Thesaurus-based Transformation: A Classification Method for Real Dirty Data

Maxime Perrot, Mickaël Baron, Brice Chardin, Stéphane Jean

Thesaurus-based Transformation: A Classification Method for Real Dirty Data

Maxime Perrot[1,2], Mickaël Baron[2], Brice Chardin[2], and Stéphane Jean[3]

[1] Bimedia
[2] LIAS, ISAE-ENSMA
[3] LIAS, Université de Poitiers
firstname.lastname@ensma.fr

Abstract. In this paper, we consider a retail store classification problem supported by a real word dataset. It includes yearly sales from several thousand stores with dirty categorical features on product labels and product hierarchies. Despite the fact that classification is a well-known machine learning problem, current baseline methods are inefficient due to the dirty nature of the data. As a consequence, we propose a practical thesaurus-based transformation. It uses an intermediary global approximate classification of products, based on local products hierarchies. Activities for a subset of stores are human-labeled to serve as ground truth for validation, and enable semi-supervision. Experiments show the effectiveness of our approach compared to baseline methods. These experiments are based on datasets and solutions made available for reproducibility purposes.

1 Introduction

Bimedia is a French company which markets hardware (cash registers) and software for convenience stores such as grocery stores, tobacco stores, bakeries, florists, etc. It supplies more than 6,000 stores with a wide range of activities, archiving approximately 60 million transactions monthly. Given the freedom granted to each customer to manage their own product catalog, Bimedia's data collection is heterogeneous in terms of quality and quantity depending on stores size, activities and customs. Store activities (e.g, grocery or tobacco) are filled in by Bimedia's sales staff when contracts are set up. However, due to the large number of stores, this information is difficult to manage manually, while stores tend to diversify their activities over the years. As a result, the level of confidence in the activities entered in the database is low. Yet, the activities of client stores are valuable, for example to report accurate revenue breakdowns.

In this paper, we address the problem of identifying store activities from their sales. From a scientific point of view, this is a retail store classification problem supported by a real word dataset. Despite the fact that classification is a well-known problem in machine learning, we show in this paper that current baseline solutions are inefficient to accurately classify stores in this case study.

This is mainly due to the dirty nature of the considered real dataset. From our experience, this is also the case for numerous other datasets in practice.

To overcome the limitations of state-of-the-art solutions, we propose a preprocessing step named ThesaurusBT. This method is based on a thesaurus built using business knowledge. This thesaurus enables an intermediary global approximate classification of products, based on multiple examples of hierarchical structures used to group products. Even if the proposed thesaurus is specific to our domain of applications, the same methodology with a different thesaurus could be used for different use cases. Our main contributions in this paper are the following.

1. We propose a thesaurus-based transformation method that uses business knowledge to overcome the dirtiness of data.
2. We evaluate experimentally the impact of our method compared to state-of-the-art solutions.
3. For the purpose of facilitating reproducibility, we offer access to both real datasets and implemented solutions used in our experiments.

This paper is organized as follows. Section 2 details the problem addressed in this paper and the considered use case. Section 3 analyses contributions reporting similar problems and solutions. Section 4 presents our thesaurus-based transformation method. Sections 5 and 6 describe our implementation, experiments and results obtained. We conclude in Section 7 and introduce future work.

2 The store classification problem

In Bimedia's operation, store owners manage their product catalogs with some flexibility on the hierarchical structure used to group products into *families*. There exists two types of families.

- *Global families* are defined by Bimedia. They cover products which are either: 1) subject to special legislation that restrict potential providers, such as tobacco, vape and press, or 2) dematerialized, such as money transfer, prepaid phone cards or gift cards.
- *Local families* are defined by store owners. Any name can be used to label these families. Moreover, products of these families are frequently defined without a global product ID such as a normalized barcode.

The objective is to classify stores according to their activities. Over a third of the stores offer products and services related to activities such as restaurants, bakeries, grocery stores, florists, bars, and more. Products related to these activities belong to local families. These are difficult to handle because of the differences in the naming of the products (synonyms, acronyms or spelling mistakes), differences in the codification systems (normalized scanned barcodes, hand-typed barcodes with potential typos, or even unspecified) and the arrangements into families (by brand, by product types with different granularity, etc.). Therefore, it is almost impossible to compare products between stores and identify corresponding activities without complex data transformations.

Table 1. Dataset sample[4] of products names and families

	Store ID	Product name	Product family
1	ccb...2d6	Tray of Fry	On-site catering
2	3e1...cf7	large fry	Catering to take away 10 - 707140
3	609...ba4	TRAY FRY	ALIM5.5
4	379...949	FRY PLATE ONLY	BAR
5	379...949	MID. TRAY FRY TAKEOUT	Catering to take away
6	aa1...590	Fries Tray	Snack
7	bc3...3d3	TRAY OF FRY	CATERING 10
8	ab1...8ef	NOODLE FRY	Tabletry
9	8ab...c19	PIK FRY	Confectionery 20

We illustrate the considered problem with a simple example from real data of the provided dataset in Table 1. The first seven of the nine selected rows refer to the same product: a portion of fries, but these are expressed by different strings and arranged in various families depending on the stores or even on the way of consumption (as in rows 4 and 5 of the table, corresponding to the same store). Moreover, it is not possible to rely on a product ID (e.g., a barcode) to map products between stores, as this identifier is generated locally for this type of product. A data transformation step is required. The original dataset has more than 2 millions unique values for product names, and more than 6,000 stores with variable revenues and sales quantities. As the illustrated problem is recurring, classifying store activities becomes complex.

The store classification problem can be broken down into three sub-problems: 1) dirty categorical variables encoding: as illustrated in Table 1, sales data is structured under categories with a two-level (products and families) local hierarchy and dirty textual values, 2) variable input shapes: raw input sales data is, for each store, of variable length and non-sequential, its transformation is therefore not trivial, 3) multi-labeling classification: one or more business activities can be assigned to stores, depending on their sales.

Table 2 illustrates some of the raw data extracted from transactions for three stores that we wish to classify, along with the expected output. In this sample, we can observe that stores sell a various number of products, resulting into a different number of rows per store. The hardest part is not to assign labels to stores, but to identify relationships between products from different stores. In this paper, we consider the problem as a whole, including the multi-labeling task, because we are not able to produce a consistent validation dataset for products (2 millions products), while it is feasible for stores activities (6,000 stores). Consequently, the multi-labeling task will mainly be included in this study as a way to evaluate the quality of the data transformation step.

[4] Irrelevant columns are not displayed, irrelevant information are cropped and textual values are translated, as literally as possible, from French to English in all Figures.

Table 2. Dataset sample of sales with targeted activities predictions

Store ID	Barcode	Product	Family	Quantity	Activities
db...5c2	31...01	Corn Bread	Bread	4	Bakery
	31...02	Bread	Bread	2338	
	31...04	Baguette	Bread	13377	
	31...51	Cookie	Pastry	2378	
e9...f43	00...03	Larks Pie	Bakery	6315	Bakery
	00...19	Traditional	Bakery	135445	
	31...07	Almond cream galette...	Patisserie	3	
bb...49f	04...94	ZIPPO GASOLINE	Other smoking items	40	Tobacco store Coffee shop
	11...10	Clearomiser Q16 PRO	Misc 20.	3	
	31...45	COFFEE	HOT	83253	
	_A...36	PHILIP MORRIS 20	Cigarettes	26125	
	_A...65	NEWS RED 20	Cigarettes	6126	

3 Related work

Many methods exist to encode text-based categorical variables with synonyms and morphological variability, such as Bert [11], Gamma-Poisson [4], Similarity Encoding [6], MinHash [3], PairClass [12], and others [5, 13]. Based on the experiments conducted by Cerda and Varoquaux [5], the MinHash technique provided the best overall performance for this task, better than large language models.

Processing methods for arbitrary length inputs can be distinguished into two groups: input processing and problem transformation [2]. Due to the difficulty to implement and adapt problem transformation techniques to our context, those are not covered. Input processing techniques include truncation [8], padding [9], aggregation, PCA [1] and models with feature selection, such as XGBOOST [7] and CatBoost [10]. Since our dataset is non-sequential, many other solutions cannot be considered. The basic approach to deal with arbitrary length inputs in our case is the aggregation data pivoting transformation. Truncation, padding and others do not make sense given the shape of our dataset. Experiments conducted by Borisov *et al.* [2] highlight CatBoost as an efficient solution for feature selection.

4 Thesaurus-based transformation

The basis of our proposition is a thesaurus-based transformation, abbreviated as ThesaurusBT, that groups products into categories before labeling store activities. Product categorization is performed for two reasons. First, being able to categorize products sold by stores, independently of the store's catalog, serves multiple purposes, including generating dashboards and targeted advertising. This capability makes our method superior to non-explanatory encoding solutions for companies seeking understandable product labeling. Second, as previously mentioned, the primary challenge posed by the multi-labeling of store activities does not lie in the labeling process itself, but in the data transformation

required to encode product sales. Specifically, it involves encoding dirty textual categorical variables such as products or families labels, which often have a high cardinality. Classifying products represents a solution for this task.

To categorize products, our approach is based on a thesaurus that links recurring terms occurring in family labels to coarse-grained categories defined by the company. In our use case, we have defined a total of 60 product categories, plus an *unknown* category for unidentified product types. Examples of these categories are: *takeaway catering*, *clothes*, or *tobacco*. Our assumption is that, on average over the whole dataset, family labels can be correctly mapped to predefined categories. We consider this assumption to be reasonable when the company, as is the case with Bimedia, possesses significant insight about the kind of products sold by their stores. Thus, ThesaurusBT requires some business knowledge to be applicable, namely:

1. a list of categories of products that the company wants to identify,
2. a dictionary mapping common keywords—i.e. n-grams that are commonly used by store owner—to product categories,
3. (optional) a list of ambiguous keywords, that are commonly associated with several categories, along with differentiation rules. For example, the word *drink* could be used to describe both alcoholic and non-alcoholic drinks. As the VAT (Value Added Tax) applied to alcoholic and non-alcoholic drinks differs under French law, it can be used to resolve ambiguous keywords.

The goal of ThesaurusBT is to automatically create a mapping dictionary between products sold by stores and categories provided by the company. Our method is based on the following principle. While store owners retain the freedom to assign labels to their products and to group them into families, due to the vast number of stores, it is likely that a significant proportion of them will feature some keywords identified by business experts.

The first processing step is a partial cleaning of textual values of dirty categorical variables to encode. This cleaning step includes conversion to lower case, removal of extra spaces, accents, special characters, stop words—including domain-specific stop words.

The second step identifies products that can be categorized using simple rules. In Bimedia's case, this is performed on global families, which are normalized and supervised by the company. In the provided dataset, the identification and the labeling of global families products is based on the family identifier field and a mapping between family identifiers and categories. The remaining processing steps only apply to product from families with dirty labels—local families in Bimedia's case.

The third step creates unique identifiers for product instances. This identification does not have to be exact: some instances can be incorrectly merged or kept separate without significantly impacting the whole process. Best results are still obtained when erroneous identification are minimized. In our dataset, this step starts with the identification of store-internal product identifiers and generic product identifiers, such as normalized barcodes. Internal identifiers generated by Bimedia software are local to a store, and cannot be used to map a product

between one store and another. The creation of a unique product identifier in this case is therefore performed using the product name.

The fourth step lists family labels for each product identifier. It then computes the number of occurrences of each predefined keyword (n-gram) within this list. The most frequent keyword is used to identify the category.

The result of these steps is a mapping from products to categories. In our use case, ThesaurusBT categorizes 94% of products appearing in the dataset. This process does not guarantee a correct categorization of products, and its accuracy is difficult to assess due to the lack of an annotated dataset. We consider this categorization usable in practice if, when included as a transformation step, it improves the accuracy of another classification workflow. This validation is described in our experiments, where we compared ThesaurusBT with existing methods.

5 Experimental design

The originality of our approach concerns the data transformation step that we use for the multi-labeling task (finding the activities of store). As a consequence, the aim of the experiments is to compare the efficiency of baseline machine learning approaches with our method on the data transformation step. We assume that the performance of the data transformation step has a direct impact on the accuracy of the multi-labeling results for store activities. Datasets and implementations are available online[5].

We provide two sets of real data supplied by Bimedia. The sales dataset is composed of anonymized data over a one-year period for 2325 selected stores. This dataset variables are: store identifier, product barcode, product label, family identifier, family label, total amount sold during the year and VAT rate. The dataset contains 11,637,397 rows, 817,385 unique product names and 8,427 unique family labels. The labeled store activity dataset is composed of 400 stores with their activities annotated by experts. These stores were randomly selected from the previous sales dataset and their activities were annotated based on sales aggregated by product family and information extracted from external sources. This dataset contains two variables: a store identifier and activity tags. There are 9 possibles activity tags and 29 unique combinations of those in this dataset.

We use accuracy and macro F1 scores to evaluate multi-labeling performance. Considering that store activities are very unbalanced, we need a model that is efficient in identifying both rare activities, such as hotels, and common ones, such as tobacco shops. The macro F1 score is a reliable metric for identifying models that meet these criteria. For our experiments, 300 annotated stores are randomly selected as the training dataset and the remaining 100 are the test dataset.

We divide our experiments into three workflows: a *basic approach* **(1)**, a *literature approach* **(2)** and a *business specific approach* **(3)**. These are illustrated

[5] `https://forge.lias-lab.fr/thesaurusbt`

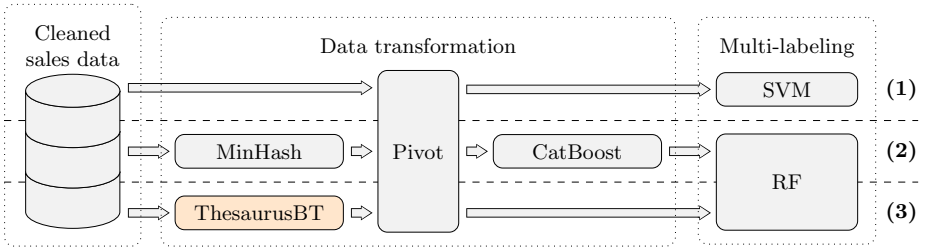

Fig. 1. Processing steps of considered workflows

Cleaned sales data **(1)**

Store ID	Barcode	Product	Family	Quantity	VAT
db...5c2	31...02	bread	bread	2338	5.5
db...5c2	31...51	cookie	pastry	2378	5.5
bb...49f	11...10	clearomiser q16 pro misc 20		3	20
bb...49f	31...45	coffee	hot	83253	7

ThesaurusBT categorization **(2a)**

Product identifier	Family	Store count	Category
coffee	hot drink	675	on-site non-alc. drink
	hot	12	
11...10	e-cigarette	341	vape
	misc 20	2	

Sales data with categories **(2b)**

Store ID	Product ID	Quantity	Category
db...5c2	bread	2338	bakery
db...5c2	31...51	2378	food
bb...49f	11...10	3	vape
bb...49f	coffee	83253	on-site non-alc. drink

Input of the multi-labeling model **(3)**

Store ID	Vape	On-site non-alc. drink	Bakery ...
db...5c2	0	14	2338 ...
bb...49f	3	83253	0 ...

Fig. 2. Transformation steps of the business worklow

in Figure 1 and further described in this section. Additional workflows are also considered and evaluated, mixing the processing steps from the three aforementioned workflows, excluding some non-relevant combinations. Hyperparameters for the CatBoost feature selection, Random Forest and SVM are optimized for each workflow using grid search with cross-validation.

Basic approach The basic approach is a three-step workflow using baseline methods: 1) data cleaning (conversion to lower case, removal of extra spaces, accents, special characters and stop words, lemmatisation), 2) pivoting with the store identifier as the key, product family labels as columns, and the sum of product sold as values, and 3) multi-labeling with a Binary Relevance model with SVM as its base classifier.

Literature approach The literature approach consists of five steps. It uses the best methods reported in the literature: 1) data cleaning (same as for the basic approach), 2) MinHash encoding on the product family labels, 3) pivoting with the store identifier as the key and product family signatures generated by MinHash as columns, 4) feature selection using CatBoost, and 5) multi-labeling with a Binary Relevance model with Random Forest as its base classifier.

Business-specific approach The business-specific approach consists of four steps: 1) data cleaning (same as for the basic approach), 2) application of the ThesaurusBT method, resulting in a mapping from products to categories, 3) piv-

Table 3. Multi-labeling classification experimental results

Data transformation	Classifier	Classifier input dimensions	Accuracy	Macro F1	Execution time (min)
Pivot	RF	6574	0.64	0.48	0.41
	SVM	6574	0.77	0.61	0.23
Pivot–CatBoost	RF	56	0.79	0.61	5.4
	SVM	56	0.77	0.64	5.23
MinHash–Pivot	RF	5104	0.65	0.48	52.18
	SVM	5104	0.77	0.61	52.16
MinHash–Pivot–CatBoost	RF	50	0.79	0.62	57.18
	SVM	50	0.80	0.55	57.02
ThesaurusBT–Pivot	RF	61	0.85	0.66	9.16
	SVM	61	0.80	**0.73**	9.01
ThesaurusBT–Pivot–CatBoost	RF	34	**0.87**	0.67	14.33
	SVM	34	0.80	0.72	14.16

oting with the store identifier as the key and categories (identified by ThesaurusBT) as columns, and 4) multi-labeling with a Binary Relevance model with Random Forest as its base classifier.

The sequence of transformations for the business-specific approach is shown in Figure 2. ThesaurusBT creates a new product identifier (subfigure 2-2a) using either its barcode or its label, depending on the rules defined to detect generated local product identifiers or global IDs. For each product, family labels are listed along with the corresponding number of store. Each product is then processed by ThesauruBT to search for the most common keyword within family labels, weighted by store count. When a match occurs, the product is tagged with the corresponding category, or *unknown* if there is no match. ThesaurusBT product categories are merged with the sales dataset (subfigure 2-2b) before pivoting (subfigure 2-3). This pivot table is built with the store identifier as the key, categories identified by ThesaurusBT as columns, and the sum of quantities sold as values. This result is then used as input for the multi-labeling model.

6 Results and discussion

Table 3 reports total execution times for each workflow. These workflows, implemented in Python, were executed on an i7-11800H 2.30 GHz CPU and 16 GB of RAM. Execution times include both training and scoring. The execution time of the literature workflow is dominated by the hash calculation step of the MinHash method (52 min). Other steps with execution times higher than a minute are ThesaurusBT (9 min) and CatBoost (5 min). Data preparation based on ThesaurusBT is therefore significantly slower than a simple pivoting method (3 seconds).

The accuracy and macro F1 score of the multi-labeling classification task are presented in Table 3. Without CatBoost, the MinHash method does not significantly improve classification performance but reduces the complexity of the dataset from 6574 to 5104 variables. It does not succeed in improving the

performances by deduplicating family labels with multiple morphological values (typos, misspellings, etc.) and increases the execution time significantly. This insignificant impact on performance can be explained by the characteristics of family labels, which include synonyms and variable strategies of arrangement of products (by type, by brands, etc.) with variable granularity. These cannot be captured by MinHash. Moreover, increasing the MinHash sensibility leads to the grouping of unrelated families.

CatBoost can significantly improve the performance of workflows with RF-based classifiers when it is used in conjunction with Pivot (+23% accuracy and +27% macro F1) or MinHash (+22% accuracy and +29% macro F1). These are positive results considering that this method also reduces the complexity of the dataset from 6574 (resp. 5104) to 56 (resp. 50) variables. The benefits of CatBoost are lower when used with SVM as the base classifier. When CatBoost is used in conjunction with ThesaurusBT, its feature selection has close to no impact on classification performance (between -1% and +2%).

A conclusion drawn from these results is that ThesaurusBT outperforms all other transformations. The improvement is especially visible with the macro F1 score. Based on these performances, we can assume that this method has successfully labeled a significant part of the products, partially solving the data transformation problem. If we focus on the macro F1 score, the results are heterogeneous due to the unbalanced nature of the dataset (for instance, there are many tobacco shops, but not many hotels or restaurants). In fact, the models are overfitted to recognize tobacco shops or newspapers, but underfitted to recognize hotels or restaurants. This directly affects the macro F1 score for all models that were not able to detect rare cases. The best performing models in that regard are those that include ThesaurusBT in their workflow. More generally, a significant improvement is achieved by incorporating ThesaurusBT, as shown in Table 3, since the business-specific solution is always the most efficient.

7 Conclusion

In this paper, we have considered the problem of assigning activity tags to stores according to their sales. Even if we have considered a specific use case provided by the Bimedia company, this is a general problem that affects software providers of different industries. Compared to the use cases found in the literature, the one proposed in this paper raises the challenges of dirty data as stores can use any label to name their products and families of products. As a baseline, we have proposed and implemented two approaches: one with baseline methods and one with the most efficient methods known in the literature. All the datasets and implementations are available online for reproducible purpose. These approaches are compared with the one we proposed named ThesaurusBT, a thesaurus-based transformation. This transformation is based on some business knowledge that a company such as Bimedia can provide and maintain over time. As we have shown in our experiments, this approach outperforms the baseline approaches for the task of labeling store activities using dirty data. As a drawback, the

implementation of this solution incurs a significant business-specific development cost. By making this dataset available, Bimedia wishes to raise interest for this kind of problem, as having an efficient machine learning solution, with limited human involvement, usable by a medium-scale company would bring a significant improvement to this field.

As a future work, we plan to consider deep learning approaches. One difficulty is to have enough training data. Thus, automating the production of such data is a perspective of our work. Another challenge is to extend the business knowledge used by our approach. Currently, we only use lexical resources but we are convinced that more complex models such as ontologies could be useful to improve our method.

References

1. H. Abdi and L.J. Williams: Principal component analysis. Wiley interdisciplinary reviews: computational statistics 2(4), 433–459 (2010)
2. V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci: Deep Neural Networks and Tabular Data: A Survey. IEEE Transactions on Neural Networks and Learning Systems (2022)
3. A.Z. Broder: On the resemblance and containment of documents. In: Proceedings. Compression and Complexity of SEQUENCES 1997, pp. 21–29 (1997)
4. J. Canny: GaP: a factor model for discrete data. In: Proceedings of the 27th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 122–129 (2004)
5. P. Cerda and G. Varoquaux: Encoding high-cardinality string categorical variables. IEEE Transactions on Knowledge and Data Engineering (2020)
6. P. Cerda, G. Varoquaux, and B. Kégl: Similarity encoding for learning with dirty categorical variables. Machine Learning 107(8), 1477–1494 (2018)
7. T. Chen and C. Guestrin: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
8. J.F. Crow and M. Kimura: Efficiency of truncation selection. Proceedings of the National Academy of Sciences 76(1), 396–399 (1979)
9. M. Dwarampudi and N. Reddy: Effects of padding on LSTMs and CNNs. arXiv preprint arXiv:1903.07288 (2019)
10. L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, and A. Gulin: CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems 31 (2018)
11. I. Tenney, D. Das, and E. Pavlick: BERT Rediscovers the Classical NLP Pipeline. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4593–4601 (2019)
12. P.D. Turney: A uniform approach to analogies, synonyms, antonyms, and associations. 22nd International Conference on Computational Linguistics (COLING-08) (2008)
13. S. Wu and U. Manber: Fast text searching: allowing errors. Communications of the ACM 35(10), 83–91 (1992)