

Preprocessing product descriptions with byte pair encoding: a solution for abbreviation-heavy texts

Vasyl Teslyuk[†], Anastasiya Doroshenko^{*,†} and Olga Narushynska[†]

Lviv Polytechnic National University, Stepan Bandera 12 79013 Lviv, Ukraine

Abstract

This research introduces a novel approach to short text preprocessing, particularly for the hierarchical classification of product descriptions within the Global Product Classification (GPC) system. Traditional models often falter when faced with abbreviated or incomplete product descriptions. To address this, we propose the integration of Byte Pair Encoding (BPE) with BERT embeddings to segment text into meaningful subword units, thereby improving the model's ability to capture the whole semantic meaning. Enhancing input data representation significantly improves classification performance without retraining BERT. A detailed experimental setup was carried out using a dataset consisting of product descriptions, specifically focusing on the final level of the hierarchical structure. Results demonstrate that BPE-enhanced preprocessing increases classification accuracy by 12%, particularly in classes with abbreviated terms, outperforming traditional word2vec, TF-IDF, and one-hot encoding methods. This research provides valuable insights into the efficacy of BPE as a preprocessing step, highlighting its role in optimizing classification systems dealing with short texts.

Keywords

Hierarchical classification, byte pair encoding (BPE), short text preprocessing, BERT Embeddings, product description classification, natural language processing (NLP), global product classification (GPC), abbreviation handling, subword tokenization, text data segmentation, nlp preprocessing techniques, product taxonomy classification, text embedding enhancement, classification model accuracy.

1. Introduction

In industries such as e-commerce, supply chain management, and logistics, short texts, particularly product descriptions, are a common yet challenging data form. These texts are typically ultra-concise and often composed of abbreviations, limiting the context necessary for accurate classification. Despite their brevity, accurate categorization of these descriptions is vital for organizing product information, managing inventory, and facilitating smooth operations globally. Systems like the Global Product Classification (GPC) heavily rely on precise categorization to maintain uniformity in product labeling across different regions and industries. Accurate classification ensures products are easily searchable, properly categorized, and effectively managed within inventory systems, directly impacting business operations and customer experience.

Classifying such abbreviated texts with standard Natural Language Processing (NLP) models, such as BERT embeddings, poses significant challenges [1]. BERT and other conventional models typically perform well with longer texts where rich contextual information is available [2]. However, they struggle with short, incomplete product descriptions, leading to inaccurate categorizations. These inaccuracies can manifest in various issues across business operations, including poor product retrieval, errors in inventory tracking, and miscommunication in cross-border transactions [3]. Misclassification is particularly problematic for global businesses that rely on the consistency of their

CIAW-2024: Computational Intelligence Application Workshop, October 10-12, 2024, Lviv, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ vasyli.m.teslyuk@lpnu.ua (V. Teslyuk); anaastasiia.v.doroshenko@lpnu.ua (A. Doroshenko); olha.o.narushynska@lpnu.ua (O. Narushynska)

ORCID 0000-0002-5974-9310 (V. Teslyuk); 0000-0002-7214-5108 (A. Doroshenko); 0009-0000-0628-8218 (O. Narushynska)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

product classification for managing international trade, logistics, and sales operations. The repercussions include customer dissatisfaction, operational inefficiencies, and potential financial losses due to system disruptions.

Our research addresses the critical challenge of classifying short, fragmented product descriptions. We achieve this by integrating Byte Pair Encoding (BPE) into the preprocessing pipeline for short text classification. BPE, a data compression technique, breaks down complex or abbreviated terms into smaller subword units, enabling NLP models to better interpret these descriptions. This method enhances the input representation and enables the model to capture the meaning of previously unrecognized abbreviations or rare words. By doing so, it improves the overall classification accuracy without the need for retraining the BERT model—a process that is both costly and computationally intensive.

Existing methodologies like word2vec, TF-IDF, and one-hot encoding are often insufficient to handle specific classification tasks, especially in hierarchical classification frameworks like GPC. These methods fail to account for the nuanced ways in which abbreviations or truncated text influence product categorization. By applying BPE, our model benefits from the ability to generate more accurate embeddings and efficiently classify even the most abbreviated product descriptions. This approach optimizes the model's ability to handle complex hierarchies in product classification without requiring significant computational resources.

The relevance of this research is underscored by the increasing reliance on automated systems for classifying vast amounts of short-text data across industries. In e-commerce, retail, and supply chain management, businesses must process and classify millions of product descriptions, many of which are highly abbreviated. For instance, product categories in an e-commerce setting must be accurate to allow users to find relevant products easily, enhancing the user experience and increasing sales potential. Moreover, inventory systems depend on accurate classifications to manage stock levels, avoid overstocking, and optimize product distribution across global supply chains. Improving classification efficiency directly enhances these systems, leading to greater operational accuracy, reduced costs, and improved customer satisfaction.

Furthermore, hierarchical classification systems like GPC play a crucial role in global standardization efforts. Ensuring that products are uniformly categorized across different markets enables smoother international trade and cooperation between businesses. Inaccuracies in classification can result in errors that propagate throughout global supply chains, leading to disruptions and inefficiencies. By introducing BPE-enhanced preprocessing, this research offers a scalable solution that meets the industry's growing need for precise, automated classification tools capable of handling short, abbreviated texts without sacrificing accuracy.

This study fills a critical gap in the literature on short text classification within hierarchical structures. While BPE has been applied in various NLP contexts, its potential for improving classification accuracy in systems like GPC has not been fully explored. By demonstrating how BPE can enhance the performance of hierarchical classification models, our research contributes to a deeper understanding of how preprocessing techniques can impact the accuracy and reliability of NLP models in real-world, industry-specific applications.

2. Relevant research

Text preprocessing is a crucial step in improving the accuracy of text classification tasks. Various preprocessing techniques, including bag of words, stemming, lemmatization, and keyword record counting, have been shown to enhance classification performance across different algorithms and datasets [4]. The impact of preprocessing can vary depending on the text domain, language, and feature dimensions, suggesting that carefully selecting appropriate combinations of preprocessing tasks is more effective than applying all techniques indiscriminately [5]. Feature selection methods, such as Term Frequency-Inverse Document Frequency (TF-IDF), have demonstrated superior performance compared to Bag-of-Words approaches in some cases [6]. Additionally, reducing the number of features through preprocessing and feature selection can lead to improved classification

accuracy, although the optimal number of features may depend on the specific preprocessing and feature selection techniques employed [6], [7]. Effective preprocessing strategies are essential for transforming unstructured textual data into a structured format suitable for classification tasks [8].

Recent research on short text classification using BERT has explored various approaches to enhance performance. Jrc Jayakody et al. [9] found that concatenating all BERT layers for embedding representation, combined with bagging or support vector machine algorithms, yielded the best results. Dongxue Bao et al. [10] proposed a fusion network model integrating BiLSTM, attention, and max-pooling mechanisms with BERT to capture rich semantic features. Aliyah Kurniasih & L. Manik [11] investigated the effects of text preprocessing on BERT-based models, concluding that it had an insignificant impact on most architectures, with CNN producing the best performance. Tong Zhang et al. [12] introduced a graph-based method incorporating topic information to expand the feature space for short text classification. These studies demonstrate the effectiveness of BERT-based approaches for short text classification, with various enhancements such as layer weighting, fusion networks, and graph-based methods showing promising results.

Short text classification presents challenges due to noise and data sparseness. Various preprocessing techniques have been explored to address these issues. Kumar & Harish [13] evaluated methods like removing URLs, hashtags, and stopwords, along with n-gram representation and classifiers such as SVM and KNN. Chayangkoon & Srivihok [14] proposed NDTMD, combining Bag-of-Words and word embedding for feature reduction, showing improved performance across multiple classifiers.

RaghavanA et al. [15] utilized Byte-Pair Token Encoding with SVM for multi-label classification, transforming the problem into single-label before reverting. Ma et al. [16] employed word embeddings trained on Wikipedia and a Gaussian process approach, outperforming traditional methods like MaxEnt and LDA [17]. These studies demonstrate the effectiveness of various preprocessing techniques, feature reduction methods, and distributional representations in improving short-text classification performance across different domains and classification tasks.

Byte-pair encoding (BPE) is widely used for subword tokenization in language models, offering benefits like handling out-of-vocabulary words and reducing vocabulary sparsity [18]. However, BPE can produce unexpected results, especially with ambiguous parses of words [19]. To address challenges with infrequent spellings and misspellings, Aguilar et al. [18] propose a character-based subword module (char2subword) that builds representations from characters outside the subword vocabulary, improving performance on code-switching tasks. Church [19] suggests minimizing the number of word pieces to resolve ambiguous parses. Wei et al. [20] introduce byte-level BPE (BBPE) for training multilingual pre-trained language models, demonstrating improved performance on multilingual NLU tasks compared to Google's multilingual BERT and vanilla NEZHA. These approaches aim to enhance the robustness and effectiveness of subword tokenization in various language processing tasks, particularly for multilingual and social media contexts [21].

Recent short text classification (STC) advancements have addressed challenges like feature sparsity and semantic complexity. Arumugam [22] proposed a Multivariate Relevance Frequency Analysis method, achieving high macro-F1 scores across various datasets. Wu et al. [23] introduced the Quartet Logic framework, leveraging large language models and chain-of-thought reasoning to improve STC performance. Gong et al. [24] developed a method combining explicit and implicit multiscale weighted semantic information, outperforming classical algorithms on multiple datasets. Zhu et al. [25] presented a prompt-learning approach incorporating knowledge expansion, showing significant improvements over existing methods. These studies demonstrate diverse strategies for enhancing STC, including feature selection [22], reasoning frameworks [23], semantic information fusion [24], and knowledge-based prompt-learning [25]. Each approach offers unique solutions to the persistent challenges in short text classification, showcasing the field's ongoing evolution.

3. Materials and Methods

In this research, we leverage the Global Product Classification (GPC) system, developed by GS1, to classify products. The GPC system provides a comprehensive and standardized way to categorize products, using the lowest hierarchical level, known as 'bricks.' Our dataset, derived from web scraping of product descriptions from the DirectionsForMe website [26], contains over 59,000 samples. Each product description is linked to its respective brick label, which allows us to build a flat predictive model.

In multiclass hierarchical classification, instances are classified through multiple levels, starting with broader categories and narrowing down to more specific ones. Each instance belongs to a single, most specific class at the lowest level of the hierarchy. By mentioning hierarchical classification, we highlight how our approach diverges from it. Instead of navigating the full hierarchy, we simplify the task by focusing exclusively on the lowest level, or "brick," treating each as a distinct, mutually exclusive class.

This flat classification approach allows us to develop more focused models, optimizing classification for specific product categories without the complexity of hierarchical structures. For example, the dataset includes product categories such as "Paper Towels" (304 samples), "Spirits" (506 samples), and "Cheese" (3382 samples), making it ideal for evaluating various preprocessing techniques and models (Table 1) [27].

By focusing on the flat classification of bricks, we aim to improve the model's performance in distinguishing between highly specific product categories, which is critical for real-world applications in e-commerce, supply chain management, and logistics. Using ground truth brick labels ensures we can accurately evaluate model performance using metrics such as accuracy, precision, recall, and F1 score. This flat classification approach simplifies the modeling process while delivering insights into how short, ultra-condensed product descriptions can be accurately categorized using advanced techniques like Byte Pair Encoding (BPE).

The dataset poses unique challenges due to the extensive use of abbreviations and shorthand in the product descriptions. This complexity makes it particularly challenging for natural language processing models to accurately interpret and classify the text. For instance, entries like "Idp pizza shredded reg 1x2kg" (pizza shredded regular), "Fzn chk griller 10x1000g saida" (frozen chicken griller), and "Kraft brwn salad bwl,1300ml, 300's" (Kraft brown salad bowl) demonstrate the depth of abbreviation. These texts are densely packed with shortened product names and measurements, providing models like BERT with limited contextual information to work with.

Even more challenging are entries such as "Uht nk ff 1l tp nf" (UHT non-fat milk) or "Mwc-hd-re 1 sec black 1 x 250" (microwave container). These abbreviations, often non-standardized, complicate the ability of traditional models to correctly categorize products. For instance, phrases like "Fzn chk wgs bi so 12*900g" (frozen chicken wings bone-in) or "Fr bf 900g" (fresh beef) are so compact that basic NLP preprocessing techniques like tokenization fail to break them down into meaningful components.

Table 1
Source Dataset

Segment	Family	Class	Brick	Samples count
Cleaning/ Hygiene Products	Cleaning Products	Cleaners	Paper Towels	304
	Cleaning/Hygiene Supplies	Dish Care	Hand Dish - Detergent	250
	Cleaning/Hygiene Supplies	Cleaning Aids	Household Sponges	339

Segment	Family	Class	Brick	Samples count
	Waste Management Products	Waste Products	Storage Refuse Bags	234
	Beverages	Alcoholic Beverages (Includes De-Alcoholised Variants)	Spirits	506
		Coffee/Coffee Substitutes	Coffee - Ground Beans	1472
			Coffee - Soluble Instant	345
		Non Alcoholic Beverages - Ready to Drink	Drinks Flavoured - Not Ready to Drink	750
			Drinks Flavoured - Ready to Drink	4035
			Fruit Juice Drinks - Ready to Drink (Shelf Stable)	641
			Packaged Water	939
			Stimulants/Energy Drinks - Ready to Drink	427
		Tea and Infusions/Tisanes	Tea - Bags/Loose	1402
			Tea - Liquid/Ready to Drink	917
	Bread/Bakery Products	Baking/Cooking Mixes/Supplies	Baking/Cooking Supplies (Shelf Stable)	527
		Biscuits/Cookies	Biscuits/Cookies (Perishable)	270
			Biscuits/Cookies (Shelf Stable)	3263
			Dried Breads (Shelf Stable)	211
		Bread	Bread (Shelf Stable)	933
		Sweet Bakery Products	Cakes - Sweet (Shelf Stable)	444
	Cereal/Grain/Pulse Products	Grains/Flour	Flour - Cereal/Pulse (Shelf Stable)	249
			Grains/Cereal - Not Ready to Eat - (Shelf Stable)	700
			Grains/Cereal - Ready to Eat - (Shelf Stable)	423
		Processed Cereal Products	Cereals Products - Not Ready to Eat (Shelf Stable)	241
			Cereals Products - Ready to Eat (Shelf Stable)	1847
	Confectionery/Sugar Sweetening Products	Confectionery Products	Chocolate and Chocolate/Sugar Candy Combinations - Confectionery	2972
			Sugar Candy/Sugar Candy Substitutes Confectionery	2501
		Sugars/Sugar Substitute Products	Sugar/Sugar Substitutes (Shelf Stable)	357

Segment	Family	Class	Brick	Samples count		
Food/Be- verage/To bacco	Fruits/Vegetables/Nu ts/Seeds Prepared/Processed	Fruit Prepared/Processed	Syrup/Treacle/Molasses (Shelf Stable)	258		
			Fruit - Prepared/Processed (Frozen)	199		
		Vegetables Prepared/Processed	Fruit - Prepared/Processed (Shelf Stable)	1522		
			Vegetables Prepared/Processed (Frozen)	- 772		
			Vegetables Prepared/Processed (Perishable)	- 774		
			Vegetables Prepared/Processed (Shelf Stable)	- 2420		
			Meat/Poultry/Other Animals Prepared/Processed	Meat/Poultry/Othe r Animals - Prepared/Processed	Chicken Prepared/Processed	- 1747
				Turkey Prepared/Processed	- 674	
				Meat/Poultry/Othe r Animals - Unprepared/Unpro cessed	Pork Unprepared/Unprocessed	- 389
			Milk/Butter/Cream/ Yogurts/Cheese/Egg s/Substitutes	Meat/Poultry/Othe r Animals Sausages - Prepared/Processed	Mixed Species Sausages Prepared/Processed	- 307
	Butter/Butter Substitutes	Butter (Perishable)			316	
	Cheese/Cheese Substitutes	Cheese (Perishable)		3382		
	Cream/Cream Substitutes	Cream (Perishable)		525		
	Milk/Milk Substitutes	Cream (Shelf Stable)		158		
		Milk (Shelf Stable)		317		
		Milk Substitutes (Perishable)		877		
		Yogurt/Yogurt Substitutes		Yogurt (Perishable)	2727	
	Prepared/Preserved Foods	Desserts/Dessert Sauces/Toppings		Oils Edible	Oils Edible - Vegetable or Plant (Shelf Stable)	718
				Dessert Sauces/Toppings/Fillings (Shelf Stable)	409	
			Desserts (Frozen)	399		
Desserts (Perishable)			232			

Segment	Family	Class	Brick	Samples count
			Ice Cream/Ice Novelties (Frozen)	2825
		Sandwiches/Filled Rolls/Wraps	Sandwiches/Filled Rolls/Wraps (Frozen)	1061
			Sandwiches/Filled Rolls/Wraps (Perishable)	236
		Snacks	Snacks Other	218
		Sweet Spreads	Confectionery Based Spreads (Shelf Stable)	167
			Jams/Marmalades (Shelf Stable)	383
		Vegetable Based Products / Meals	Vegetable Based Products / Meals - Ready to Eat (Perishable)	179
			Vegetable Based Products / Meals - Ready to Eat (Shelf Stable)	215
	Seasonings/Preservatives/Extracts	Herbs/Spices/Extracts	Extracts/Salt/Meat Tenderisers (Shelf Stable)	237
			Extracts/Seasonings/Flavour Enhancers (Shelf Stable)	209
			Herbs/Spices (Shelf Stable)	2165
			Stock/Bones (Shelf Stable)	173
		Sauces/Spreads/Dips/Condiments	Dressings/Dips (Perishable)	451
			Dressings/Dips (Shelf Stable)	1436
			Sauces - Cooking (Shelf Stable)	1653
Kitchenware and Tableware	Kitchenware	Cookware/Bakeware	Bakeware/Ovenware/Grill ware (Non Disposable)	395
		Kitchen Storage	Disposable Food Bags	220

Other entries, such as "Mb_igloo_4x4ltr_tub_vanilla" (vanilla ice cream) and "S.s.u.m rl 135m 2ply eco 6x1" (toilet paper roll), introduce even more complexity by mixing abbreviations and domain-specific jargon, further reducing the effectiveness of models in distinguishing products across categories.

These examples brightly illustrate the challenges posed by the dataset's format, where short, highly compressed product descriptions often leave out critical semantic information, making classification tasks extremely difficult (Table 2) [28]. Standard preprocessing techniques, like tokenization, are inadequate to handle such complex abbreviations. This underscores the necessity of advanced methods like Byte Pair Encoding (BPE) for segmenting words into meaningful subword units [29]. By dissecting these abbreviations, BPE facilitates a more accurate interpretation of the text, thereby enhancing the performance of classification models for such demanding datasets.

Table 2

Product description in dataset

Category	Abbreviated Description	Full Description
Salad Bowl	Kraft brwn s bwl,1300ml, 300's	Kraft brown salad bowl 1300 ml, 300 units
Cream Whipping	E&v-uht cr whipping 35.1% fat-12x1ltr	E&V UHT cream whipping 35.1% fat, 12 x 1 liter
Frozen Shrimp	F z shrmp pd 16/20 1kg x10 35%	Frozen shrimp peeled deveined 16/20, 1 kg x 10, 35% water
Milk Substitutes	Uht nk ff 1l tp nf	UHT milk full fat, 1 liter, top notch
Chicken Wings	Fzn chk wgs bi so 12*900g	Frozen chicken wings bone-in skin-on, 12 x 900 g
Ice Cream	Mb_igloo_4x4ltr_tub_vanilla	Ice cream tub vanilla, 4 x 4 liters
Toilet Paper	S.s.u.m rl 135m 2ply eco 6x1 *tm52em13001r13	Toilet paper roll 135 m, 2- ply eco, 6 x 1
Cling Film	St-cling film 45 cm x 1.50kg	Stretch cling film 45 cm x 1.50 kg
Desserts (Frozen)	Fzn-van-crm-pie-10x1kg	Frozen vanilla cream pie, 10 x 1 kg
Cream (Shelf Stable)	UHT-crm-dlb-whp-1L	UHT cream double whipped, 1 liter
Milk (Perishable)	Substitutes Uht nk ff 1l tp nf	UHT milk full fat, 1 liter, top notch
Disposable Containers	Food Mwc-hd-re 1 sec black 1 x 250	Microwave container heavy duty rectangular 1 section black, 1 x 250 units
Disposable Containers	Food Microwave cont 1000cc +lid (1x500pcs)	Microwave container 1000 cc with lid, 1 x 500 pieces
Beef Unprepared/Unprocessed	- Fr bf 900g	Fresh beef, 900 g
Chicken Prepared/Processed	- Fzn chk wgs bi so 12x900g	Frozen chicken wings bone-in skin-on, 12 x 900 g
Disposable Containers	Food Mwc-hd-ro 24 oz black nt 1 x 150	Microwave container heavy duty round 24 oz black no top, 1 x 150 units
Pizza Shredded	Idp pizza shredded reg 1x2kg	Pizza shredded regular, 1 x 2 kg
Salad Bowl	Kraft brwn salad bwl,1300ml, 300's	Kraft brown salad bowl, 1300 ml, 300 units
Tuna Loin	Tuna loin yf mldvs 3/4kg (1x1)	Tuna loin yellowfin, Maldives, 3/4 kg, 1 x 1 unit

The field of text preprocessing for natural language processing (NLP) has evolved significantly over the years, with various approaches developed to optimize the representation of textual data for machine learning models. Traditional methods, such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF), have long been used for text classification. These techniques represent texts as numerical vectors based on word occurrences, making them computationally

simple and efficient. However, these methods have several limitations, particularly in capturing the semantic meaning of words and phrases. Since they do not consider word order or context, they fail to differentiate between homonyms and polysemy, which are critical for accurate understanding in classification tasks.

To address these limitations, more advanced techniques like word embeddings were introduced. Word2Vec, introduced by Mikolov et al. [30], revolutionized text representation by embedding words in a continuous vector space where semantically similar words are located closer together. This allows models to capture relationships between words and provides better context than BoW and TF-IDF. However, Word2Vec relies on a fixed vocabulary and struggles with out-of-vocabulary (OOV) words, especially when dealing with short texts or domain-specific terminology. GloVe, another word embedding method, improved upon Word2Vec by capturing global word co-occurrence statistics. However, it shares similar limitations regarding OOV words and performs poorly with highly abbreviated or sparse data.

The advent of contextual embeddings, particularly with the introduction of models like BERT (Bidirectional Encoder Representations from Transformers), marked a significant leap in NLP. BERT captures the context of words in a sentence by considering the entire input sequence, both forward and backward, providing much richer and more nuanced word representations. This has proven highly effective in various NLP tasks, including classification. BERT embeddings are particularly powerful for longer texts, where context is crucial in understanding meaning. However, BERT's performance tends to decline when applied to short texts like product descriptions with frequent abbreviations. These models require rich contextual information, often absent in ultra-short texts, leading to reduced accuracy in classification tasks.

Byte Pair Encoding (BPE) has emerged as a solution to some of these challenges. Developed initially as a data compression technique, BPE is now widely used in NLP to address issues related to rare or OOV words. By segmenting words into smaller subword units, BPE effectively reduces the vocabulary size and enables models to process previously unseen words by breaking them into recognizable components. This is particularly useful for classifying short texts where abbreviations and domain-specific terms are typical. BPE has been integrated into transformer-based models like BERT and GPT, which improves the handling of rare or complex words. However, it still struggles with specific semantic nuances that purely contextual models can better capture.

While BPE enhances text preprocessing, it does not solve all issues. It requires careful balancing between segmentation granularity and model performance, as overly fine segmentation can lead to loss of semantic meaning. Additionally, BPE alone may not fully resolve issues of polysemy or context, as it focuses primarily on word structure rather than meaning. Combining BPE with powerful contextual models like BERT can provide a more holistic approach, leveraging the strengths of both techniques. BPE handles rare and abbreviated words effectively. BERT's contextual capabilities capture the overall meaning, making this combination highly suitable for short text classification tasks, such as product descriptions.

While traditional methods like BoW and TF-IDF are efficient, they lack the ability to capture semantic relationships. Word embeddings such as Word2Vec and GloVe offer improvements but face challenges with OOV words. BERT provides rich contextual understanding but struggles with short texts. BPE, particularly when combined with models like BERT, provides a promising approach to addressing these issues, especially in domains where short, highly abbreviated texts are prevalent. This combination represents the current state-of-the-art in text preprocessing for hierarchical classification tasks.

Retraining BERT on a highly abbreviated and domain-specific dataset such as ours presents several practical and technical challenges. First and foremost, BERT is a model pre-trained on large, general-purpose corpora like Wikipedia and BookCorpus. These corpora contain full-length, contextually rich sentences, which BERT uses to learn nuanced relationships between words. Our dataset, on the other hand, consists of highly abbreviated, context-sparse product descriptions, often lacking grammatical structure or sufficient contextual information.

1. Lack of Training Data Volume

BERT requires a vast amount of diverse data to be fine-tuned effectively. While our dataset contains thousands of samples, the overall data size and diversity are insufficient to retrain BERT properly. Fine-tuning BERT for domain-specific tasks typically requires a significantly large corpus that captures the specific terminology and provides enough variability for the model to learn effectively. Our dataset, being limited to specific product categories and highly abbreviated descriptions, does not provide the wide variety necessary for retraining a language model of BERT's scale.

2. Computational Cost

Retraining BERT from scratch or even performing domain-specific fine-tuning demands significant computational resources in terms of hardware (GPUs or TPUs) and time. Training BERT involves processing a large number of layers and parameters, making it computationally expensive. For our dataset, the benefits of retraining BERT do not justify the substantial costs in resources and time, especially given the relatively small size of our dataset. Instead, leveraging BERT's pre-trained knowledge through techniques like Byte Pair Encoding (BPE) is a more efficient way to adapt the model to our task without the full retraining overhead.

3. Poor Alignment with BERT's Pre-training

BERT is pre-trained on coherent, well-formed sentences, where word order and sentence structure play a critical role in understanding context. Our dataset's highly abbreviated, fragmented product descriptions do not align well with BERT's pre-training. Terms like "Fr bf 900g" (Fresh beef 900g) or "Fzn chk griller" (Frozen chicken griller) lack the syntactic structure that BERT relies on to understand the context. Retraining BERT on such data would result in a model that struggles to maintain its original language understanding capabilities, as it would be forced to learn from fragmented, domain-specific inputs.

4. Loss of Generalization

Retraining BERT on our dataset might improve performance on product descriptions, but it would likely reduce the model's ability to generalize to other tasks. BERT's strength lies in its generalization across various NLP tasks due to its pre-training on broad datasets. By retraining it on particular, abbreviated product descriptions, we risk overfitting the model to this narrow domain. This could hinder its performance on broader text classification tasks involving more diverse and complex language.

5. Efficient Alternatives

Rather than retraining BERT, incorporating Byte Pair Encoding (BPE) during the preprocessing stage is a more efficient way to adapt the model to our dataset. BPE allows us to break down abbreviated words into subword units, making the input more digestible for BERT's pre-trained layers. This approach maintains the advantages of BERT's pre-trained contextual understanding while addressing the specific challenges of our abbreviated dataset.

Thus, retraining BERT on our dataset is not feasible due to the limited size and context of the data, the high computational cost, and the risk of losing generalization capabilities. Instead, enhancing the preprocessing step with techniques like BPE provides a more practical solution, enabling BERT to handle abbreviated product descriptions without sacrificing the efficiency or versatility of the model.

4. Results and discussion

The experimental setup is meticulously designed to evaluate a range of text preprocessing techniques and machine learning models on a dataset of product descriptions categorized using Global Product Classification (GPC) labels. The primary objective is to compare the performance of Byte Pair Encoding (BPE) combined with BERT embeddings against traditional methods like word2vec, TF-IDF, and one-hot encoding.

Product descriptions are preprocessed into various formats in the dataset preparation stage, including BPE-enhanced BERT embeddings, word2vec vectors, and TF-IDF representations. Each dataset is then paired with its corresponding GPC "brick" labels, which serve as the target for

classification. The product descriptions, encoded initially as strings, are transformed into a list of numerical vectors using custom preprocessing functions (`string_to_list`), making them suitable for machine learning models.

A series of machine learning models are evaluated during this experiment, including XGBoost, Logistic Regression, Random Forest, and Decision Trees. Each model is optimized with hyperparameters that control aspects such as tree depth, learning rate, and the number of estimators. The XGBoost model, for instance, is configured with a max depth of 8 and a learning rate of 0.15, making it well-suited to handle complex classification tasks.

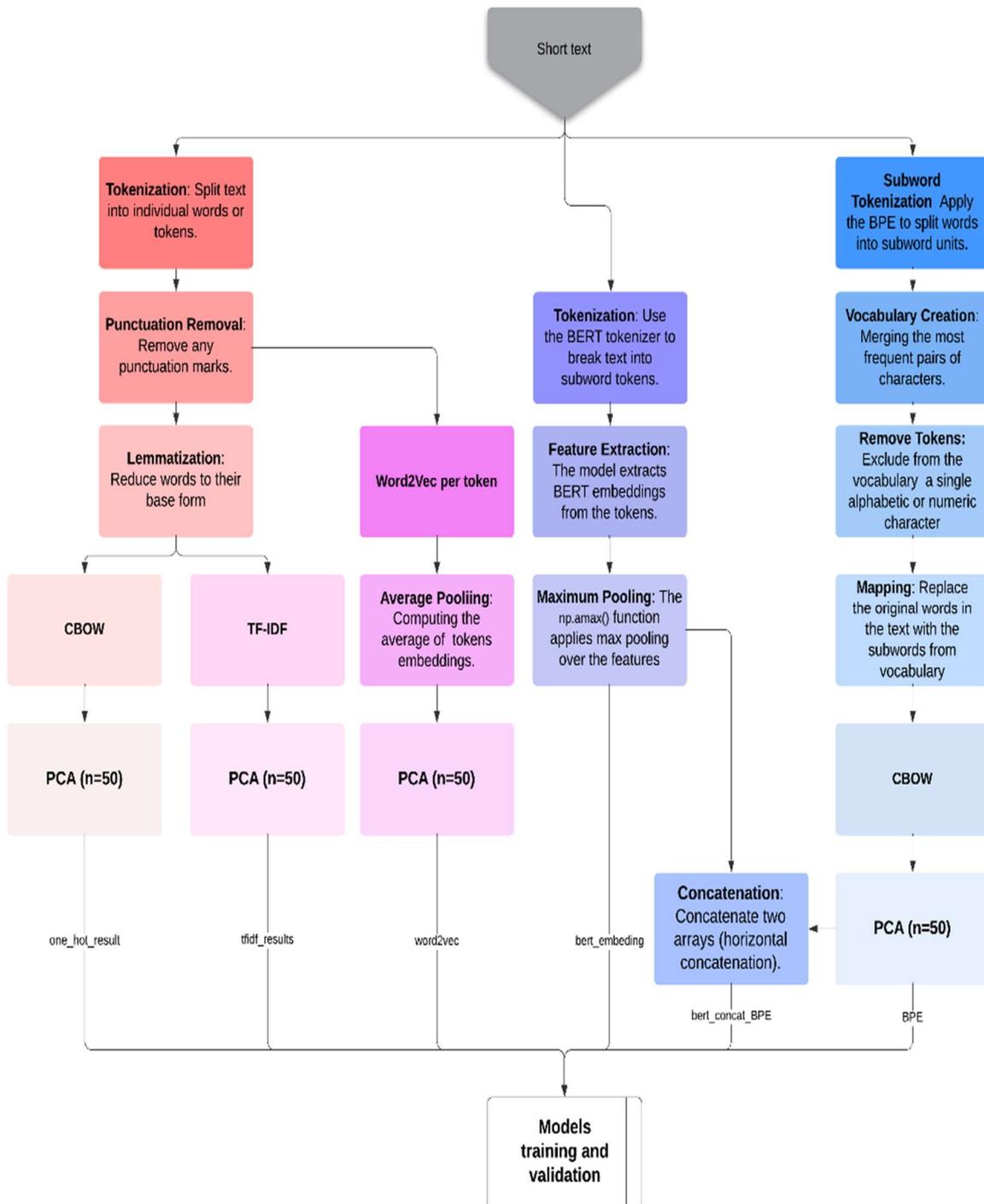


Figure 1: Experiments pipelines.

The experimental pipeline follows a structured flow. First, product descriptions are processed into different feature sets according to the evaluated preprocessing technique (Figure 1). For example, BPE-processed text is converted into embeddings, while other datasets like one-hot encoding and TF-IDF are prepared as sparse matrices. The dataset is then split into training and testing subsets with an 80/20 split, ensuring the test set contains only those "brick" labels present in the training set to prevent unseen categories from negatively impacting performance.

The data is standardized using StandardScaler to ensure uniform scaling across all features [31]. This helps mitigate potential biases due to varying feature scales and ensures that machine learning algorithms, especially tree-based models, perform optimally.

For monitoring and tracking purposes, the experiment utilizes Weights and Biases (wandb), a powerful tool for experiment tracking and visualization [32]. During each run, the models are trained, and their performance on both the training and test sets is meticulously logged. After the training phase, predictions are made on both sets, and various evaluation metrics—such as precision, recall, F1 score, and accuracy—are calculated for each hierarchical level of the GPC taxonomy (Segment, Family, Class, and Brick).

An essential aspect of the evaluation is how well models can generalize from training to unseen test data. To assess this, learning curves are generated for each model using wandb's built-in visualization tools (Figure 2, Figure 3). These curves help determine whether the models are overfitting to the training data or if they generalize well across different datasets.

These visualizations reinforce the effectiveness of BPE in improving the model's ability to generalize from training to unseen test data. Specifically, integrating BPE into the preprocessing pipeline boosts the performance of complex models like XGBoost and Random Forest, leading to more accurate classification in the short-text product description classification task.

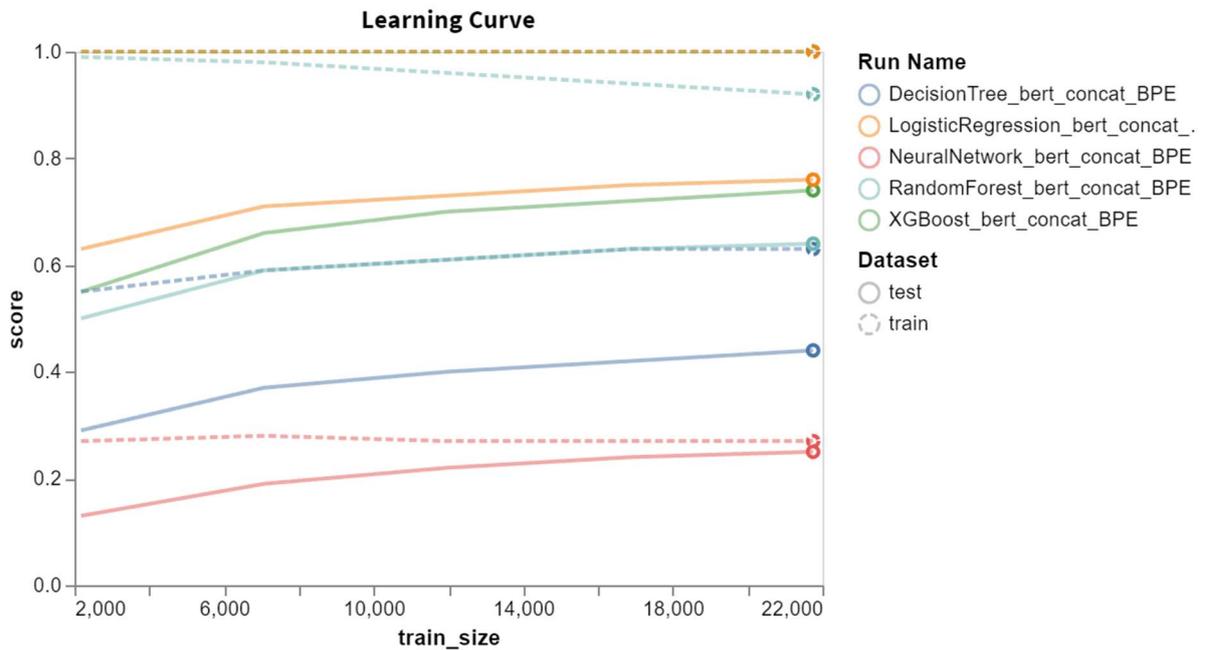


Figure 2: Learning Curve: BERT Concatenated with BPE.

In summary, this experimental setup involves preprocessing product descriptions into different vector formats, training multiple models, and evaluating their performance across the GPC hierarchy. The inclusion of Byte Pair Encoding (BPE) as a preprocessing step aims to handle highly abbreviated product descriptions more effectively, with the expectation that BPE-enhanced BERT embeddings will provide a performance boost over traditional methods like word2vec and TF-IDF [33],[34]. Performance metrics, including precision, recall, and F1 score, are logged and analyzed at multiple levels of the GPC hierarchy. This comprehensive analysis provides a detailed view of model effectiveness.

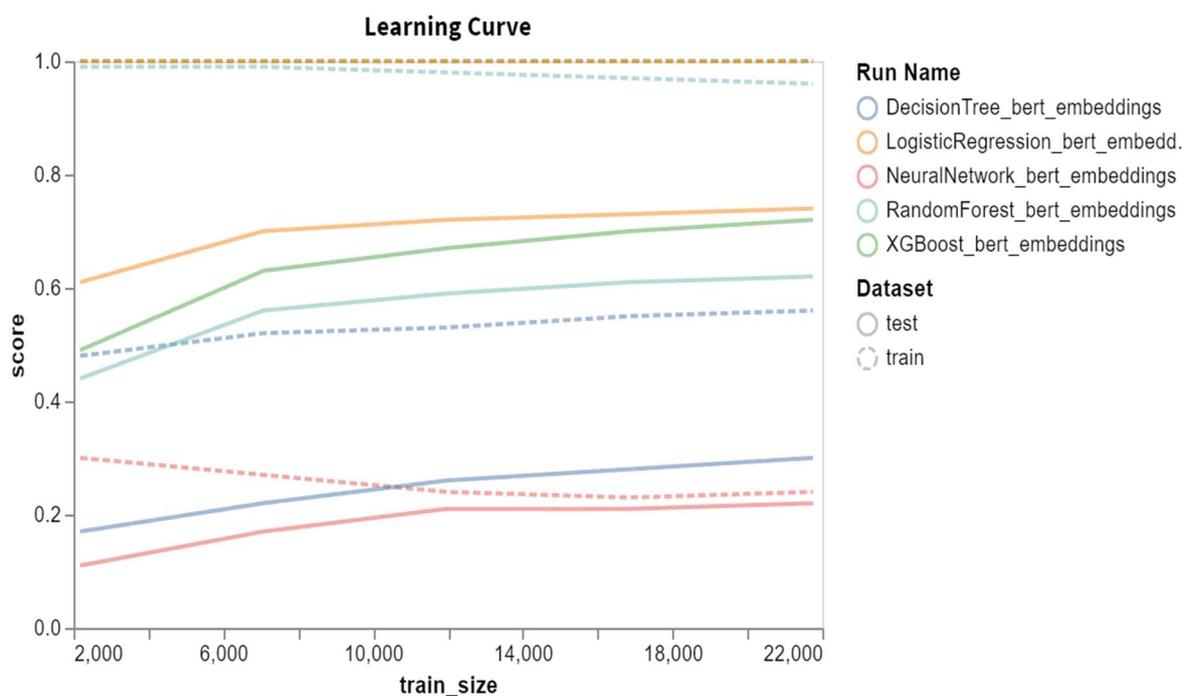


Figure 3: Learning Curve: BERT embeddings.

Embedding-based methods like bert_concat_BPE and models like XGBoost and Random Forest provide the best performance for classifying short product descriptions. On the contrary, traditional feature extraction methods like tfidf_result and one_hot_result yield poor results, demonstrating their limitations for this task.

Table 3

Comparing results for testing of different preprocessing approaches

Dataset	Model	Metrics for test set			
		Precision	Recall	F1 Score	Accuracy
BPE	LogisticRegression	0,53	0,53	0,52	0,53
bert_concat_BPE	LogisticRegression	0,76	0,76	0,76	0,76
bert_embeddings	LogisticRegression	0,74	0,74	0,74	0,74
word2vec	LogisticRegression	0,61	0,63	0,62	0,63
one_hot_result	LogisticRegression	0,00	0,02	0,00	0,02
tfidf_result	LogisticRegression	0,00	0,02	0,00	0,02
BPE	RandomForest	0,61	0,48	0,46	0,48
bert_concat_BPE	RandomForest	0,69	0,67	0,63	0,66
bert_embeddings	RandomForest	0,67	0,63	0,60	0,63
word2vec	RandomForest	0,66	0,65	0,62	0,65
one_hot_result	RandomForest	0,00	0,02	0,00	0,02
tfidf_result	RandomForest	0,00	0,02	0,00	0,02
BPE	DecisionTree	0,32	0,31	0,31	0,31
bert_concat_BPE	DecisionTree	0,46	0,45	0,45	0,45
bert_embeddings	DecisionTree	0,47	0,44	0,45	0,44
word2vec	DecisionTree	0,41	0,41	0,41	0,41
one_hot_result	DecisionTree	0,00	0,02	0,00	0,02
tfidf_result	DecisionTree	0,00	0,02	0,00	0,02
BPE	NeuralNetwork	0,16	0,22	0,16	0,22
bert_concat_BPE	NeuralNetwork	0,22	0,29	0,24	0,29

bert_embeddings	NeuralNetwork	0,15	0,22	0,17	0,22
word2vec	NeuralNetwork	0,18	0,25	0,20	0,25
one_hot_result	NeuralNetwork	0,00	0,02	0,00	0,02
tfidf_result	NeuralNetwork	0,00	0,02	0,00	0,02
BPE	XGBoost	0,64	0,64	0,63	0,64
bert_concat_BPE	XGBoost	0,75	0,75	0,74	0,75
bert_embeddings	XGBoost	0,73	0,73	0,72	0,73
word2vec	XGBoost	0,68	0,68	0,67	0,68
one_hot_result	XGBoost	0,22	0,27	0,22	0,27
tfidf_result	XGBoost	0,38	0,37	0,37	0,37

Generally, the highest-performing methods use BERT embeddings or a combination of BERT and BPE. The bert_concat_BPE dataset consistently delivers the best results across multiple models, from Logistic Regression (F1 score of 0.76) to XGBoost (F1 score of 0.75), highlighting the robustness of combining byte pair encoding (BPE) with embeddings (Table 3). This suggests that more complex, context-aware embeddings are essential for effectively handling short, abbreviation-heavy text classifications (Table 4).

This comprehensive evaluation highlights the importance of selecting the proper preprocessing techniques (like BERT and BPE embeddings) and using models that can generalize well to unseen data (like XGBoost and Random Forest) when dealing with complex, abbreviation-heavy product descriptions.

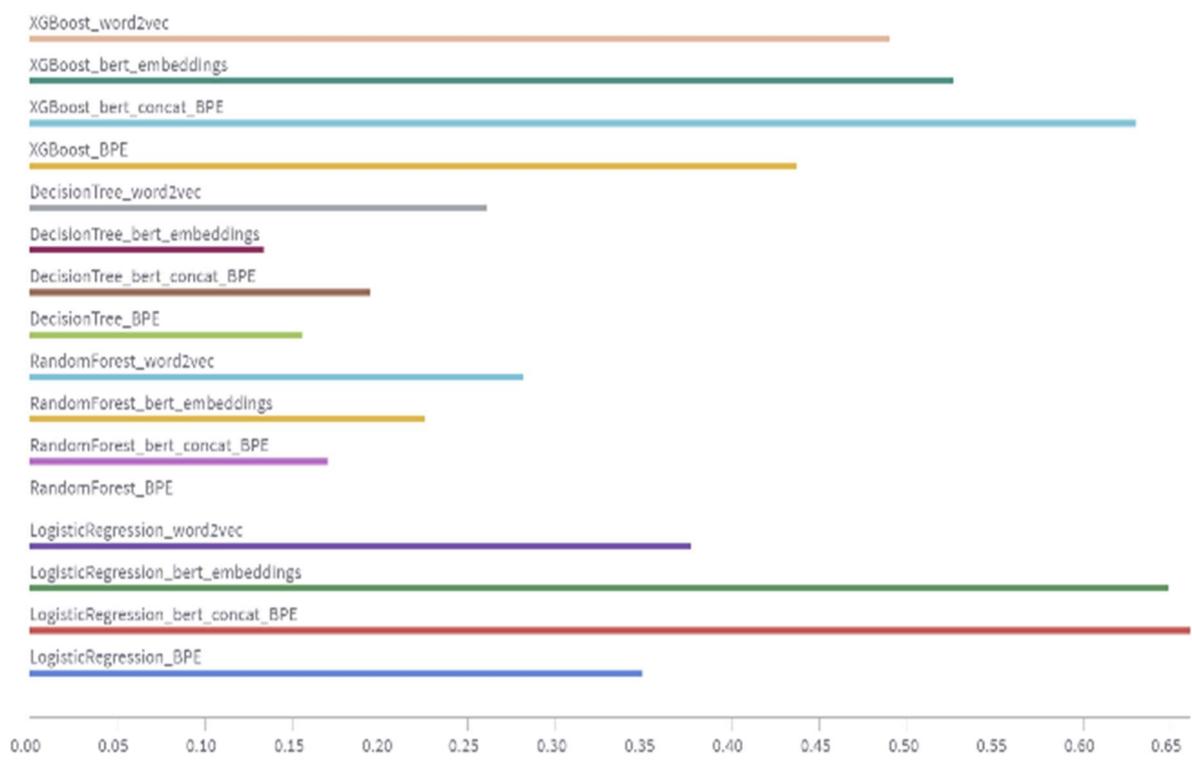
Table 4

Comparing results for training of different preprocessing approaches

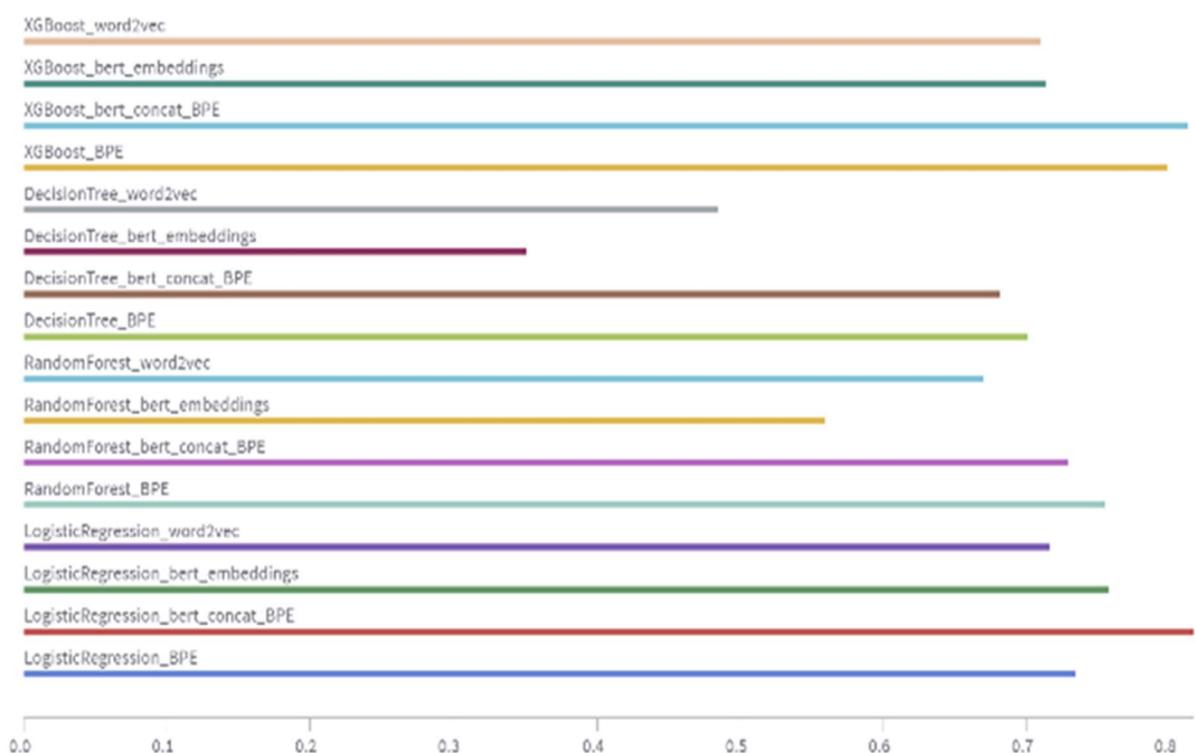
Dataset	Model	Metrics for train set			
		Precision	Recall	F1 Score	Accuracy
BPE	LogisticRegression	0,55	0,54	0,53	0,54
bert_concat_BPE	LogisticRegression	0,89	0,89	0,89	0,89
bert_embeddings	LogisticRegression	0,87	0,87	0,87	0,87
word2vec	LogisticRegression	0,63	0,64	0,63	0,64
one_hot_result	LogisticRegression	0,00	0,02	0,00	0,02
tfidf_result	LogisticRegression	0,00	0,02	0,00	0,02
BPE	RandomForest	0,73	0,56	0,55	0,56
bert_concat_BPE	RandomForest	0,95	0,95	0,95	0,95
bert_embeddings	RandomForest	0,92	0,91	0,90	0,91
word2vec	RandomForest	0,90	0,89	0,89	0,89
one_hot_result	RandomForest	0,00	0,02	0,00	0,02
tfidf_result	RandomForest	0,00	0,02	0,00	0,02
BPE	DecisionTree	0,58	0,57	0,57	0,57
bert_concat_BPE	DecisionTree	0,64	0,64	0,63	0,64
bert_embeddings	DecisionTree	0,67	0,63	0,63	0,63
word2vec	DecisionTree	0,61	0,61	0,60	0,61
one_hot_result	DecisionTree	0,00	0,02	0,00	0,02
tfidf_result	DecisionTree	0,00	0,02	0,00	0,02
BPE	NeuralNetwork	0,18	0,24	0,18	0,24
bert_concat_BPE	NeuralNetwork	0,22	0,29	0,24	0,29
bert_embeddings	NeuralNetwork	0,16	0,23	0,18	0,23
word2vec	NeuralNetwork	0,18	0,26	0,20	0,26
one_hot_result	NeuralNetwork	0,00	0,02	0,00	0,02
tfidf_result	NeuralNetwork	0,00	0,02	0,00	0,02

BPE	XGBoost	0,82	0,82	0,82	0,82
bert_concat_BPE	XGBoost	0,87	0,94	0,90	0,94
bert_embeddings	XGBoost	0,99	0,99	0,99	0,99
word2vec	XGBoost	0,80	0,80	0,80	0,80
one_hot_result	XGBoost	0,46	0,46	0,46	0,46
tfidf_result	XGBoost	0,60	0,58	0,59	0,58

report.Brick_Test_classification_report.Biscuits/Cookies (Perishable).f1-score



report.Brick_Test_classification_report.Chicken - Prepared/Processed.f1-score



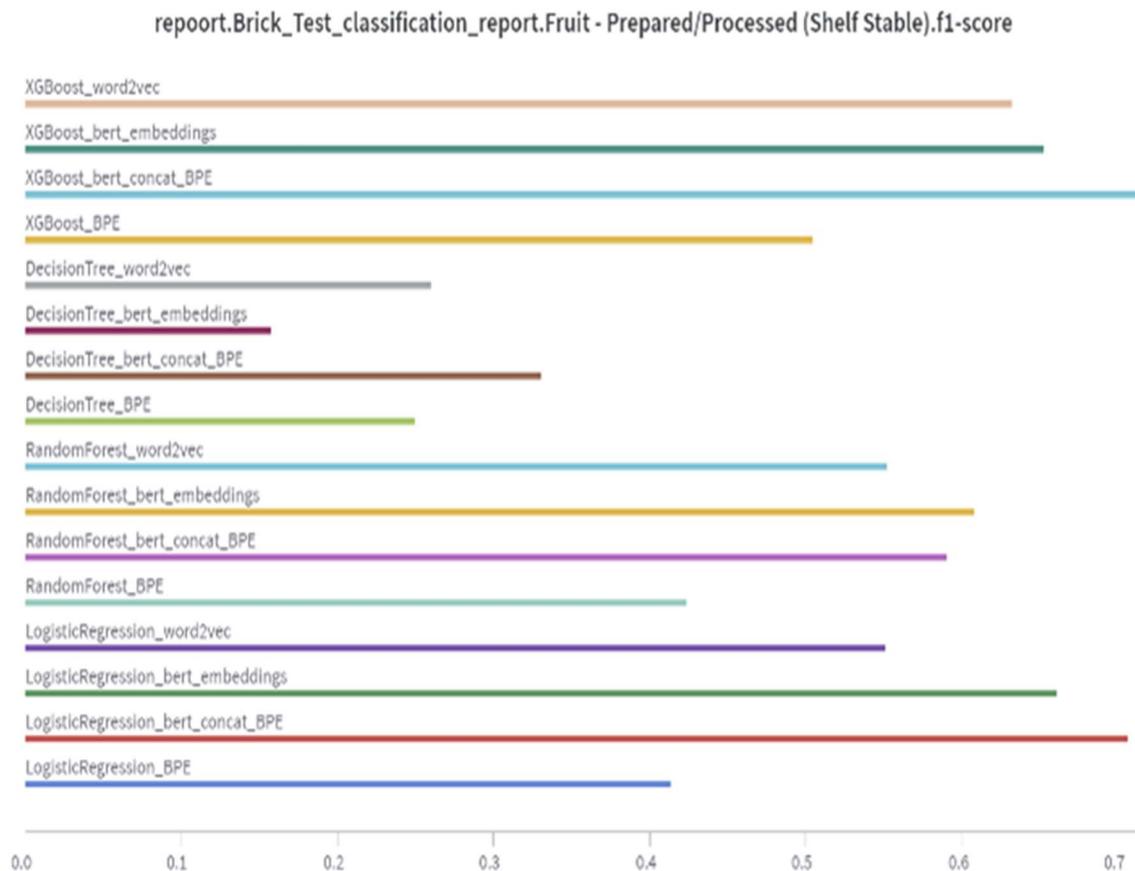


Figure 4: Comparison of F1 Scores for Different Preprocessing Techniques and Models across Various Bricks (Biscuits/Cookies, Chicken - Prepared/Processed, Fruit - Prepared/Processed, Extracts/Seasonings/Flavour Enhancers).

Some categories, like "Extracts/Seasonings/Flavour Enhancers (Shelf Stable)," include various products with vague or ambiguous descriptors. This introduces complexity that can confuse both simple and advanced models. A deeper analysis here helps to reveal whether context-rich embeddings (e.g., bert_concat_BPE) offer a significant advantage in making distinctions among subtly different product descriptions.

By diving deeper into these bricks, we can see how models handle more common, well-defined categories and more niche, diverse categories. This analysis is crucial because real-world product classification systems must handle this full range of products.

By focusing on these bricks, which serve as a practical test case, we can evaluate how well models perform when faced with abbreviated descriptions (Figure 4). This practicality makes the evaluation more relatable and applicable. These bricks are a practical test case for how well Byte Pair Encoding (BPE) enhances the capability of models to understand and classify such difficult inputs.

The F1 scores across the various bricks clearly show that preprocessing significantly determines model performance, particularly in handling abbreviated and short text descriptions. For instance, when we compare the models using traditional embeddings like word2vec or one-hot encodings, it becomes evident that they consistently underperform compared to models using more sophisticated preprocessing techniques like BPE (Byte Pair Encoding) combined with BERT embeddings. This shows that BPE's ability to break down complex, abbreviated product descriptions into subwords significantly improves the models' ability to classify correctly, especially in categories with non-standard abbreviations or particular terminology. This highlights the limitations of traditional approaches in handling datasets where abbreviations, short text, and incomplete information are prevalent. In contrast, models using BPE consistently outperform, as BPE can capture subword structures, improving semantic understanding.

5. Conclusions

This study focused on advancing short text preprocessing for product classification within the Global Product Classification (GPC) system, specifically tackling the unique challenges posed by highly abbreviated and context-limited product descriptions. By introducing Byte Pair Encoding (BPE) as a preprocessing step in combination with BERT embeddings, our research demonstrated significant improvements in classification performance compared to traditional methods like word2vec, TF-IDF, and one-hot encoding [35].

One of the most notable findings was the capacity of BPE to effectively handle abbreviation-heavy texts by segmenting them into subword units, allowing models better to interpret the meaning of highly condensed product descriptions. This enhancement, however, not only elevates classification accuracy but also bypasses the need for expensive and time-consuming retraining of the BERT model [36]. Instead of retraining BERT on a domain-specific and abbreviated dataset, which is computationally intensive and risks reducing the model's generalization capabilities, we leveraged BPE to adapt the input text to BERT's pre-trained architecture. This approach preserved the model's generalization ability while improving its handling of domain-specific terminology.

While traditional methods such as word2vec and one-hot encoding are frequently used in similar classification tasks, our results revealed their limitations in dealing with abbreviated texts [37]. These methods often fail to capture the semantic meaning of truncated terms, leading to poor classification results, especially for categories with complex product names [38]. On the other hand, BPE's capacity to decompose abbreviations into recognizable subwords allowed the BERT embeddings to capture the subtle semantic nuances, thus leading to superior performance.

Interestingly, the research also provided insight into the limitations of retraining BERT directly on this dataset. Due to the highly domain-specific nature and extreme abbreviation of the product descriptions, retraining BERT would require vast computational resources and likely reduce the model's ability to generalize across other tasks. By integrating BPE into the preprocessing stage, we addressed these issues without the risk of overfitting the model to a narrow domain, thus preserving the robustness of BERT's architecture while enhancing its adaptability to our dataset.

The experimental results consistently favored BPE-augmented embeddings across models, with XGBoost and Random Forest achieving the highest performance, particularly regarding F1 score and accuracy. This suggests a balance between preprocessing sophistication and model complexity is crucial when dealing with short-text classification tasks. When paired with models capable of capturing deep hierarchical relationships like XGBoost, BPE proved to be an optimal solution for addressing the challenges posed by short, context-poor text descriptions.

In conclusion, this study highlights the potential of BPE as a powerful preprocessing technique that enhances model performance without requiring retraining of complex architectures like BERT. This approach offers a scalable, resource-efficient solution for hierarchical classification tasks in industries such as e-commerce and supply chain management by enabling more accurate classification of abbreviated product descriptions. Future research may focus on refining the integration of BPE with other language models or exploring its applicability in domains where short, condensed text is prevalent. Additionally, examining more domain-specific fine-tuning strategies for BERT in conjunction with advanced preprocessing methods like BPE could yield further improvements in model accuracy and robustness across diverse classification tasks.

References

- [1] M. Omar, S. Choi, D. Nyang, and D. Mohaisen, "Robust Natural Language Processing: Recent Advances, Challenges, and Future Directions," *IEEE Access*, vol. 10, pp. 86038–86056, 2022, doi: 10.1109/ACCESS.2022.3197769.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," presented at the NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics:

- Human Language Technologies - Proceedings of the Conference, 2019, pp. 4171–4186. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083815650&partnerID=40&md5=4986c6d6076c0c91df84d17216b47216>
- [3] Tsmots I, Teslyuk V, Łukaszewicz A, Lukashchuk Y, Kazymyra I, Holovatyy A, Opotyak Y. An Approach to the Implementation of a Neural Network for Cryptographic Protection of Data Transmission at UAV. *Drones*. 2023; 7(8):507. <https://doi.org/10.3390/drones7080507>
 - [4] M. Mali and M. Atique, “The Relevance of Preprocessing in Text Classification,” in *Proceedings of Integrated Intelligence Enable Networks and Computing*, K. K. Singh Mer, V. B. Semwal, V. Bijalwan, and R. G. Crespo, Eds., in *Algorithms for Intelligent Systems*. , Singapore: Springer Singapore, 2021, pp. 553–559. doi: 10.1007/978-981-33-6307-6_55.
 - [5] A. K. Uysal and S. Gunal, “The impact of preprocessing on text classification,” *Information Processing & Management*, vol. 50, no. 1, pp. 104–112, Jan. 2014, doi: 10.1016/j.ipm.2013.08.006.
 - [6] N. S. M. Nafis and S. Awang, “The Impact of Pre-processing and Feature Selection on Text Classification,” in *Advances in Electronics Engineering*, vol. 619, Z. Zakaria and R. Ahmad, Eds., in *Lecture Notes in Electrical Engineering*, vol. 619. , Singapore: Springer Singapore, 2020, pp. 269–280. doi: 10.1007/978-981-15-1289-6_25.
 - [7] D. Savchuk and A. Doroshenko, “Investigation of machine learning classification methods effectiveness,” in *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, LVIV, Ukraine: IEEE, Sep. 2021, pp. 33–37. doi: 10.1109/CSIT52700.2021.9648582.
 - [8] A. Doroshenko, “Application of Global Optimization Methods to Increase the Accuracy of Classification in the Data Mining Tasks,” *Computer Modeling and Intelligent Systems*, vol. 2353, pp. 98–109, 2019, doi: 10.32782/cmis/2353-8.
 - [9] J. Jayakody, V. Vidanagama, I. Perera, and H. Herath, “BERT Layer Weighting Comparison with Short Text Classification,” in *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*, Peradeniya, Sri Lanka: IEEE, Aug. 2023, pp. 163–168. doi: 10.1109/ICIIS58898.2023.10253544.
 - [10] D. Bao, D. Qin, X. Liang, and L. Hong, “Short Text Classification Model Based on BERT and Fusion Network,” in *2021 5th International Conference on Computer Science and Artificial Intelligence*, Beijing China: ACM, Dec. 2021, pp. 168–174. doi: 10.1145/3507548.3507574.
 - [11] A. Kurniasih and L. P. Manik, “On the Role of Text Preprocessing in BERT Embedding-based DNNs for Classifying Informal Texts,” *IJACSA*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.01306109.
 - [12] T. Zhang, A. Tang, and R. Yan, “Enhanced BERT with Graph and Topic Information for Short Text Classification (S),” presented at the *The 35th International Conference on Software Engineering and Knowledge Engineering*, Jul. 2023, pp. 656–659. doi: 10.18293/SEKE2023-158.
 - [13] H. M. Keerthi Kumar and B. S. Harish, “Classification of Short Text Using Various Preprocessing Techniques: An Empirical Evaluation,” in *Recent Findings in Intelligent Computing Techniques*, vol. 709, P. K. Sa, S. Bakshi, I. K. Hatzilygeroudis, and M. N. Sahoo, Eds., in *Advances in Intelligent Systems and Computing*, vol. 709. , Singapore: Springer Singapore, 2018, pp. 19–30. doi: 10.1007/978-981-10-8633-5_3.
 - [14] N. Chayangkoon and A. Srivihok, “Feature Reduction of Short Text Classification by Using Bag of Words and Word Embedding,” *IJCA*, vol. 12, no. 2, pp. 1–16, Feb. 2019, doi: 10.33832/ijca.2019.12.2.01.
 - [15] K. RaghavanA, V. Umaashankar, and G. K. Gudur, “Label Frequency Transformation for Multi-Label Multi-Class Text Classification,” in *Conference on Natural Language Processing*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:208334363>
 - [16] C. Ma, W. Xu, P. Li, and Y. Yan, “Distributional Representations of Words for Short Text Classification,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado: Association for Computational Linguistics, 2015, pp. 33–38. doi: 10.3115/v1/W15-1505.

- [17] P. Wang, H. Zhang, Y.-F. Wu, B. Xu, and H.-W. Hao, "A robust framework for short text categorization based on topic model and integrated classifier," in 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China: IEEE, Jul. 2014, pp. 3534–3539. doi: 10.1109/IJCNN.2014.6889589.
- [18] G. Aguilar, B. McCann, T. Niu, N. Rajani, N. Keskar, and T. Solorio, "Char2Subword: Extending the Subword Embedding Space Using Robust Character Compositionality," 2020, arXiv. doi: 10.48550/ARXIV.2010.12730.
- [19] K. W. Church, "Emerging trends: Subwords, seriously?," *Nat. Lang. Eng.*, vol. 26, no. 3, pp. 375–382, May 2020, doi: 10.1017/S1351324920000145.
- [20] J. Wei, Q. Liu, Y. Guo, and X. Jiang, "Training Multilingual Pre-trained Language Model with Byte-level Subwords," 2021, arXiv. doi: 10.48550/ARXIV.2101.09469.
- [21] H. Yanagimoto and K. Hashimoto, "Improved Topic-bound Caption Generation with VGG-19, Batch Normalization, and Subword-based Tokenization," in 2022 13th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter), Phuket, Thailand: IEEE, Dec. 2022, pp. 51–56. doi: 10.1109/IIAI-AAI-Winter58034.2022.00020.
- [22] S. Arumugam, "A Multivariate Relevance Frequency Analysis Based Feature Selection for Classification of Short Text Data," *CSSE*, vol. 0, no. 0, pp. 1–10, 2024, doi: 10.32604/csse.2024.051770.
- [23] H. Wu et al., "Quartet Logic: A Four-Step Reasoning (QLFR) framework for advancing Short Text Classification," 2024, arXiv. doi: 10.48550/ARXIV.2401.03158.
- [24] J. Gong, J. Zhang, W. Guo, Z. Ma, and X. Lv, "Short Text Classification Based on Explicit and Implicit Multiscale Weighted Semantic Information," *Symmetry*, vol. 15, no. 11, p. 2008, Nov. 2023, doi: 10.3390/sym15112008.
- [25] Y. Zhu, Y. Wang, J. Qiang, and X. Wu, "Prompt-Learning for Short Text Classification," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 10, pp. 5328–5339, Oct. 2024, doi: 10.1109/TKDE.2023.3332787.
- [26] "Directionsforme." [Online]. Available: <https://www.directionsforme.org/>
- [27] O. Narushynska, V. Teslyuk, A. Doroshenko, and M. Arzubov, "Data Sorting Influence on Short Text Manual Labeling Quality for Hierarchical Classification," *BDCC*, vol. 8, no. 4, p. 41, Apr. 2024, doi: 10.3390/bdcc8040041.
- [28] T. Xu and P. Zhou, "Feature Extraction for Payload Classification: A Byte Pair Encoding Algorithm," in 2022 IEEE 8th International Conference on Computer and Communications (ICCC), Chengdu, China: IEEE, Dec. 2022, pp. 1–5. doi: 10.1109/ICCC56324.2022.10065977.
- [29] Z. Huang, "An Ensemble LLM Framework of Text Recognition Based on BERT and BPE Tokenization," in 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China: IEEE, Mar. 2024, pp. 1750–1754. doi: 10.1109/AINIT61980.2024.10581466.
- [30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," 2013, arXiv. doi: 10.48550/ARXIV.1301.3781.
- [31] R. Sekar and C. C. C., "Enhancing Credit Card Application Approval through Data Scaling in Machine Learning Algorithms," in 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), Theni, India: IEEE, Nov. 2023, pp. 1388–1394. doi: 10.1109/ICSCNA58489.2023.10370237.
- [32] U. Tıraşoğlu, A. Türker, A. Ekici, H. Yiğit, Y. E. Bölükbaşı, and T. Akgün, "Open Source Software Tools for Data Management and Deep Model Training Automation," in 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), Luxembourg, Luxembourg: IEEE, Sep. 2023, pp. 1814–1818. doi: 10.1109/ASE56229.2023.00014.
- [33] P. Tennage, A. Herath, M. Thilakarathne, P. Sandaruwan, and S. Ranathunga, "Transliteration and Byte Pair Encoding to Improve Tamil to Sinhala Neural Machine Translation," in 2018 Moratuwa Engineering Research Conference (MERCon), Moratuwa: IEEE, May 2018, pp. 390–395. doi: 10.1109/MERCon.2018.8421939.

- [34] E. Yang and Z. Long, "Research on the Weighting Method Based on Tf-IDF and Apriori Algorithm," in 2023 IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China: IEEE, Sep. 2023, pp. 1003–1005. doi: 10.1109/ICISCAE59047.2023.10393523.
- [35] J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying Detection using Pre-Trained BERT Model," in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India: IEEE, Jul. 2020, pp. 1096–1100. doi: 10.1109/ICESC48915.2020.9155700.
- [36] C. Min, J. Ahn, T. Lee, and D.-H. Im, "TK-BERT: Effective Model of Language Representation using Topic-based Knowledge Graphs," in 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, Republic of: IEEE, Jan. 2023, pp. 1–4. doi: 10.1109/IMCOM56909.2023.10035573.
- [37] Y. Moriya and Gareth. J. F. Jones, "Improving Noise Robustness for Spoken Content Retrieval Using Semi-Supervised ASR and N-Best Transcripts for BERT-Based Ranking Models," in 2022 IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar: IEEE, Jan. 2023, pp. 398–405. doi: 10.1109/SLT54892.2023.10023197.
- [38] Y. Tan, L. Jiang, P. Chen, and C. Tong, "DQMix-BERT: Distillation-aware Quantization with Mixed Precision for BERT Compression," in 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, Oahu, HI, USA: IEEE, Oct. 2023, pp. 311–316. doi: 10.1109/SMC53992.2023.10394642.