Data Quality Estimation Framework for Faster Tax Code Classification

Ravi Kondadadi and **Allen Williams** and **Nicolas Nicolov**
Avalara Inc.
255 South King St., Suite 1800
Seattle, WA 98104
{*ravi.kondadadi, allen.williams, nicolas.nicolov*}*@avalara.com*

Abstract

This paper describes a novel framework to estimate the data quality of a collection of product descriptions to identify required relevant information for accurate product listing classification for tax-code assignment. Our Data Quality Estimation (DQE) framework consists of a Question Answering (QA) based attribute-value extraction model to identify missing attributes and a classification model to identify bad quality records. We show that our framework can accurately predict the quality of product descriptions. In addition to identifying low-quality product listings, our framework can also generate a detailed report at a category level showing missing product information resulting in a better customer experience.

1 Introduction

As a global tax compliance company, Avalara enables businesses to use the correct sales tax rate by mapping their product catalogs to a tax code taxonomy built by Avalara. The tax codes, in turn, inform the tax calculation engine how to apply the tax for a transaction. This mapping process is very laborious today due to many reasons. One of the main challenges is the quality of the product catalog data we receive from customers. Many times, this data is quite vague and noisy. This can be caused by many factors.

1. Not enough context about the business: For tax code classification, we only receive a collection of product titles. This product information does not give enough context about the industry in general, causing problems in tax code mapping, especially if the language in the product information is ambiguous. This lack of context results in the mapping team having to talk to the business to get more information about the business and the corresponding industry. This is a very tedious process requiring a lot of manual effort, causing delays in the customer onboarding process.

2. Missing attributes in the product titles and descriptions: Many product descriptions do not have relevant attributes. This makes it hard for the models to map the products in the catalog to applicable tax codes. For example, a clothing product without specific attributes like knitted/crocheted cannot be mapped to the appropriate tax code.

3. Product information contains rare words, and acronyms: If the product information includes words that were not seen before, acronyms or abbreviations, it makes it harder for the model to classify.

4. The industry of the business is unknown or not currently covered by the tax code taxonomy: If the business belongs to a new sector or belongs to an industry with low tax code coverage, the mapping would be more challenging.

A model including these factors to identify the quality of product titles would help the mapping team request additional information for those products from the business and accelerate the onboarding process for that customer.

In this paper, we describe a novel data quality estimation framework which businesses can interact with and provide all relevant information required to map all entries in a product catalog to the corresponding tax codes. Iteratively, the tool can map input product records to tax codes, identify low quality records and present pertinent questions to the user for the bad records. The tool repeats the process until all records are fixed, and the mappings are complete for the entire product catalog.

Next, we discuss our Data Quality Estimation framework. We then describe our methodology and experiments followed by relevant recent work.

Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5), pages 29 - 34

2 Data Quality Estimation Framework

In this section, we present details of the Data Quality Estimation framework. The framework includes a tax code classification model, an attribute-value extraction model, and a quality assessment model. Next, we will discuss each of these components in detail.

2.1 Tax Code Classification

The Avalara Tax code system consists of thousands of codes hierarchically organized by categories and the nature of the business. The codes fall into a dozen major categories ranging from products to food and beverages. The automatic tax code classification system is responsible for identifying the appropriate tax code for any given product in a customer's inventory catalog. The tax codes are mapped when the customer is onboarded to the Avalara system. The classification system at Avalara uses a tiered approach where a top-level model predicts the probable category, and then a category-specific model predicts a probable tax code. This approach was chosen predominantly to keep the number of labels for each model down to a manageable number and allow for targeted improvements for each category without interfering with other categories. Each of the models is a BERT (Devlin et al., 2019) model fine-tuned for classification.

2.2 Attribute Value Extraction (AVE)

The most important parts of product information to determine the relevant tax code are the product title and product description. An attribute is a feature that describes a specific property of a product. Some examples of attributes include brand, color, material, etc. An attribute-value is a particular value assumed by the attribute. For example, for the product title *"Apple iPhone 13 Pro, 128GB, Sierra Blue"*, iPhone is the main entity. The corresponding attribute-values are "Apple", "13 pro" and "Sierra Blue". Apple is the brand, "13 pro" is the model and "Sierra Blue" is the color.

The presence of attributes is quite important to classifying a product title to the most relevant tax code. Often, we lack attribute information in the product title data we receive from our customers. This usually results in lot of back and forth with the customer and causes significant delays in the time to fully onboard a customer. A model that can extract attribute-values from product titles and identify missing attributes would be of great help in determining the quality of the customer data.

Input to the attribute-value extraction model would include the product listing and a set of attributes. These attributes come from a tax code ontology developed internally by Avalara that covers a wide range of tax code categories. The tax code classification model is used to identify the relevant category for the product listing. We can then identify the related attributes for that category from the ontology.

For our experiments, we formulated the attribute-value extraction as a Question Answering problem as mentioned in (Wang et al., 2020a). The advantage of a Question Answering (QA) formulation is that it can scale well with more attributes and can work well with unseen attributes in the training data. We can treat the product listing as the document, attribute name as the question and retrieve the value as the answer. We used the MAVE dataset (Yang et al., 2021) for training the QA model. MAVE is a product dataset for Multi-source Attribute-Value Extraction, created by Google. MAVE is the largest product attribute-value extraction dataset by the number of attribute-value examples containing over 3M attribute-value annotations from 2.2M Amazon product descriptions.

2.3 Quality Assessment

The goal of the quality assessment model is to identify the product listings that require more information in order to be correctly mapped to the relevant tax codes. We created a logistic regression (LR) (Cox, 1958) model for this classification task. Our features include prediction probabilities from the tax code classification model, missing attribute information, title length, and category meta data, etc.

Here is an overview of the steps involved in running the framework.

1. First run the current tax code classification model.
2. Remove the records with good predictions based on the prediction probabilities.
3. For the remaining records, run the attribute-value extraction to identify missing attributes.
4. Identify the quality score using the quality assessment model.
5. Generate a detailed report listing relevant questions for each category to cover the ma-

jority of the bad records and share the report with the user for feedback.

6. Repeat steps 1-5 on the updated records set from the user until the number of bad records fall below a predefined threshold.

3 Experimental Results

In this section, we present the evaluation of both the attribute-value extraction and the quality assessment models.

3.1 Attribute-Value Extraction Evaluation

We evaluated various attribute-value extraction methods on two different datasets.

1. MAVE dataset: A random subset of around 10,000 records from MAVE for evaluation.

2. Compliance dataset: A subset of around 1,000 product listings that was manually annotated with attribute-value information.

We compared two different approaches for attribute-value extraction.

1. AVE-FT: This is a hybrid approach of look-up and classification. We created a list of 1,200 most frequent attributes and the possible values they could take. For example, for the attribute "material-type" we included phrases like “plastic”, “pvc”, “synthetic rubber” as possible values. We first check if we can find an attribute-value in the listing using a look-up approach. We used a SpaCy (Honnibal and Montani, 2017) matcher to identify such values. In order to work with attributes that are not in our list, we used a fastText (Bojanowski et al., 2017) model to identify if a product listing contains a specific attribute. The model was trained on our historical data where we know whether a specific attribute is present.

2. AVE-QA: We developed a QA model fine-tuned on a DistilBERT (AVE-QA-DISTILBERT) (Sanh et al., 2019) model on a subset of records from the MAVE dataset as mentioned in (Wang et al., 2020a). We also created a different QA model by fine-tuning on the MiniLM model (AVE-QA-MINILM) (Wang et al., 2020b).

We used the F1-score metric as defined in the SQuAD (Rajpurkar et al., 2016) evaluation dataset for Question Answering.

Attribute-Value Extractor	**MAVE F1-score**	**Compliance set F1-score**
AVE-FT	0.15	0.19
AVE-QA-DISTILBERT	0.95	0.54
AVE-QA-MINILM	0.93	0.63

Table 1: Comparison of various attribute-value extraction methods on the MAVE and Compliance datasets

Table 1 shows the evaluation of various attribute-value extraction methods on the MAVE and the Compliance datasets. We can see that the Question-Answering based models outperformed the FastText baseline on both datasets and MiniLM performs slightly better than Distilbert version. Not surprisingly, both QA models performed well on the MAVE datasets as the QA models were fine-tuned on MAVE. The compliance dataset includes attributes related to domains like Insurance and Medical care whereas the MAVE data was predominantly about e-commerce. Although our performance is currently low on the compliance dataset, we are working on augmenting MAVE with compliance related information and retraining the model with more compliance data.

3.2 Evaluation of Data Quality Estimation

In this section, we compare different configurations of Attribute-value extraction and Data quality assessment for estimating the data quality of a set of product listings. In addition to the Logistic Regression model for quality assessment, we also evaluated two baselines. The first baseline simply predicts the quality based on the prediction probability. The second baseline includes both tax-code level precision (this can be determined from the historical performance of the tax-code) and prediction probabilities. For the evaluation, we used the same sample of 1,000 listings from the attribute-value extraction experiment. The dataset was reviewed by our tax coding experts to classify each listing as good/bad quality. Table 2 shows the evaluation of various data quality estimation methods. For this experiment, we used a threshold of 0.5 for prediction probabilities. It can be seen from the results that our classification model for data quality assessment outperformed the baselines based on prediction probabilities and tax-code level precision. Although adding the missing attribute

Attribute-Value Extractor	Data Quality Estimator	F1-score
None	Pred.Prob. Thresholding	0.755
None	Pred.Prob.+Taxcode precision	0.741
None	Logistic Regression	0.823
AVE-FT	Logistic Regression	0.826
AVE-QA-MINILM	Logistic Regression	0.828

Table 2: Comparison of various data quality estimation models

information to the model did not help, it is useful in explaining why the data is inadequate to our customers.

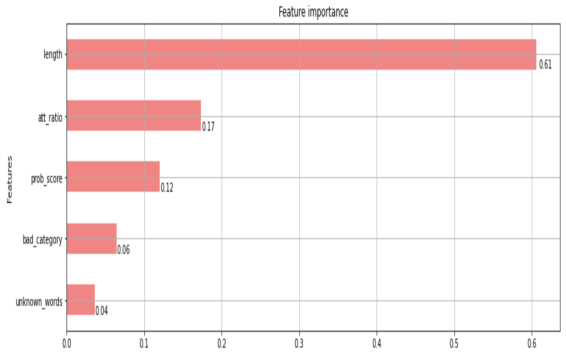

Figure 1: Importance of quality estimation features

Figure 1 shows the importance of various features in the quality estimation model. It can be seen that title length and missing attribute information are the most important features for quality estimation. It also shows that using attribute value extraction model alone is not enough in assessing the data quality of product listings.

We generate a summary report at a category level showing the missing attribute information to help our customers understand how they can enhance their product descriptions. Figure 2 shows a sample screenshot of the detailed report showing missing attribute information at a category level. We are currently working on including this tool in production to estimate the data quality of product catalogs from our customers.

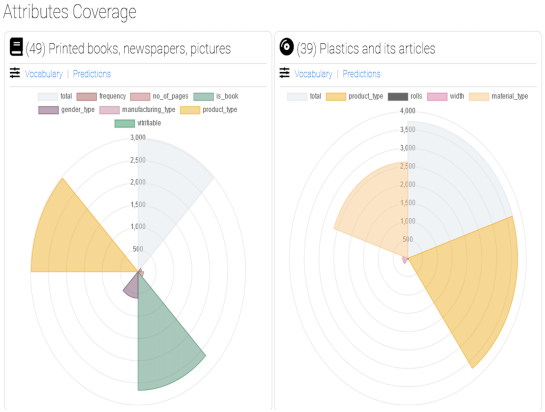

Figure 2: A sample data quality report

4 Related Work

The quality of predictions of a machine learning model is dependent on the quality of the data it is trained on. Poor data results in bad predictions from the model, translating to a poor customer experience. Due to increased usage of data in businesses, researchers have been seeking to define data quality. Batini et al. (2009) compares data quality and assessment methodologies along several dimensions. Pipino et al. (2002), Cai and Zhu (2015) identify various dimensions for defining data quality. ONeill (2020) proposed a decision tree algorithm to predict data quality. Schelter et al. (2018) proposed a declarative API to "unit-test" data. They also discussed methods such as anomaly detection to assess data quality. Active learning (Settles, 2009) has also been used to determine most confusing entries in a dataset. Active learning suggests labeling samples that are most uncertain based on prediction probabilities. But the prediction probabilities are not always good enough to identify data quality and to understand what information is missing from the product listings.

Attribute-value extraction was predominantly solved using rule-based approaches (Nadeau and Sekine 2007; Vandic et al. 2012) in the past. The disadvantage with these methods is that they are domain-specific and require extensive feature engineering. More recently, with the advances in Neural Networks-based methods, approaches like BiLSTM-CRF (Kozareva et al. 2016; Zheng et al. 2018) have been proposed. Wang et al. (2020a) formulated attribute extraction as a Question Answering problem. They proposed a multi-task framework to address generalizability. Yang et al. (2021)

extended this work by adopting an ETC encoder (Ainslie et al., 2020) to generate the contextual embeddings for title and description of the product listing to handle longer descriptions.

5 Conclusion

We presented a novel data quality estimation framework for the e-commerce domain that can identify product listings with incomplete information. The framework includes a Question Answering based attribute-value extraction model trained on the MAVE dataset. We prove that our framework can reliably identify inadequate product listings resulting in faster tax code classification.

Beyond mapping products to tax codes, our framework is applicable to services (in fact, our top-level categories already include a Services group), as well as, utilities/energy, or in general any domain where items can be described in terms of attributes and values. We are applying this framework to other tax code ontologies like the Harmonized Commodity Description and Coding System (HS) which provides codes for traded products as part of international transactions.

6 Acknowledgements

We would like to thank our coworkers Mike Lash and Brandon Van Volkenburgh for helping us with the data annotation. We also would like to thank Vsu Subramanian, and Rajesh Muppalla for their support and valuable feedback.

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Kenneth Fisher, Philip Pham, Anirudh Ravula, Sumit K. Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *EMNLP*.

Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. 2009. Methodologies for data quality assessment and improvement. *ACM computing surveys (CSUR)*, 41(3):1–52.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Li Cai and Yangyong Zhu. 2015. The challenges of data quality and data quality assessment in the big data era. *Data science journal*, 14.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Zornitsa Kozareva, Qi Li, Ke Zhai, and Weiwei Guo. 2016. Recognizing salient entities in shopping queries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 107–111.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Allen ONeill. 2020. Data quality evaluation using probability models. *arXiv preprint arXiv:2009.06672*.

Leo L Pipino, Yang W Lee, and Richard Y Wang. 2002. Data quality assessment. *Communications of the ACM*, 45(4):211–218.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Victor Sanh, L Debut, J Chaumond, and T Wolf. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arxiv 2019. *arXiv preprint arXiv:1910.01108*.

Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger. 2018. Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, 11(12):1781–1794.

Burr Settles. 2009. Active learning literature survey.

Damir Vandic, Jan-Willem Van Dam, and Flavius Frasincar. 2012. Faceted product search powered by the semantic web. *Decision Support Systems*, 53(3):425–437.

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020a. Learning to extract attribute value from product via question answering: A multi-task approach. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 47–55.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression

of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Li Yang, Qifan Wang, Zac Yu, Anand Kulkarni, Sumit Sanghai, Bin Shu, Jon Elsas, and Bhargav Kanagal. 2021. Mave: A product dataset for multi-source attribute value extraction.

Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. Opentag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1049–1058.