What Do Customers Say About My Products? Benchmarking Machine Learning Models for Need Identification

Sven Stahlmann
University of Cologne
stahlmann@wim.uni-koeln.de

Oliver Ettrich
Kühne Logistics University
oliver.ettrich@the-klu.org

Marco Kurka
University of Cologne
kurka@wim.uni-koeln.de

Detlef Schoder
University of Cologne
schoder@wim.uni-koeln.de

Abstract

Needmining is the process of extracting customer needs from user-generated content by classifying it as either informative or uninformative regarding need content. Contemporary studies achieve this by utilizing machine learning. However, models found in the literature cannot be compared to each other because they use private data for training and testing. This study benchmarks all previously suggested needmining models including CNN, SVM, RNN, and RoBERTa. To ensure an unbiased comparison, this study samples and annotates a dataset of customer reviews for products from 4 different categories from amazon. Henceforth, the dataset is publicly available and serves as a gold-set for future needmining benchmarks. RoBERTa outperformed other classifiers and seems to be best suited for needmining. The relevance of this study is reinforced by the fact that this benchmark creates a different hierarchy between models than otherwise suggested by comparing the results of previous studies.

Keywords: machine learning, natural language processing, customer needs, product innovation

1. Introduction

Knowing and understanding customer needs is an important tool to increase customer satisfaction and the quality of products and services (Matzler & Hinterhuber, 1998). For marketing departments, this can help to segment the market, identify strategic decisions (Park et al., 1986), and lead to better channel management decisions (Timoshenko & Hauser, 2019). For research and development departments, customer needs help to identify new product opportunities (Eppinger & Ulrich, 2015; Herrmann et al., 2000) or improve existing ones (Matzler & Hinterhuber, 1998). Therefore, knowing and understanding these needs is of great economic value.

To exploit this economic value, methods such as observations, surveys, and interviews are traditionally used to identify such needs (Edvardsson et al., 2012; Griffin & Hauser, 1993). However, these methods do not scale for large amounts of data, since a major part of the work is manual labor (Kühl et al., 2020). Additionally, these methods are cost-intensive (Fisher et al., 2014), time-consuming (Griffin & Hauser, 1993) and can result in a delay of time to market.

User-generated content (UGC) such as Amazon reviews and Twitter microblogs can be a low-cost source of customer needs (Kuehl et al., 2016) and are readily available in large quantities. Using traditional methods to identify needs in UGC is not feasible since the bulk of the content is either uninformative (i.e. it does not contain any needs) or repetitive. This results in high labor costs when analyzing UGC manually (Timoshenko & Hauser, 2019) because a lot of time is wasted on reading content that adds no value to market researchers. Consequently, a new, efficient approach to extract customer needs from UGC is needed. Machine learning looks to be a promising method to filter this vast amount of available UGC for need-containing content.

Studies tackling this problem already exist (e.g. Christensen et al., 2017; Kuehl et al., 2016; Stahlmann et al., 2022; Timoshenko & Hauser, 2019; Zhang et al., 2021). These studies apply various supervised machine learning classifiers to separate UGC into informative (defined as containing customer needs) and uninformative content. This is referred to as 'needmining' (Kuehl et al., 2016). Experts process the informative content further to extract insights for purposes such as innovation and product development. However, these studies have shortcomings. They all recommend different machine learning classifiers for the identification of needs in UGC and are evaluated on their own private datasets, making a direct comparison between these classifiers impossible. Therefore, between the different classifiers seen in previous studies, the research question arises "what algorithm or model excels at the task of need identification?".

URI: https://hdl.handle.net/10125/102895
978-0-9981331-6-4
(CC BY-NC-ND 4.0)

To close this research gap, this paper proposes a publicly available labeled dataset[1] that can serve as a 'gold set' for evaluating the performance of machine learning classifiers for need identification. Second, we use this gold set to evaluate previously proposed models in the literature, namely support-vector machines (SVM), naïve Bayes classifiers (Kuehl et al., 2016), convolutional neural nets (CNN) (Timoshenko & Hauser, 2019), ensemble embeddings based on a recurrent neural network (Zhang et al., 2021) and explore a pretrained transformer-based approach, RoBERTa, regarding its need identifying capabilities (Stahlmann et al., 2022).

This paper aims to make two major contributions. First, it gives an extensive overview of different machine learning classifiers' performances for the task of need identification in UGC. Second, it provides a publicly accessible gold set for the previously mentioned task, which future researchers can use to benchmark their models against. This unifies the data source for evaluating models and removes the necessity for each researcher to create a new labeled dataset for the purpose of needmining.

2. Related research

2.1. Customer needs

This study builds on the needmining literature to benchmark supervised machine learning models regarding their ability to binarily classify sentences into either informative or uninformative with regard to customer needs. To create a gold set, it is essential to clearly define the meaning of 'needs' to create as much consensus between labelers as possible. The exact definition of a need and especially the differentiation between need, want and demand has been tackled multiple times with varying results from different literature streams. In the following, we present the definitions of need primarily used in the marketing and product development literature.

One common approach for need definitions is to further divide need into wants and demands. Line (1974) has defined a need as what actually helps an individual reach one of their goals, a want as what they would like to have, and a demand as what the individual is verbally asking for. Demand, want and need can be inconsistent and conflicting. Arndt (1978) has defined a need as a requirement for an individual's well-being. Under the influence of internal and external factors, a need can become a more specific want and a want can become an even more specific demand. In contrast to Line (1974), a demand, want and need are consistent and only differ in specificity. Kotler et al. (2019) define a need as a basic human requirement. A want is the direction of a need towards a specific object and is shaped by society. Different people can have the same need, but different wants that satisfy it. The ability and willingness of an individual to pay for this transforms a want into a demand. Demand, want and need are consistent and differ in specificity and economic relevance. By understanding the needs and wants of their customers, marketing managers can create demand for their products.

Griffin and Hauser (1993) do not differentiate between the needs, wants, and demands of customers. They define a need as a description, in the customer's own words, of the benefit that the customer seeks to obtain from a product or service. This definition incorporates the different aspects of wants and demands into a single definition. It is possible to further map these needs onto product attributes that satisfy them.

Distinguishing between need, want and demand is a challenge, especially in a setting where the only available information is a sentence from a product review. Identifying small differences between these definitions is not feasible. For the purpose of this work, separating needs further also yields no significant benefits, since all subcategories of a need contain value for market researchers. Therefore, we refrain from this distinction and use the product-centric need definition of Griffin and Hauser (1993) to label our dataset. The idea of customer reviews is also most closely related to this concept because customers describe, in their own words, their experience with a product. This allows them, implicitly and explicitly, to express what they were seeking from a product they purchased and reviewed versus the experience they actually had.

2.2. Previous work on needmining

Several scholarly branches recognize the presence of relevant information, like customer needs or product attributes, in UGC such as product reviews. Product attribute mining, compared to needmining, focuses on more specific features that are wanted by customers. While these are important, it is only able to discover what the user specifically asks for. Needmining goes further by uncovering the benefit that they want to be fulfilled and leaves the expert to interpret how to achieve the result.

Already in the early 2000s, popular products on e-commerce platforms could generate hundreds and thousands of reviews. However, methodologies have

[1] Available under https://github.com/SvenStahlmann/HICSS-2023-Benchmarking-Machine-Learning-Models-for-Need-Identification

strongly advanced since people started exploring user-generated content for extracting needs. In the following, we examine several studies that have applied a variety of methods to predict the presence of customer needs or concepts closely related to needs in various bodies of user-generated content.

In the early days of the internet, websites were not as extensive in terms of UGC as they are today. Yet Hu and Liu (2004) predicted that in the future this source of data would become more relevant. They explored a mix of data mining and natural language processing to summarize reviews according to their features and opinions, an early attempt at what we would nowadays consider needmining. Their study employs part-of-speech tagging, association mining, and dictionary-based sentiment analysis with WordNet to determine features and opinions (Hu & Liu, 2004).

Misopoulos et al. (2014) primarily use sentiment analysis to analyze tweets about the airline industry. Their goal is to identify important features of customer service that either contribute to positive experiences or that require refinement. Using keyword search, they first allocate tweets into three categories. They then manually screen the data for keywords to create a customized lexicon that is the key driver of their sentiment analysis. The result of this study implies that sentiment analysis can be useful to determine parts of services that have negative or positive impacts on customer experience. While this study does not implicitly undertake needmining, it does generate results that overlap with the goals thereof.

Lee & Bradlow (2011) also find that customer reviews can contain product features and dimensions of these features. However, they rely entirely on methods outside the modern natural language processing domain to extract these. They transform preprocessed sentences into vectors, group the sentences according to their cosine-similarity, and finally apply k-means clustering to group product attributes (Lee & Bradlow, 2011). While their goal is ultimately to analyze market structure, their intermediate steps are situated well within the scope of needmining.

The first study that, to our knowledge, turns to more contemporary methods of needmining based on machine learning was conducted by Kuehl et al. (2016), who also coined the term 'needmining' itself. They analyze microblog data from Twitter with the goal of reducing a large body of user-generated content to a need-containing subset. Focusing on the domain of e-mobility, they label 2,400 German tweets and apply Bayes classifiers, SVMs, and tree-based classifiers with various configurations to predict whether a tweet contains customer needs or not. They benchmark these models by comparing performance metrics based on respective goals an innovation manager may have and provide model recommendations.

SVMs gained additional attention in the study by Christensen et al. (2017) who investigated 252 configurations of the algorithm to determine its performance on a binary classification (need versus no need in user-generated content) like that of Kuehl et al. (2016). They sample 3,000 reviews in the 'Toys & Games' domain and conclude that the linear support-vector machine in the right configuration can learn patterns that reflect ideas within textual data. Moreover, they carry out financial analysis of what an idea generated with such methods may cost compared to the cost of hiring raters and determine that an automated method may reduce the overall cost of identifying ideas in user-generated content.

The next iteration of machine learning methods with the goal of identifying user-generated content with customer needs are neural networks. Timoshenko and Hauser (2019) explore the feasibility of deep-learning for extracting customer needs from user-generated content. In their study, they first train word embeddings on unlabeled reviews which are then used to enhance a convolutional neural network to ultimately separate informative from uninformative content in a large body of Amazon reviews. They then cluster sentence embeddings to identify needs that can be differentiated from each other. In the last step, an expert team examines the resulting informative content and manually extracts customer needs. This method appears to identify more customer needs than a manual, traditional expert team would do, and at a lower overall cost.

Zhang et al. (2021) introduce a long short-term memory model for which they use a recurrent neural network-based ensemble from different concatenated word embeddings as input to capture a contextual representation of a sentence. This architecture is hereafter referred to as an RNN ensemble. They compare this model to other methods such as naïve Bayes, logistic regression, and SVMs and find that their deep-learning approach significantly outperforms all other models when considering the F1-score. Moreover, they test the impact of various ensemble embeddings and examine the impact of different embedding methods. In conclusion, their method excels at predicting the presence of innovation-related ideas in user-generated content compared to previous methods.

In summary, there have been a variety of approaches to mine customer needs from user-generated content. For the purpose of this study, they can be differentiated as non-machine learning concepts like text mining (Lee & Bradlow, 2011) and sentiment analysis (Hu & Liu, 2004; Misopoulos et al., 2014), and machine learning classification techniques such as naïve

Bayes, tree-based classifiers (Kuehl et al., 2016), SVMs (Christensen et al., 2017; Kuehl et al., 2016), and neural networks in combination with word embeddings (Timoshenko & Hauser, 2019; Zhang et al., 2021).

3. Method

We use design science research as a framework for this study. Design science has become established as a fundamental research paradigm in information systems (Gregor & Hevner, 2013). Its goal is to acquire knowledge through the development of novel artifacts (Hevner et al., 2004). Design science research grows from a knowledge base that is not only constantly being expanded but also validated.

We structure our research based on the approach presented by Kuechler and Vaishnavi (2008). In the first step of understanding the problem, we carried out an extensive literature review on the topic of need extraction from UGC, as well as interviews with practitioners and experts. After gaining a clear understanding of the problem domain, we identified possible UGC data sources for our dataset, as well as candidate models for the evaluation. We selected a data source and used a web scraper to retrieve the data. Guided by our literature analysis, we implement all previously proposed machine learning models for need identification from UGC found in the literature. We first executed an inter-rater agreement test on a subset of the data to determine if human raters can reliably and objectively identify needs from product reviews based on a given definition. Following the positive inter-rater agreement test, human raters manually classified each sentence in our scraped data according to whether or not it contains a customer need. Finally, we use the dataset as a gold set to evaluate all implemented models. In the following sections, we describe the previously mentioned steps in more detail.

4. Data

4.1. Data retrieval

The first step in creating a labeled gold set for the evaluation is to identify potential data sources that can be used for the extraction of needs. Prior research dealing with customer needs has used travel websites like Tripadvisor (Barreda & Bilgihan, 2013) or Expedia (Büschken & Allenby, 2016) as well as microblogs such as Twitter (Kuehl et al., 2016) and product reviews from e-commerce platforms (Timoshenko & Hauser, 2019; Zhang et al., 2021). We disregarded travel websites as a potential source of data since the specialized application profile makes the extraction of needs only possible for products and services in the domain of travel and we want the data source to be as generally applicable as possible. Second, we excluded microblogs as a possible data source. Microblogs such as Twitter are extremely diverse in terms of the content that is discussed on the platform. Therefore, retrieving the content discussing the target topic that should be analyzed can be challenging. For example, Kuehl et al. (2016) set up a workshop with three industry professionals to create a keyword set to retrieve content relative to their target topic of e-mobility. In contrast, e-commerce platforms such as Amazon offer product review sections where customers can freely discuss a product. This has the advantage that content is already linked to the matching product, making retrieval of related content more accessible. Compared to specialized sites such as Tripadvisor, Amazon also offers a wide range of product categories, opening up possibilities for a large variety of product domains. Additionally, Amazon data are easy to acquire and publicly available in large quantities, either as a provided dataset (e.g. He & McAuley, 2016) or through web scraping. This leads us to believe that Amazon is also a practical source for extraction needs for the industry. Therefore, due to the variety and availability of its content, we have chosen Amazon as the data source for our gold set.

To retrieve the data from Amazon, we programmed a Python-based web scraper. We opted to obtain data of four different product categories, namely ‘Baby’, ‘Sports & Outdoors’, ‘Electronics’, and ‘Pet Supplies’. All four categories are offered by Amazon as a selection criterion when searching for products. We chose these categories to get a diverse set of products for the analysis since our initial examination of reviews for those products revealed only a small overlap of needs between these categories. The scraper ran in November 2019 and collected all product reviews from the top three most popular products in each category on Amazon’s US platform. We followed Timoshenko and Hauser (2019) and split the product reviews into individual sentences using the Natural Language Toolkit in Python (Bird et al., 2009).

4.2. Data coding

The distinction governing whether a customer need is truly articulated in a sentence cannot be fully assessed without further feedback from the author. Therefore, we first tested whether human raters agree on what constitutes a need-containing sentence. After all, if humans have difficulties agreeing on whether there is a need in a sentence or not, then a machine learning model will be equally confused as it will not outperform the data it trains on. We analyzed the degree of agreement among raters by randomly selecting 1,500 sentences

from reviews in the 'Pet Supplies' category. Three independent raters were instructed to label each sentence according to whether it contained a customer need based on the definition by Griffin and Hauser (1993). This resulted in three labels per sentence, one for each rater.

Following Egger et al. (2015), we measure the agreement of the raters on each sentence, also known as inter-rater reliability. Additionally, the free-marginal multirater kappa was computed (Randolph, 2010). The free-marginal multirater kappa was chosen over the more popular Fleiss' multirater kappa because the latter requires raters to have guidelines for the distribution of labels when performing the coding (Randolph, 2010), which in this case is not possible. For the free-marginal multirater kappa the distribution of the labels does not need to be known in advance. The metric is defined between -1 and 1, where -1 denotes a below-stochastic match, 0 equals a stochastic hit, and 1 represents a perfect match (Egger et al., 2015).

Table 1. Rater agreement on the task of need identification.

	Rater 1	Rater 2	Rater 3
Rater 1	-	82%	84%
Rater 2	82%	-	82%
Rater 3	84%	82%	-

This study achieves an average agreement of 83% between all raters and a kappa of 0.6613. Table 1 displays the results on a per rater basis. Notably, all raters share a commonly high agreement on what constitutes a need in a sentence and there are no outliers (large disagreement) between two raters, therefore no action to further improve the agreement was necessary. These results led us to the conclusion that the identification of needs from sentences can be objectively achieved using only textual features present in a sentence.

Table 2. Example sentences of the labeled data.

Need Label	No Need Label
The sound system is great	She's my everything.
It can hear me all over my small house.	I love it.
In my opinion there is not enough padding for more than an hour or two drive.	So beautiful...

Based on the positive outcome of the inter-rater agreement test, we continued with labeling a larger dataset to be used as a gold set for the evaluation of the models. As mentioned, we chose to include multiple product categories in our dataset to be able to evaluate the models on multiple domains. These categories are 'Baby', 'Sports & Outdoors', 'Electronics' and 'Pet Supplies'. From each category, we extracted 2,000 sentences which were evenly distributed from the top three most popular products in that category. This resulted in a dataset with a total size of 8,000 sentences. We asked the same human raters to annotate the sentences according to whether they contained a customer need based on the definition of Griffin and Hauser (1993), examples are shown in Table 2. The raters were paid an hourly wage for the task.

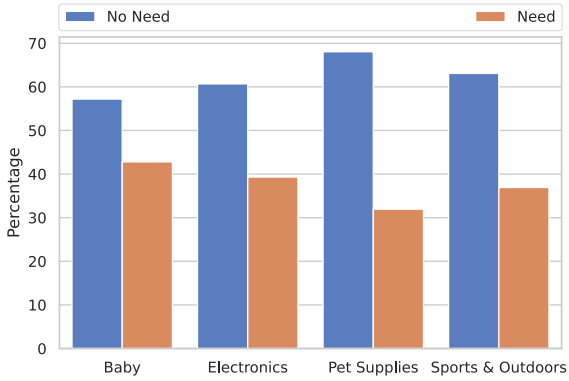

Figure 1. Class label distribution of different product categories.

Figure 1 displays the class distribution of customer needs for the different product categories. We can observe that all categories have an unbalanced distribution of the class labels. The majority of all sentences do not contain a need. The lowest percentage of need-containing sentences is the 'Pet Supplies' category with around 32%, while the 'Baby' category has the highest proportion of needs with 42%.

5. Models

5.1 Model implementation

This section lays out how we implemented the previously suggested model architectures for the extraction of needs as well as the key libraries we used. Our implemented models and data are publicly available[2].

For RoBERTa we used the pretrained 'roberta-base' model weights provided by HuggingFace's Transformers library (Wolf et al., 2019). As suggested

[2] https://github.com/SvenStahlmann/HICSS-2023-Benchmarking-Machine-Learning-Models-for-Need-Identification

by Devlin et al. (2019), we used the pooled output of the special [CLS] token and fed it into two fully connected linear layers with 768 neurons each. We applied dropout between each layer. We chose Cross Entropy as the loss function and used Adam for optimization.

For the SVM and naïve Bayes implementation we used the scikit-learn library (Pedregosa et al., 2011). The SVM used a linear kernel, and for naïve Bayes we used the multinomial naïve Bayes implementation, suitable for text classification with an alpha value of 1. For both models we tested two different tokenization formats, a bag of words and a tf-idf vector approach. We found that the tf-idf vector approach yielded better results across all product categories for both models, and therefore we report the results based on the tf-idf vector input.

We implemented the best performing CNN as described by Timoshenko and Hauser (2019) based on Kim (2014). We have chosen the same hyperparameters reported by Timoshenko and Hauser (2019) using filters with the size 3, 4, and 5. We used three filters of each size resulting in a total of nine filters. We initialized the embedding layer using pretrained GloVe word vectors trained on Wikipedia and Gigawords (Pennington et al., 2014) with a vector dimension of 300.

In the following, we describe how we implemented the RNN-based ensemble embeddings from Zhang et al. (2021). Due to missing implementation details, we had to make educated guesses in some cases which are presented here. Since the authors describe using vectors with 1,024 dimensions for BERT and XLNet, we selected the large models of BERT and XLNet, namly 'bert-large-uncased' and 'xlnet-large-cased' to generate the BERT and XLNet embeddings. To generate the GloVe embeddings we used the GloVe model pretrained on a 2014 Wikipedia dump and Gigawords 5 corpus (Pennington et al., 2014), since it is the most popular model matching the dimensionality reported by Zhang et al. (2021). BERT and XLNet produce embeddings based on subwords, while GloVe produces embeddings based on words, and therefore it is not possible to concatenate these embeddings directly. Zhang et al. (2021) do not report how they dealt with this problem. We merged the subword embeddings from BERT and XLNet by averaging the embedding of each subword of a word. We implemented the ensemble embedding using a SimpleRNN layer with 256 nodes. The created ensembled word embeddings are then passed to a Bi-LSTM (Hochreiter & Schmidhuber, 1997) layer with 256 nodes which represents the output for the entire sentence. The resulting vector is then passed to a linear layer with a sigmoid activation function which performs the final classification. We used the focal loss function as stated in the paper to counter the imbalance of the data with the same gamma and alpha values as Zhang et al. (2021).

6. Results

In the previous sections we used the terms algorithm, classifier, and model interchangeably. Moving forward, we differentiate specifically between these terms. When we talk about a model, this is an instance of a trained algorithm or classifier, with the algorithm or classifier referring to the architecture used to calculate the results, for example an SVM.

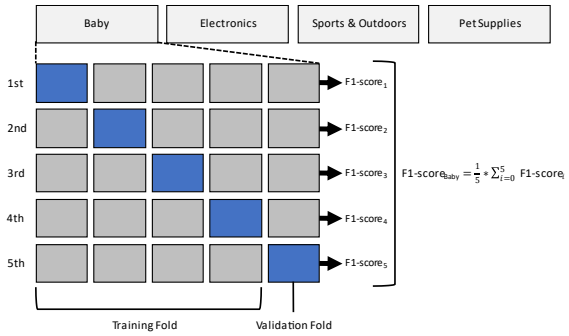

Figure 2. K-fold cross-validation on the 'Baby' product category for a single classifier.

This section presents the result of the benchmarking study based on the created gold set. We assess all classifiers previously proposed in the literature, namely SVM, naïve Bayes, CNN, and RNN ensemble as well as RoBERTa, a pretrained transformer. We evaluated the classifiers separately on each of the four product categories to match Timoshenko and Hauser (2019), who trained a new model for each product category in their further application. We use the macro averaged F1-score as the main evaluation criterion. Figure 2 shows our implementation of a k-fold cross validation, which ensures that the numbers presented are not a result of chance caused by random train–test splits (Cawley & Talbot, 2010). We selected k = 5, which means that for each category the data were split into five equally sized, mutually exclusive subsets called folds. Per category, we train five different models on different fold combinations, where one fold serves as a hold-out set while the four remaining are used to train the model. This results in five different scores per classifier and per category. The reported scores for each category are the average performance of each model on its respective hold-out set.

Table 3 provides an overview of the scores for each classifier in each category, as well as the average performance of each classifier based on each individual category. The transformer-based solution RoBERTa consistently outperforms all other methods across all categories and therefore also gained the highest average score. This is followed by CNN, SVM, RNN ensemble and finally naïve Bayes.

The RoBERTa, CNN, SVM and RNN ensemble classifiers perform best for the product category 'Pet Supplies', and second best for 'Baby'. Naïve Bayes is the notable exception, performing worst in the category 'Pet Supplies' and best in the category 'Baby'. Naïve Bayes also reports the largest diffeerence in performance among product categories, with over 10% between the best and the worst performing category. Within the product categories 'Electronics' and 'Sports & Outdoors', each classifier reports a somewhat similar performance, with RoBERTa and naïve Bayes performing slightly better in 'Sports & Outdoors' while the other classifiers perform better in 'Electronics'.

Looking solely at the average performance of the models can be misleading since the performance of the models can vary based on different factors, e.g., the training test fold split or the random initialization of the model. Therefore it is also important to look at the distribution of the performance metric of each classifier. Since we applied 5-fold cross validation we have the performance measures of five different models in each product category. As seen in Figure 2 this results in a total of 20 different F1-scores per classifier. Figure 3 visualizes the distribution for each classifier.

The box plots show different quartiles as well as outliers sorted by classifier. The lower bound of the box represents the first quartile (25th percentile) while the upper bound of the box represents the third quartile (75 percentile). Therefore, the box displays the range of values in which the middle 50% of the data is located, also called the interquartile range. The horizontal line inside the box represents the median of the distribution. The whiskers extend to the nearest data point that is within 1.5 times the distance of the interquartile range from the upper or lower bound of the box. Datapoints outside the whiskers are plotted as outliers.

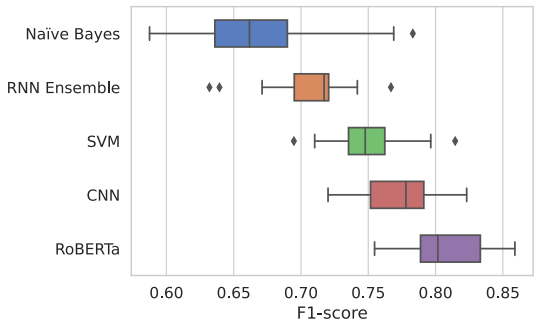

Figure 3. Box plot of the F1-score of the classifiers.

The ranking of the median of the classifiers follows the same sequence as the average performance across all categories. We can see that RoBERTa has the highest median as well as first and third quartile. RoBERTa's median, and almost the classifier's first percentile, is higher than all third quartiles of the other classifiers. The first percentile of RoBERTa is higher than every median of a different classifier; this can be interpreted as over 75% percent of RoBERTa models performing better than the median of the second best classifier, the CNN.

We can also observe that the SVM and RNN ensemble have a relatively dense interquartile range but both have notable outliers. RoBERTa and the CNN have bigger interquartile ranges but no extreme performance values that can be identified as outliers, making them more consistent in their overall performance. In contrast to this, naïve Bayes has the largest interquartile range and notable outliers making it the most inconsistent. This large deviation in performance can also be observed in Table 3 with the 'Baby' product category.

Table 3. Macro averaged F1-scores of the classifiers on different product categories.

	Naïve Bayes	RNN Ensemble	SVM	CNN	RoBERTa
Baby	75.8%	71.6%	76.5%	78.5%	81.5%
Electronics	65.0%	70.0%	73.3%	75.43%	78.8%
Pet Supplies	62.6%	72.8%	77.4%	79.4%	83.0%
Sports & Outdoors	66.3%	68.3%	73.0%	75.0%	79.1%
Average Performance	67.4%	70.7%	75.1%	77.1%	80.6%

7. Discussion

This study aimed at benchmarking the performance of various machine learning algorithms on the binary classification task that predicts whether or not a sentence from a customer review contains a need or not. Throughout the needmining literature, this task is considered highly relevant as it separates the amount of useless, uninformative content from the valuable, informative content (Christensen et al., 2017; Kuehl et al., 2016; Timoshenko & Hauser, 2019; Zhang et al., 2021). This makes any subsequent processing, which in the current state of needmining is mostly manual, both less resource intensive and faster (Timoshenko & Hauser, 2019). However, before this study there was a

lack of information on which machine learning classifier delivers the best performance because there was no unbiased way to compare them. We trained and tested all major types of needmining models on a well-validated gold set, which comprises product reviews from four distinguished product categories sourced from Amazon.com. This allows for an unbiased comparison between the models and results in some interesting outcomes regarding the differences in performance reported in the studies of origin and in our direct comparison.

First, our results indicate that the implementation of RoBERTa, and by extension transformers, significantly outperforms all other machine learning methods identified throughout the literature on needmining. Although this is unsurprising in itself given that transformers have been shown to outperform other models in a variety of tasks (Liu et al., 2019; Talmor et al., 2020), this benchmark presents this finding very consistently, as Figure 3 shows.

Second, we found several major inconsistencies between our benchmark and some of the results reported in the original studies. Most notably, the RNN ensemble employed in Zhang et al. (2021) performed second worst in our benchmark and on average only got an F1-score of 70.7% in comparison to the original study's F1-score of 88% - 89%. Kuehl et al. (2016) show F1-scores of between 38.5% - 45.4% for SVM and between 36.1% - 39.7% for naïve Bayes, compared to our F1 results of 75.1% (SVM) and 67.4% (naïve Bayes), respectively. Only the results of Timoshenko and Hauser (2019) were remotely close to the results we reported, as our CNN gained an F1-score of 77.1% on average to their 74% F1-score. On a second note to this particular study, Timoshenko and Hauser (2019) also tested other machine learning methods, most importantly SVMs, which gained an F1-score of 65.7%, which is also much closer to our SVM than the implementation by Kuehl et al. (2016).

While there may be many relevant explanations for these deviations and similarities, we are going to discuss some that we consider most likely to have had an impact on the results. First, differences between source platforms occur depending on how users write their content or on platform restrictions. This is most relevant for the comparison between our results and Kuehl et al. (2016) as the data used were sourced from Amazon and Twitter, respectively. While Amazon reviews can be written with a large amount of freedom, Twitter restricts users to a certain number of characters. Moreover, Amazon reviews directly talk about experiences with products and their potential benefits and shortcomings. Twitter, however, is a discussion forum for all sorts of topics where discussing specific needs may occur more rarely.

Next, we consider implementation differences. This mostly applies to the comparison between our study and that of Zhang et al. (2021) but may also be relevant for some of the other studies. As noted previously, the implementation details in the original paper lack some information, forcing us to make educated guesses based on what would be considered industry standard. This could lead to some of the significant differences we observe where the original model outperforms any of our models by quite a large margin. Since they were also using review data from the Amazon website, the differences cannot originate from source platform variations. However, they may be caused by differences between product categories.

We account for differences in product categories by testing all different machine learning algorithms on four categories that significantly differ from each in terms of the needs that may be expressed. However, similarly to the differences that could be caused by the source platform, there may also be some product categories where detecting customer needs is inherently more difficult. Anecdotally, categories such as 'gift certificates' or 'magazine subscriptions' simply do not contain any customer needs or are much more difficult to detect. While we did our best to spread our gold set across a variety of product categories, some categories we did not test could still lead to much stronger or weaker results on that category. However, our effort to test the models on the same data across four product categories provides sufficiently robust information for a solid benchmark between the models, thus giving a general overview of which current approach is best in the needmining environment. We also selected the three most popular items out of each category, resulting in a possible bias for well-rated products.

To summarize, regardless of the reasons for the differences between this study and the others we discussed, these differences themselves highlight the importance of this benchmark for the overall needmining literature. It provides a clear and unbiased comparison between major models by testing them on a single, self-compiled gold set. Future scholars and practitioners alike can consult this study to compare solutions for mining customer needs in user-generated content. The clear suggestion we provide based on our results is to use a transformer-based approach given the performance we observed with the RoBERTa model. In the future, the dataset and model implementations we use in this study will be freely available for comparing any newer approaches to classifying sentences in terms of their need-content. This is important because all needmining studies we investigated during the course of this study train and test their models on a private, hard to reproduce dataset making it difficult to directly

compare other models to theirs, or to reproduce their results.

8. Conclusion and further research

In this study we benchmarked the most relevant machine learning models in the context of mining customer needs from user-generated content by separating informative from non-informative content. So far, other studies have applied SVMs, CNNs, naïve Bayes classifiers, and an RNN ensemble. To this collection we add one implementation of RoBERTa as a representation of the novel transformer architecture given these models' superior performance in many natural language processing tasks. In a preceding step we validated whether customer needs can be objectively identified in product review sentences by measuring inter-rater agreement of three raters for this task. Following the positive outcome of this, we created a labeled dataset of customer reviews within four categories from Amazon. This set represents a 'gold set' in the needmining literature which we use to evaluate the different models. We make this labeled dataset publicly available for further researchers to have a commonly agreed-upon dataset for future needmining research.

Our results indicate that RoBERTa outperformed all other models by a significant margin and maintained stable prediction results across all categories. This aligns with the outcome based on the literature on recent transfer learning-based transformers. CNN performed the second best, followed by SVM, RNN ensemble and naïve Bayes. Except for naïve Bayes, the performance of the models did not significantly differ between the four product categories analyzed.

Comparing the results of our benchmark to previous studies we found deviations in performance, especially with regard to the RNN ensemble model. We were not able to verify the RNN ensemble results reported by Zhang et al. (2021). Our SVM results exceeded the scores reported by Kuehl et al. (2016) and Christensen et al. (2017) while broadly matching the scores published by Timoshenko and Hauser (2019).

While this study gives a good overview of model performances, further research in this area is needed to uncover their strengths and weaknesses and do a qualitative assessment. More knowledge needs to be gathered regarding whether the choice of data sources (e.g. Amazon versus Twitter) affects the performance of the model. Current research also largely focuses on the improvement of products, whereas services have vastly different characteristics but could also benefit from the mining of large amounts of user-generated content. Future research is needed to ascertain whether performance differences occur when the target is a service rather than a product. Lastly, machine learning models can be further improved using various techniques derived from the area of natural language processing.

9. Acknowledgements

We gratefully acknowledge the support of the German Research Foundation (DFG), project FOR 1452 / TP1 SCHO 1321/3-2. We also thank Henri Beyer, Felicia Preuss-Neudorf and Patrick Seidel for the support of the research.

10. References

Arndt, J. (1978). How Broad Should the Marketing Concept Be? Should it be developed into a full-fledged behavioral science? Journal of Marketing, 42(1), 101–103.

Barreda, A., & Bilgihan, A. (2013). An analysis of user-generated content for hotel experiences. Journal of Hospitality and Tourism Technology, 4(3), 263–280. https://doi.org/10.1108/JHTT-01-2013-0001

Bird, S., Loper, E., & Klein, E. (2009). Natural language processing with python O'reilly media Inc.

Büschken, J., & Allenby, G. M. (2016). Sentence-Based Text Analysis for Customer Reviews. Marketing Science, 35(6), 953–975. https://doi.org/10.1287/mksc.2016.0993

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. The Journal of Machine Learning Research, 11, 2079–2107.

Christensen, K., Nørskov, S., Frederiksen, L., & Scholderer, J. (2017). In Search of New Product Ideas: Identifying Ideas in Online Communities by Machine Learning and Text Mining. Creativity and Innovation Management, 26(1), 17–30. https://doi.org/10.1111/caim.12202

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs]. http://arxiv.org/abs/1810.04805

Edvardsson, B., Kristensson, P., Magnusson, P., & Sundström, E. (2012). Customer integration within service development—A review of methods and an analysis of insitu and exsitu contributions. Technovation, 32(7), 419–429. https://doi.org/10.1016/j.technovation.2011.04.006

Egger, M., Lang, A., & Schoder, D. (2015). Who Are We Listening to? Detecting User-generated Content (UGC) on the Web. ECIS 2015 Completed Research Papers. https://doi.org/10.18151/7217308

Eppinger, S., & Ulrich, K. (2015). Product design and development. McGraw-Hill Higher Education.

Fisher, M., Houghton, M., & Jain, V. (2014). Cambridge IGCSE® Business Studies Coursebook with CD-ROM. Cambridge University Press.

Gokaslan, A., & Cohen, V. (2019). OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. MIS quarterly, 337–355.

Griffin, A., & Hauser, J. R. (1993). The voice of the customer. Marketing science, 12(1), 1–27.

He, R., & McAuley, J. (2016). Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. Proceedings of the 25th International Conference on World Wide Web, 507–517. https://doi.org/10.1145/2872427.2883037

Herrmann, A., Huber, F., & Braunstein, C. (2000). Market-driven product and service design: Bridging the gap between customer needs, quality management, and customer satisfaction. International Journal of production economics, 66(1), 77–96.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS quarterly, 28(1), 75–105.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735–1780.

Hu, M., & Liu, B. (2004). Mining and Summarizing Customer Reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168–177. https://doi.org/10.1145/1014052.1014073

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751. https://doi.org/10.3115/v1/D14-1181

Kotler, P. T., Brady, M., Goodman, M., & Hansen, T. (2019). Marketing Management PDF eBook. Pearson Deutschland. https://elibrary.pearson.de/book/99.150005/9781292248479

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: Anatomy of a research project. European Journal of Information Systems, 17(5), 489–504. https://doi.org/10.1057/ejis.2008.40

Kuehl, N., Scheurenbrand, J., & Satzger, G. (2016). NEEDMINING: IDENTIFYING MICRO BLOG DATA CONTAINING CUSTOMER NEEDS. Research Papers. https://aisel.aisnet.org/ecis2016_rp/185

Kühl, N., Mühlthaler, M., & Goutier, M. (2020). Supporting customer-oriented marketing with artificial intelligence: Automatically quantifying customer needs from social media. Electronic Markets, 30(2), 351–367. https://doi.org/10.1007/s12525-019-00351-0

Lee, T. Y., & Bradlow, E. T. (2011). Automated marketing research using online customer reviews. Journal of Marketing Research, 48(5), 881–894.

Line, M. B. (1974). Draft definitions: Information and library needs, wants, demands and uses. Aslib Proceedings, 26(2), 87–87. https://doi.org/10.1108/eb050451

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Matzler, K., & Hinterhuber, H. H. (1998). How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. Technovation, 18(1), 25–38.

Misopoulos, F., Mitic, M., Kapoulas, A., & Karapiperis, C. (2014). Uncovering customer service experiences with Twitter: The case of airline industry. Management Decision, 52(4), 705–723. https://doi.org/10.1108/MD-03-2012-0235

Park, C. W., Jaworski, B. J., & MacInnis, D. J. (1986). Strategic Brand Concept-Image Management. Journal of Marketing, 50(4), 135–145. https://doi.org/10.1177/002224298605000401

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825–2830.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1532–1543.

Randolph, J. (2010). Free-Marginal Multirater Kappa (multirater κfree): An Alternative to Fleiss Fixed-Marginal Multirater Kappa. Advances in Data Analysis and Classification, 4.

Stahlmann, S., Ettrich, O., & Schoder, D. (2022). Deep Learning Enabled Consumer Research for Product Development. ECIS 2022. https://aisel.aisnet.org/ecis2022_rip/19

Talmor, A., Elazar, Y., Goldberg, Y., & Berant, J. (2020). OLMpics-On What Language Model Pre-training Captures. Transactions of the Association for Computational Linguistics, 8, 743–758. https://doi.org/10.1162/tacl_a_00342

Timoshenko, A., & Hauser, J. R. (2019). Identifying customer needs from user-generated content. Marketing Science, 38(1), 1–20.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., & Funtowicz, M. (2019). HuggingFace's Transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Zhang, M., Fan, B., Zhang, N., Wang, W., & Fan, W. (2021). Mining product innovation ideas from online reviews. Information Processing & Management, 58(1), 102389. https://doi.org/10.1016/j.ipm.2020.102389

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. Proceedings of the IEEE international conference on computer vision, 19–27.