assigned by the network based on an incorrectly coded test set. By comparing the assigned codes before and after changes, one can assess the good the change was.

22. Actual use of the method

1. This semantic network method has been used since 2006 until now at the Dutch Chambers of Commerce (in collaboration with Statistics Netherlands) for the coding of economic activity. The ACTR tool has been used by Statistics Canada (who made it; Wenzowski, 1988) and IStat (D'Orazio and Macchia, 2002). Cascot is used by ONS; Statistics Netherlands currently uses it for the coding of occupation.

Interconnections with other modules

23. Themes that refer explicitly to this module

1. Coding – Main Module
2. Coding – How to Build the Informative Base
3. Coding – Different Coding Strategies
4. Coding – Measuring Coding Quality

24. Related methods described in other modules

1. Coding – Manual Coding
2. Coding – Automatic Coding Based on Pre-coded Datasets
3. Coding – Computer-Assisted Coding

25. Mathematical techniques used by the method described in this module

1.

26. GSBPM phases where the method described in this module is used

1. 5.2 Classify and code

27. Tools that implement the method described in this module

1. ACTR (Wenzowski, 1988)
2. Cascot (Cascot)

28. Process step performed by the method

Coding

Administrative section

29. Module code

Coding-M-Automatic Coding Based on Semantic Networks

30. Version history

Version	Date	Description of changes	Author	Institute
0.1	02-04-2013	first version	Wim Hacking	CBS
0.2	20-01-2014	following review by Stefania Macchia	Wim Hacking	CBS
0.3	30-01-2014	following review by EB	Wim Hacking	CBS
0.3.1	30-01-2014	preliminary release		
1.0	26-03-2014	final version within the Memobust project		

31. Template version and print date

Template version used	1.0 p 4 d.d. 22-11-2012
Print date	21-3-2014 18:06