Title:

Procode: a machine-learning tool to support (re-)coding of free-texts of occupations and industries

Nenad Savic[1*], Nicolas Bovio[1], Fabian Gilbert[2], José Paz[1] and Irina Guseva Canu[1]

***The corresponding author:**

Nenad Savic
Centre for primary care and public health – Unisanté
Route de la Corniche 2
1066 Epalinges – Lausanne
nenad.savic@unisante.ch

[1] Centre for Primary Care and Public Health (Unisanté), University of Lausanne, Route de la Corniche 2, CH-1066 Epalinges-Lausanne, Switzerland

[2] Research Institute for Environmental and Occupational Health, 28 rue Roger Amsler, CS 74521, 49045 Angers, France

Keywords: epidemiology, occupational classifications, machine learning, Naïve Bayes, cross-validation

Number of words: 2500 (excluding title page, abstract, acknowledgment, funding and references)

Abstract

This paper describes the algorithm, performance evaluation and future goals regarding the development for Procode. This tool incorporates Complement Naïve Bayes as a machine-learning technique to support automatic coding of free-texts against classifications established for occupations and industries. About 30’000 free-text entries with manually assigned classification codes of French classification of occupations (PCS) and French classification of activities (NAF) were used for Procode training. Using 5-fold cross-validation, the study found 57-81% and 63-83% predictions agreed with the manually assigned codes for PCS and NAF, respectively, depending on the hierarchy level of the classification codes. Procode also supports recoding between different classifications. For its first release, however, focus was mainly on the free-text coding. Regarding both operations, the availability of more data in different languages coded using different classifications is curtail for further Procode development and performance testing.

Introduction

Large occupational exposure databases and job exposure matrices (JEM) exist and could be of a meaningful value in studies that associate different exposures (e.g. chemical) to specific occupations, industries or populations worldwide (Fadel et al., 2020). Their use, however, is limited due to prerequisite of translating the data into a usable format. JEMs include free-text entries, which must be in alignment with the job or industry titles defined in different national or international classifications. This task, however, is time-consuming, expensive and requires adequate skills (Koeman et al., 2013; Peters, 2020).

Several tools have been developed to provide support to the users coding the free-texts (De Matteis et al., 2017; Patel et al., 2012; Remen et al., 2018; Russ et al., 2016; Warwick Institute for Employment Research, 2018). Some of these tools assign classification code to the job entries automatically, while the others are rather (manual) search assistants. For example, OSCAR (De Matteis et al., 2017) is a web-tool for coding the workers' job history, which provides a tree structure of The British Standard Occupational Classification (UK SOC) to help the users in selecting the appropriate job titles. Another tool, CAPS-Canada (Remen et al., 2018), assigns automatically the most likely classification code and title for a free-text entry designating a job. This tool supports seven classifications, i.e. four for occupation and three of industries. For each job title defined in a classification, CAPS-Canada calculates a score depending if it finds the word(s) of a free-text in the job title, its definition or synonyms. SOCcer (Russ et al., 2016) and CASCOT (Warwick Institute for Employment Research, 2018) apply more advanced approaches. SOCcer, for example, combines multiple classifiers based on training datasets to derive the most probable outcome for the entered free-text criteria. CASCOT was trained to automatically assign job titles of UK SOC and ISCO-08 (International Standard Classification of Occupations) and industry of SIC (Standard Industrial Classification). CASCOT also supports a variety of languages (e.g. English, French, Finish, Slovak, etc.).

The use of the tools that are not based on a training data, such as, for example, CAPS-Canada, is limited to the free-texts containing terms existing in job titles or additional job information (e.g. their descriptions). For the tools trained on dataset features (e.g. SOCcer), their predictions are strongly affected by the given data collected upon to the tool's development. By keeping the training data constant, a prediction bias that occurs once for one user would thus repeat for the others whenever similar entries are coded. Finally, the most comprehensive tool, CASCOT, is not a freeware, except its online version, which lacks the features of the standalone version. The online version of CASCOT, for example supports only English and cannot be used to code files containing multiple free-text entries.

We aimed at designing a new approach, expected to overcome the limitations identified for the existing tools. Also, the special efforts were done to mitigate issue of small or unavailable training data for certain classifications. This was embodied in a tool named Procode (available at URL: http://www.pro-code.ch) that is offered to the users free of charge.

Methodology

Procode is intended to support both coding (free-text assignment to job title) and recoding (job title translation from one to another classification). For the first release, the focus was on the former, while the latter operation, though supported, is simplified. This means that recoding is executed only by following the "translation" rules defined in a corresponding crosswalk, if existing.

Coding algorithm

Figure 1 illustrates the workflow of the coding algorithm integrated in Procode. A training dataset must include free-text entries with corresponding manually assigned job codes. The first task is data preparation, which includes a formatting operation known as lemmatization and word filtering by importance. Lemmatization (Bird et al., 2009) is a linguistic operation that is used to group together different forms of a word (i.e. lemma) in order to facilitate their analysis. For example, "walk" and "walking" have the same lemma, i.e. "walk". In this study, the goal is to reduce the number of these predictors or, in other words, to increase the frequency of the given lemmas. If an entered free-text is misspelled (e.g. "enginner" instead of "engineer"), the algorithm fails to find lemma and a warning message is displayed in Procode. Not all lemmas are processed. For example, those of no importance, such as "and", "although", "or", "anyone", etc, which are known as stop words, are discarded. For others, their importance was evaluated using the method called *term-frequency – inverse-document-frequency* (id-idf) (Nguyen, 2014). For example, word "assistant" likely appears in many free-texts, while it can be associated with a variety of job titles. This word is a bad predictor and was thus discarded. The formatted data is then used to train a machine-learning algorithm, which is used to predict job code/title for a new free-text entry. The lemmas in free-texts are used as independent predictors (x) of assigned job codes–depending variable (y).

Procode is designed to support four languages, i.e. English, German, French and Italian. If no data exists to train the algorithm in the language of the free-text entries, the contained words are translated (e.g. English "restaurant manager" to French "gérant de restaurant"). This is done automatically in the background without consulting the end-user.

It is unlikely to expect that a training dataset may cover the whole universe of different possibilities. Therefore, Procode may fail to deliver a job code/title for an "exotic" free-text entry. In these situations, the algorithm would create a longer string including synonyms and the definitions of those words in the given free-text entry. After formatting this new entry, the coding is repeated. It is assumed that the new text would include words that appear in the training dataset.

Ideally, for each classification, a corresponding training dataset (containing manually coded free-texts) should be supplied. In case of no training data, job titles/codes of the given classification are obtained through recoding from another classification, for which the training data exist. This means that the coding is initially performed for classification B and then, using a corresponding crosswalk, recoded for classification A. In this study, we used this approach for an end-user test explained later in the text.

Finally, the predictions are displayed to the user. In case of multiple job codes assigned to a given entry, they are ordered by the probability that they match the coded criteria. Their probabilities outputted by the algorithm are used to calculate the corresponding scores depicting the prediction uncertainties. The users of Procode are then provided with the possibility to judge the displayed predictions. Biased or incorrect predictions can be reported together with providing "the best match" jobs manually. This information is then added to the training data and used to improve future prediction. The tool thus constantly learns from its mistakes.

Machine-learning classifiers

Currently, Procode integrates Complement Naïve Bayes (CNB) (Bird et al., 2009; Ikonomakis et al., 2005) as a machine-learning classifier. CNB performance was compared with that of Support Vector Classifier (SVC) (Ikonomakis et al., 2005) and Random Forest Classifier (RFC) (Cutler et al., 2011) and exhibited the best results.

The authors considered these three methods as they fit the text classification intention of this project (Bird et al., 2009; Ikonomakis et al., 2005; Korde, 2012). They are supervised learning models, which use the lemmas defined above as predictors of job titles/classes. They are suitable imbalanced datasets (Rennie et al., 2003)–training dataset contains unequal distribution of records per outcome (e.g. job code). CNB applies the Bayesian inference to calculate likelihoods that different words of a free-text determine occupations or industries of a classification. While SVC tries to separate the defined universe of words with a dimensional hyperplane, RFC establishes a set of random decision trees that contribute differently to the final prediction.

Technologies applied

Procode is designed to be a web application. Its back- and front-end sides are decoupled, where the former was developed using Django (Foundation, 2020)–Python web framework, while the later using ReactJS (Facebook, 2020)–JavaScript library. Natural Language Took Kit (NLTK) (Bird et al., 2009) was used for text formatting (e.g. lemmatization), while the language translations were enabled by using "translate" package of Python (Yin and Henter, 2020). Finally, the text-classifiers (e.g. CNB) are part of "sklearn" package of Python (Pedregosa et al., 2011).

Evaluation

Cross-validation

A 5-fold cross-validation used 30’000 free-text entries from CONSTANCES cohort (Goldberg et al., 2017; Zins et al., 2015). The data was in French language and included manually assigned occupational job codes of PCS-2003 (fr. *Professions et catégories socioprofessionnelles*) and industry codes of NAF (fr. *Nomenclature d’activités française*). Because the data included job/industry codes of different precision level (i.e. from major groups to subgroups holding different level of detail; usually different number of code digits), the evaluation was performed for each of them separately. The prediction results were compared with the previously manually assigned codes and the percentage agreement was calculated.

End-user tests

The authors invited two external testers to test the performance of Procode using their own datasets containing free-texts. One tester coded 10’000 free-texts against PCS and NAF, while the other coded 945 records against ISCO-1988. The two datasets were in French and included previously manually assigned PCS and NAF for the former and ISCO codes for the latter. It is important to note that the data was not part of the previous cross-validation. The testers calculated percentage accuracy between the predictions obtained using Procode and the manually assigned codes.

At the time of writing, Procode only contained a training dataset for PCS and NAF. For ISCO-1988, this means that Procode coded the free-texts initially against PCS and then recoded the results to ISCO-1988.

Language translation test

To test the agreement between coding of free-texts given in different languages (see Figure 1), the authors generated two lists, i.e. in English and in French, each with 200 job free-texts. The data in one corresponded to that in the second list. Since the first release of Procode is based only on the data in French, the entries in other languages must thus be translated to French prior to the coding operation. Both lists were coded and the agreement between the predicted outputs calculated.

Results

Table 1 summarizes the agreement (in %) between the predicted and manually coded data obtained for the cross-validation and the two external tests. Regarding the cross-validation, 57-81% of PCS predictions and 63-83% of those for NAF agreed with manually coded jobs. As expected, the best agreement was when predicting the major groups (i.e. top-level codes or those with one digit in their codes) of these two classifications. The external tester reported results (given in brackets in Table 1) that were very similar to those of the cross-validation test for PCS and NAF. For ISCO-1988, as no corresponding training dataset was available, Table 1 includes only the results from the external test. For this classification, 58-70% predictions agreed with the manually assigned codes. As the crosswalk between PCS and ISCO-1988 did not support translations to the final code level in ISCO (i.e. four-digit codes), the corresponding results are missing.

The obtained results for the other two considered classifiers, i.e. support vector and random forest, are given in supplementary material. Somewhat lower values (1-2%) were observed for the former (Table S1). The later (Table S2), however, showed much lower performance, where, in the best case, it accurately predicted 31% classification codes.

Finally, for 200 job free-texts in English and French, the coded PCS predictions agreed 95%. In other words, Procode assigned different codes/titles only for ten entries out of 200 (5%), which designated the same occupations, but in the two different languages.

Discussion

The “gold standard” is the manual coding and manually coded data, which can be used to train different machine-learning classifiers. Such data are usually unavailable due to confidentiality issues or simply lack of willingness of different institutes to share their data voluntarily. The available data, however, are usually limited in size or variability of free-text records. Moreover, the entries are often given in one language (e.g. English). To overcome to some extent the mentioned issues, Procode was designed with a special focus on the lack of good quality data.

At the moment of writing, only PCS and NAF training datasets were available to train CNB. Coding against another classification is thus possible only if a corresponding crosswalk exists. This was done in our external test, where a tester coded his free-text entries against ISCO-1988. For the moment, however, it is not possible to code against a classification that has no crosswalk defined with PCS or NAF. Besides the use of recoding to support the coding operation, Procode also allows the users to recode their own job titles/codes from classification A to classification B. For example, it is possible to recode occupations from ISCO-1988 to NSP (Swiss national occupational classification).

Procode is freeware with a modern interface (see Supplementary material, Figures S1-S3) that operates in multiple languages and allow the users to code up to 10’000 entries in a single iteration. Support of the users’ feedbacks is expected to make its internal database constantly increasing. This is especially important for those coding outcomes when Procode must consult dictionary (see Figure 1). Although this approach, in most cases, provides (a) classification code(s), its validity is unknown. The erroneous predictions observed once, if reported, may not appear in the next integration. To prevent biased feedbacks, Procode differentiates “trusted end-users” from the others. For a given user, the administrator periodically verifies the received feedback data and, if valid, the user is labelled as trusted.

Limitations and Outlook

Procode‘s development is ongoing and more efforts are needed to improve different aspects of this tool and perform additional validation tests. As already mentioned, the focus in the first release was on coding. Only 100% agreement between predictions and the manually coded jobs would mean an automatic coding approach in its full picture. This, however, is not the case with the current version of Procode and additional studies are expected to reveal in which domains the data must be enriched or which parts of the coding algorithm should be revised. When more data become available in different languages, the use of the language translations and the dictionary sub-algorithm (Figure 1) will be less used. For the moment, these two are necessary to mitigate the lack of data. An in-depth investigation of their performance is thus essential.

Re-coding between two classifications, although supported, is only possible if a crosswalk exists online or can be established based on a dataset. This is, however, only the case for a limited number of crosswalks. An idea, which has not yet been adopted, is to create a semi-automatic system that would define translation links based on similarity between the job titles in two classifications. The users would then score different outputs based on their validity. Some translations would thus become more valid (i.e. less uncertain) than others. Simultaneously, the recoding algorithm would then learn on how different job titles should be linked and would improve its performance.

Funding

This work was supported jointly by the Swiss State Secretariat for Economic Affairs (SECO) and Federal office of Public Health (OFPH), grant N° 0947002262 “SACoProST”.

Acknowledgements

The authors thank the Constances cohort consortium, particularly M. Zins and M. Goldberg from INSERM for providing data; C. Pilorget from Santé publique France and S.B. Petersen from the University of Copenhagen for providing crosswalks for some French and Danish classifications; and the Swiss federal statistic office for information regarding Swiss classifications. In addition, the authors would like to express their gratitude to Yara Shoman from Unisanté for testing Procode software and providing useful inputs regarding the the manuscript.

Competing Interests

The authors declare no conflict of interest relating to the work presented in this article.

References

Bird, S., Klein, E. & Loper, E. 2009. *Natural Language Processing with Python*, O'Reilly Media, Inc.

Cutler, A., Cutler, D. & Stevens, J. 2011. Random Forests.

De Matteis, S., Jarvis, D., Young, H., et al. 2017. Occupational self-coding and automatic recording (OSCAR): a novel web-based tool to collect and code lifetime job histories in large population-based studies. *Scand J Work Environ Health,* 43**,** 181-186.

Facebook. 2020. *React - A JavaScript library for building user interfaces* [Online]. Available: https://reactjs.org/ [Accessed 11 April 2020].

Fadel, M., Evanoff, B. A., Andersen, J. H., et al. 2020. Not just a research method: If used with caution, can job-exposure matrices be a useful tool in the practice of occupational medicine and public health? *Scand J Work Environ Health*.

Foundation, D. S. 2020. *The Django Project* [Online]. Available: https://www.djangoproject.com/ [Accessed].

Goldberg, M., Carton, M., Descatha, A., et al. 2017. CONSTANCES: a general prospective population-based cohort for occupational and environmental epidemiology: cohort profile. *Occup Environ Med,* 74**,** 66-71.

Ikonomakis, E., Kotsiantis, S. & Tampakas, V. 2005. Text classification: a recent overview. 125.

Koeman, T., Offermans, N. S., Christopher-De Vries, Y., et al. 2013. JEMs and incompatible occupational coding systems: effect of manual and automatic recoding of job codes on exposure assignment. *Ann Occup Hyg,* 57**,** 107-14.

Korde, V. 2012. Text Classification and Classifiers:A Survey. *International Journal of Artificial Intelligence & Applications,* 3**,** 85-99.

Nguyen, E. 2014. Chapter 4 - Text Mining and Network Analysis of Digital Libraries in R. 95 - 115.

Patel, M. D., Rose, K. M., Owens, C. R., et al. 2012. Performance of automated and manual coding systems for occupational data: a case study of historical records. *Am J Ind Med,* 55**,** 228-31.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011. Scikit-learn: Machine Learning in Python. 12**,** 2825–2830.

Peters, S. 2020. Although a valuable method in occupational epidemiology, job-exposure -matrices are no magic fix. *Scandi J Work Environ Health***,** 231-234.

Remen, T., Richardson, L., Pilorget, C., et al. 2018. Development of a Coding and Crosswalk Tool for Occupations and Industries. *Ann Work Expo Health,* 62**,** 796-807.

Rennie, J. D. M., Shih, L., Teevan, J., et al. 2003. Tackling the poor assumptions of naive bayes text classifiers. *Proceedings of the Twentieth International Conference on International Conference on Machine Learning.* Washington, DC, USA: AAAI Press.

Russ, D. E., Ho, K. Y., Colt, J. S., et al. 2016. Computer-based coding of free-text job descriptions to efficiently identify occupations in epidemiological studies. *Occup Environ Med,* 73**,** 417-24.

Warwick Institute for Employment Research, U. O. W., Coventry, Cv4 7al, United Kingdom. 2018. *Cascot: Computer Assisted Structured Coding Tool* [Online]. Available: https://warwick.ac.uk/fac/soc/ier/software/cascot/#:~:text=Cascot%20is%20a%20computer%20program,UK%20Office%20for%20National%20Statistics. [Accessed 14.12.2020 2020].

Yin, T. & Henter, R. 2020. Translate Python Documentation.

Zins, M., Goldberg, M. & Team, C. 2015. The French CONSTANCES population-based cohort: design, inclusion and follow-up. *European journal of epidemiology,* 30**,** 1317-1328.

Figure 1. Coding algorithm workflow

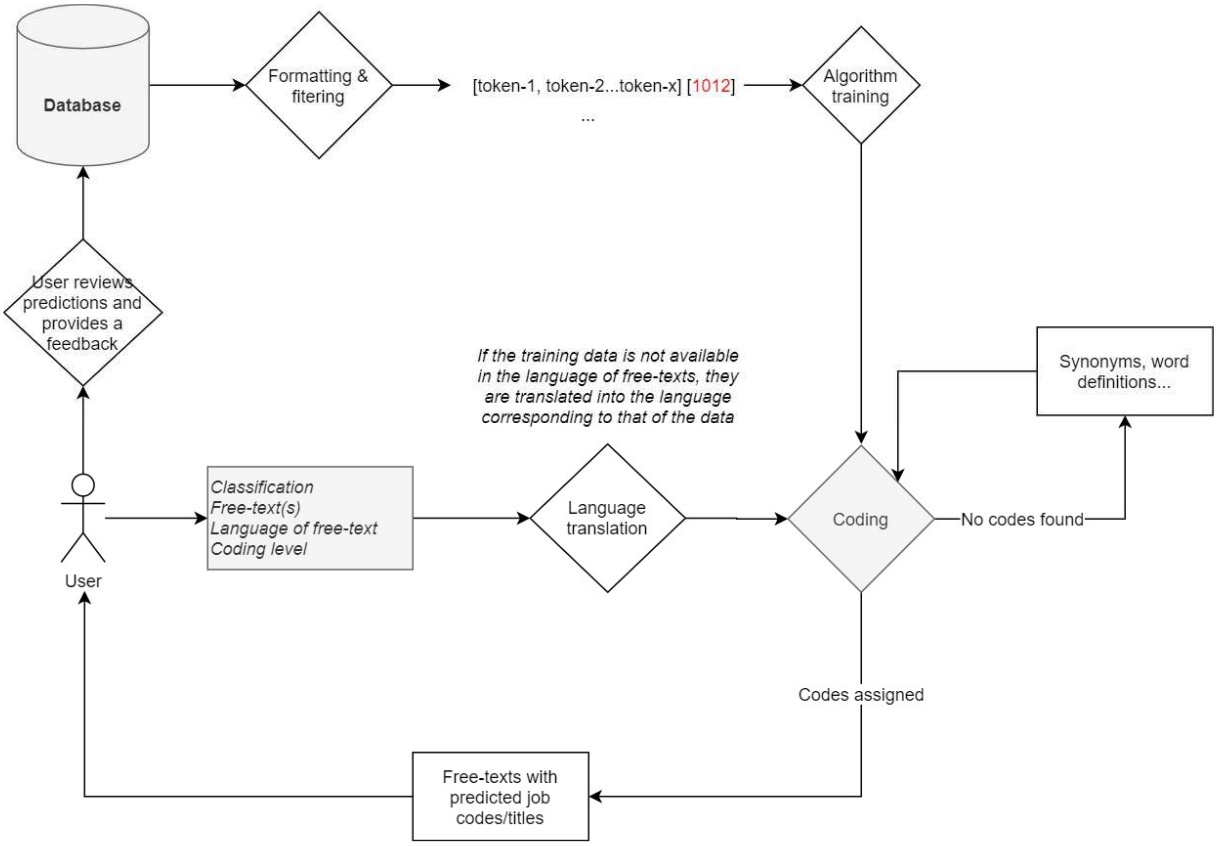
Database
Formatting & fitering
[token-1, token-2...token-x] [1012]
...
Algorithm training
User reviews predictions and provides a feedback
If the training data is not available in the language of free-texts, they are translated into the language corresponding to that of the data
Synonyms, word definitions...
User
Classification
Free-text(s)
Language of free-text
Coding level
Language translation
Coding
No codes found
Codes assigned
Free-texts with predicted job codes/titles

Table 1. Percentage agreement between predicted and manually coded occupations and industries obtained for three classifications (i.e. PCS 2003, NAF 2008 and ISCO 1988) in 5-fold cross-validation and external tests

Classification	Code level	Number of defined occupations/industries	Accuracy, %	
			Cross-validation	External test
PCS 2003	1	8	81	83
	2	24	73	72
	3	42	70	71
	4	497	57	60
NAF 2008	1	21	83	83
	2	88	79	80
	3	272	68	80
	4	615	66	71
	5	732	63	71
ISCO 1988	1	9	-	70
	2	36	-	64
	3	148	-	58
	4	493	-	-