Applying Natural Language Processing for Initial Categorizing of Product Descriptions

Nikolay Aristov, Thomas Koch, Elenna R. Dugundji
Center for Transportation and Logistics
Massachusetts Institute of Technology
Cambridge, MA, USA
naristov@mit.edu, thakoch@mit.edu, elenna_d@mit.edu

***Abstract*—The procurement process is a critical aspect of any company's operations, and optimizing it can lead to significant cost savings. This study presents a methodology for enhancing the categorization of procurement items by mapping them to the United Nations Standard Products and Services Code (UNSPSC). By applying this approach, companies can analyze procurement patterns more effectively, consolidate purchases, and negotiate better terms with suppliers, thereby reducing overall costs.**

Given the lack of substantial labeled training data often encountered in real-world scenarios, traditional machine learning techniques for categorization are not feasible. To overcome this challenge, we employed Natural Language Processing (NLP) techniques using spaCy to clean and preprocess item descriptions, followed by embedding methods to generate vector representations for both the procurement data and UNSPSC categories.

Our approach resulted in the initial mapping and categorization of approximately 29% of the dataset, providing a solid foundation for further analysis. This initial success demonstrates the potential for significant improvements in procurement efficiency. Additionally, we propose several advanced techniques to enhance the categorization process, suggesting that a higher categorization rate is achievable.

***Index Terms*—natural language processing, categorizing, procurement, spaCy, SBERT**

I. Introduction

Efficient procurement practices are crucial for organizations aiming to streamline operational costs and improve decision-making capabilities. The classification of items in Maintenance, Repair, and Operations (MRO) procurement is vital in this regard. However, many companies, especially those with legacy systems, struggle with systems that lack the precision and detail required for effective expenditure analysis.

By accurately categorizing items, a company can identify areas of overspending and negotiate better contracts with suppliers. Such optimization of procurement processes leads to significant cost savings, increased operational efficiency and improved vendor management.

However, the primary challenge in procurement categorization lies in the diversity and unstructured nature of item descriptions. Also, it is impossible to apply machine learning models directly, as they typically require extensive labeled datasets for training. Unfortunately, such datasets are often unavailable in the procurement domain, and, therefore, to make possible accurate and efficient categorization of procurement items, additional processing is required.

Existing works on Natural Language Processing (NLP) mostly focus on text analysis outside of the supply chain domain, leaving a significant gap in leveraging these techniques for procurement and supply chain optimization. This study describes an application of NLP techniques to categorize items into the United Nations Standard Products and Services Code (UNSPSC) hierarchy. This work performs the initial mapping of items by transforming and embedding procurement descriptions into a structured format, as well as processing UNSPSC descriptions at the commodity level.

To tackle the categorization problem, this work utilized spaCy [1], a powerful NLP library, for data cleaning and preprocessing. Item descriptions were standardized and transformed into vectors using embedding techniques, allowing for a more structured comparison with UNSPSC categories. Initial mapping was performed, yielding a categorization success rate of approximately 29% (51,088 of 175,113 unique rows).

In the sources we researched (for example, [2], [3], [4]), the success rates reported typically pertain to testing datasets where a pre-labeled training dataset was available. However, these studies often do not disclose the methods used to obtain their labeled data, and datasets can vary widely, including ours. This makes a direct comparison of success rates challenging. Given the uniqueness of datasets and the absence of labeled training data, achieving a 29% success rate is quite promising, particularly in terms of saving considerable time in the categorization process. Furthermore, with future enhancements, we anticipate that this rate could be improved.

This study makes several key contributions: It presents an approach to procurement categorization in the absence of labeled data; demonstrates the efficacy of NLP and embedding techniques in this domain; and provides a preliminary categorization framework.

II. State of Practice

In this chapter, we discuss the current methodologies and practices in applying Natural Language Processing (NLP) to the classification of products and services under the United Nations Standard Products and Services Code (UNSPSC). UNSPSC is a global classification system for products and services that has a hierarchical structure, comprising segments, families, and, finally, commodities at the lowest level.

Publicly available datasets of procurement records offer potential training resources for machine learning categorization in the UNSPSC classification system. These datasets are provided by the governments of California, Australia and Canada and contain a mapped UNSPSC code with a written description of the goods and services.

One of the most common matching algorithms for strings, such as words, sentences, or sequences of characters, is Levenshtein Distance [5]. This method calculates the number of single-character edits (insertions, deletions, or substitutions) required to change one string into another.

One of the most widely used and efficient libraries for string processing is spaCy. With this library, it is possible to perform string tokenization, part-of-speech tagging, and named entity recognition. It also calculates cosine similarity, which uses word vectors to measure the similarity between text snippets.

While tackling the problem, we discovered that publicly available datasets have no correlation with item descriptions in the initial dataset in the scope of our work. Therefore, the constructed models did not produce any results. Utilizing the Levenshtein Distance could be effective for identifying minor spelling mistakes, but not for accounting for the semantic meaning of sentences. Similarly, using spaCy cosine similarity for our problem did not show results with pre-trained word embeddings.

Word embedding is a type of word representation that allows words to be represented as dense vectors in a continuous vector space. This approach captures semantic meanings and relationships between words based on their context within a large corpus of text. The most notable word embedding models include Word2Vec, GloVe (Global Vectors for Word Representation), and FastText. Word2Vec [6] uses neural networks to generate word vectors by predicting word context. GloVe, developed by Pennington et al. in 2014, focuses on aggregating global word-word co-occurrence statistics from a corpus to produce word embeddings. FastText, developed by Facebook's AI Research Lab, improves on Word2Vec by considering subword information, which enhances its ability to handle rare words and words that are not in the vocabulary used for training. These word embedding techniques have become foundational in various NLP tasks, enabling more nuanced and effective text processing, including in the domains of sentiment analysis, machine translation, and named entity recognition.

Bidirectional Encoder Representations from Transformers (BERT) is a language model developed by Google that captures context from both directions: left-to-right, and right-to-left [7]. Sentence-BERT (SBERT) is a modification of BERT that uses Siamese and triplet network structures to derive semantically meaningful sentence embedding [8]. This allowed accurate comparison of string sentences (item descriptions) for our problem.

III. Methodology

Based on our dataset of 175,113 procurement item descriptions, we aim to efficiently categorize these descriptions into specific commodity categories according to the UNSPSC categories at the commodity level. Given the large volume and diversity of the item descriptions, manual categorization is impractical and would be highly time-consuming. Therefore, this chapter presents an automated approach using natural language processing techniques to tackle the problem, providing a solution to accurately match item descriptions to the relevant UNSPSC categories.

A. Processing Dataset

For the initial dataset processing, we employed the spaCy en_core_web_md model. This model is a medium-sized English language model that includes pre-trained word vectors and deep learning capabilities for various NLP tasks.

To increase the quality of processed item descriptions, we added to the spaCy model several terms that do not usually exist in the model but are part of categorization, such as U-bolt, O-ring, and so forth. Usually, these kinds of strings are processed as 'bolt' or 'ring', but in the UNSPSC catalog 'U bolts' is a separate code 31161616 and is required to be processed apart from any other kinds of bolt. It is also required in such cases to preserve single-letter modifiers as part of hyphenated words.

We also applied abbreviation replacements for several terms, such as replacing 'sz' with 'size' or 'ss' with 'stainless steel'. This allowed us to add more terms for possible matching that, in some cases, could be meaningful. For example, the initial item description 'SS Bar' could be mapped to one of the 3026450x codes under the group '30264500 Stainless steel bars' if we transform 'SS' to 'Stainless Steel'.

Additionally, we extracted specific parts of speech that are crucial for our dataset. Specifically, we retained nouns, proper nouns, adjectives, and adpositions (prepositions) to ensure that the most relevant and descriptive terms were kept for further analysis.

B. Organizing n-Grams

The next step in data processing is generating n-grams for every row of the original dataset. An n-gram is a contiguous sequence of n items from a given text or speech sample. For the initial matching, we used sets of 2 and 3 consequent words left after the original cleaning.

C. Embedding

Due to the failure of direct n-gram comparison with both Levenshtein Distance and spaCy cosine similarity, we employed a more advanced approach using sentence embeddings. Embeddings are dense vector representations of text that capture semantic meaning and contextual relationships between words, phrases, sentences, or other textual units. This broader term encompasses various types of dense vectors used to represent textual information in a continuous vector space. Specifically, we utilized the SBERT model for both datasets.

D. Matching

For matching sentences (item descriptions and commodity names from the UNSPSC dataset, in terms of our model), we utilized the sentences provided by the Sentence Transformer library cosine similarity function. We compared every n-gram with every embedded Commodity Name field of the UNSPSC dataset and calculated the top 5 results for 3-grams and 2-grams for every matching above the threshold, which was set to 0.9.

IV. Results

In this section, we present the outcomes of our experiments and analyses. We evaluate the performance of various models and techniques for categorizing products and services using the UNSPSC classification system.

A. Initial Results

The original idea was to perform matching by utilizing publicly available datasets as training data. This failed in our case, probably due to the low correlation of our initial data with unstructured descriptions and very descriptive datasets provided by governments, meaning no correct matches were identified.

B. Implementing Matching Algorithm Based on n-grams

To enhance the matching process, we implemented a matching algorithm based on word sequences (n-grams) and applied several modifications to the original datasets, including:

1) Removing special symbols and unwanted terms and patterns
2) Preserving single-letter modifiers that are part of hyphenated words
3) Replacing abbreviations with full sentences
4) Including in the spaCy model additional terms, such as 'u-bolt'
5) Removing all words that are not nouns, proper nouns, adjectives, or adpositions (prepositions)

While initially we used the simple spaCy model, it failed to detect many of the required matches, largely because it lacked the specific words necessary for accurate matching. Consequently, we switched to the medium English spaCy model, which provided significantly better results, identifying a far greater number of matches compared to the simple model

C. Improved Matching Algorithms

After refining the dataset and the models, we explored more advanced matching algorithms. We found that matching n-grams by comparing Levenshtein Distances [5] or cosine similarities are not a robust solution for the processed dataset, as these methods prioritize matches based on string similarity or frequency rather than semantic accuracy. This approach often leads to incorrect matches or misalignments, necessitating additional steps for more precise categorization.

After implementing embedding by SBERT and calculating cosine similarities, we were able to find for 29% (51,088 of 175,113 unique rows) at least one good match (with a threshold greater than 0.9) for 3-grams or 2-grams.

D. Further Analysis and Observations

Additional analysis of the matches showed that for randomly selected matched 3-grams without matched 2-grams, the best score for the UNSPSC code is the correct match. For 3-grams with matched 2-grams, it is worth looking for the best score on 2-grams and then utilizing that. For 2-grams without a matching 3-grams score, utilizing the best result also leads to accurate prediction. However, all these matches should be verified with business users, and additional options for matches could also be provided to the users.

While performing the analysis of the matches, we also found several ways to improve our results. One interesting suggestion is to create 'fake' target commodities and perform matching of the dataset to them, while further performing manual mapping of the newly added commodity to the real one. This could be valuable for positions, such as 'Steel Nails,' that don't have actual representation in the UNSPSC dataset but would require business users' input for matching. One source of such groups could be items from others levels of the UNSPSC hierarchy, like segment, family or class. For example, for the already mentioned case of 'SS Bar', the commodity could be mapped to one of the 3026450x codes under the group '30264500 Stainless steel bars' if we transform 'SS' to 'Stainless Steel'. In that case, the current algorithm could make a match to any of the related commodity-level records, as they represent more detailed commodity description - for example, '30264503 Stainless steel SAE 400 series cold drawn bar'. As previously mentioned, the correct mapping should be to '30264500 Stainless steel bars'.

We also found that using ChatGPT for mapping could give some initial help in preparation (for example, with abbreviations); however, it also gives non-existing (hallucinating) matches. For example, for the query 'What could be the UNSPSC code for steel nails?' ChatGPT returns '31161501 Nails'. In reality, this code is for 'Cap screws,' and nails are under the class '31162000 Nails.'

E. Computational Resources and Performance

For our work, we utilized a high-performing computer (HPC), [9] as we had a very large dataset (175,113 rows) to map to a dataset with 71,502 rows. It took almost 22 hours to run on 48 CPUs with 187.5GB of memory allocated. We could expect that running on servers of an average company could lead to a dramatic drop in performance, and, additional measures probably would need to be applied.

V. Lessons Learned

In this study, we applied Natural Language Processing (NLP) techniques to categorize procurement items using the United Nations Standard Products and Services Code (UNSPSC). Throughout this process, we encountered several key challenges and learned valuable lessons that can guide future work in this area:

- Handling Unlabeled Data: A significant challenge was the absence of labeled training data, which is common in real-world scenarios. We overcame this by leveraging

NLP methods, demonstrating the importance of transforming unstructured item descriptions into a structured format for effective analysis.

- Model Selection and Data Preprocessing: Utilizing the spaCy 'en_core_web_md' model for initial data cleansing and preprocessing was essential for transforming unstructured data. However, the initial models, including those using Levenshtein Distance and simple cosine similarity, were insufficient. We then employed sentence embeddings using the SBERT model, which enabled us to capture contextual and semantic relationships between sentences for more accurate comparisons.
- Importance of Data Cleansing and Transformation: We learned that detailed data cleansing and transformation steps, such as removing special symbols, replacing abbreviations, and including domain-specific terms, significantly improved categorization
- Strategies for Enhancing Categorization: Creating 'fake' target commodities and categorizing to higher levels of hierarchy could improve the accuracy and flexibility of the categorization process by allowing the algorithm to make more informed decisions when direct matches are unavailable. This approach helps in addressing gaps where specific commodity codes are missing or too narrowly defined, thereby ensuring a broader and more adaptable categorization that aligns better with business needs.
- Challenges with Computational Resources: The study highlighted the limitations of implementing these techniques on average company servers, which may not have access to high-performance computing (HPC) resources. We recommend breaking down the data into smaller chunks, pre-processing datasets, and reducing the size of the UNSPSC dataset to make the process more manageable and efficient.
- Challenges with AI Tools: While tools like ChatGPT can assist in preprocessing tasks, we learned that they should be used with caution due to the risk of inaccuracies and hallucinations.

VI. Summary

This study demonstrated the potential of NLP and embedding techniques for categorizing procurement items without extensive labeled data. By implementing a structured approach to data preprocessing and leveraging advanced models like SBERT, we achieved an initial mapping and categorization of approximately 29% of our dataset.

Acknowledgment

The author acknowledges the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing (HPC, database, consultation) resources that have contributed to the research results reported within this paper/report.

References

[1] M. Honnibal and I. Montani, "Spacy," *Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing*, 2017.

[2] Y. Ding, M. Korotkiy, B. Omelayenko, V. Kartseva, V. Zykov, M. Klein, E. Schulten, and D. Fensel, "Goldenbullet: Automated classification of product data in e-commerce," in *Proceedings of BIS 2002*. Citeseer, 2002.

[3] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20–38, 2019.

[4] A. A. Abbott and I. Watson, "Ontology-aided product classification: a nearest neighbour approach," in *International Conference on Case-Based Reasoning*. Springer, 2011, pp. 348–362.

[5] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet Physics Doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[9] A. Reuther, C. Byun, W. Arcand, D. Bestor, B. Bergeron, M. Hubbell, M. Jones, P. Michaleas, A. Prout, A. Rosa *et al.*, "Scalable system scheduling for hpc and big data," *Journal of Parallel and Distributed Computing*, vol. 111, pp. 76–92, 2018.