Lot or Not: Identifying Multi-Quantity Offerings in E-Commerce

Anonymous ACL submission

Abstract

The term *lot* in e-commerce is defined to mean an offering that contains a collection of multiple identical items for sale. In a large online marketplace, lot offerings play an important role, allowing buyers and sellers to set price levels to optimally balance supply and demand needs. In spite of their central role, e-commerce platforms often struggle to identify lot offerings, since explicit lot status identification is frequently not provided by sellers. The ability to identify lot offerings plays a key role in many fundamental e-commerce tasks, from matching offerings to catalog products, through ranking e-commerce search results, to providing effective pricing guidance. In this work, we seek to determine the lot status (and lot size) of each offering, in order to facilitate an improved buyer experience, while reducing the friction for sellers posting new offerings. We demonstrate experimentally the ability to accurately classify offerings as lots and predict their lot size using only the offer title, by adapting state-of-the-art natural language techniques to the lot identification problem.

1 Introduction

The term *lot* has its origins in the world of live auctions, where it describes the atomic unit for sale. Each such lot usually has an associated multiplicity (or *lot size*). In global e-commerce marketplaces, the variety of products for sale is several orders of magnitude larger than that of live auctions. In this latter setting, the atomic unit for sale is referred to as an *offering* or *listing* , and does not usually have an associated multiplicity (i.e., only one object is for sale). Thus, in the e-commerce setting, the term *lot* (or lot offering) is redefined to describe those offerings that contain a collection of multiple identical items. That is, not every offering is a lot. We further define the term *lot size* as the multiplicity of identical items in the collection for sale.

We adopt the definition for a lot offering given by eBay in its guidelines[1] to sellers:

> *A "lot" is a group of similar or identical items that are sold together to one buyer*

Amazon uses a similar definition for the related term *multi-packs*.[2]

Lots, or multi-packs, are distinguished from *bundle* offerings, which contain multiple *distinct* (rather than *identical*) items (Tzaban et al., 2020).

The ability to list lot offerings provides great flexibility to sellers. One reason for this is that many products come from the manufacturer as lots (e.g., a box of pencils). Another reason is that lot offerings provide the seller an additional degree of freedom (the lot size), in addition to price, to maximize marketplace value by adapting to the demand of the market for their particular product.

Online marketplaces strive to distinguish between lot and non-lot offerings for several reasons. The first reason is to enable discovery of lot offerings as first class citizens in the electronic marketplace. That is, a local retail entrepreneur may be looking for lots of products independent of what the actual product happens to be, seeking to gain profit by buying lots and reselling the component items individually.

Another important scenario is allowing price-per-unit comparison in aggregate. E-Commerce buyers, seeking the best value, may be willing to consider purchasing a larger quantity of a product in return for a per-unit discount. Consider the offerings for protective masks depicted in Figure 1. Without detecting and considering the lot size of these comparable offerings, it is difficult for a customer to recognize that some offers cost much more on a per unit basis.

[1] https://www.ebay.com/pages/cn/help/sell/contextual/lots.html

[2] https://tinyurl.com/y6kej274

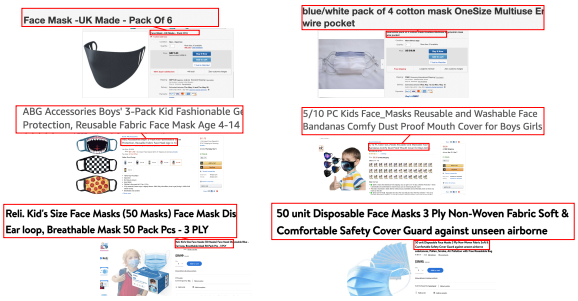

Figure 1: A comparison of lot offerings for protective masks across 3 e-commerce sites: `ebay.com`, `amazon.com`, and `walmart.com` The price per unit varies across these offerings from $0.40 to $5.00.

This work is motivated to automatically detect lot offerings, but a critical reader may ask, why not ask the sellers to explicitly designate their lot offerings (and provide an explicit lot size)? In fact, such an option does exist on many e-commerce platforms. eBay[3] , for example, has a standalone interface for sellers to input lot entries. Unfortunately, the adoption of this feature among the seller population is quite low. While sellers often have an incentive to clearly designate their offering as a lot, in practice interfaces to specify structured lot metadata are difficult to navigate. These interfaces are often unfamiliar to sellers and not standardized across marketplaces. This issue becomes more acute when sellers upload their offerings in (often large) batches, using a non-visual interface, to multiple marketplaces.

Rather than mark the offering as a lot explicitly, a common practice of many e-commerce sellers is to declare the lot status (and lot size) in the offering title using natural language. Since the title field exists across all e-commerce platforms and is prominently displayed to potential buyers, sellers can apply this technique to convey important offering information (such as lot status and size), without needing to understand the nuance of any particular marketplace interface, as well as its own terminology and attribute definitions. Table 1 illustrates example titles of lot offerings, which were not explicitly designated as lot offerings by the sellers. The examples in the table demonstrate the diversity of offerings that contain lots and the unique and colorful language of jargon and abbreviations to specify the lot status (and lot size) of the offering.

In this work, we seek to determine the lot status (and lot size) of each offering, in order to facilitate the scenarios enumerated above for buyers, while reducing the friction for sellers. Although e-commerce offerings contain multiple sources of information (e.g. images, descriptions, etc.) our methods focus exclusively on the offering title. The first reason for this is the presence of powerful natural language cues for lot status and size. This is anecdotally demonstrated in Table 1. Another reason is broad applicability: while many offerings are incomplete to some degree, with lacking or altogether-missing attributes (Ghani et al., 2006), descriptions (Novgorodov et al., 2019) and images (Goswami et al., 2012), the vast majority of offerings contain a valid title. We show experimentally that methods based on recent advances in natural language processing, but adapted to the problem of lot identification, are able to achieve high-performance on our tasks of interest.

Our main contributions can be summarized as follows:

- We introduce the first comprehensive study of lot identification in e-commerce.
- We release a dataset with nearly 20,000 offering titles across multiple categories, each labeled with lot status and lot size.
- We propose an adaptation of the naive regression approach to lot-size prediction, based on binary sequence models, which achieves high accuracy on this task.
- We empirically evaluate the performance of our proposed approach across several e-commerce domains and compare performance of several state-of-the-art methods.

2 Related Work

In this work, we apply a variety of natural language processing methods to offering titles to address the lot identification task. Accordingly, we review related work in two areas: research related to lots or multipacks in electronic commerce, and text representation and classification approaches relevant to our task.

2.1 Lots in E-Commerce

Despite their central role in online marketplaces, the current literature on lots or multipacks is very sparse. In a study from 1996, Lindskog and Lundgren (Lindskog and Lundgren, 1996) examined the use of multipacks in 41 physical stores in the UK

[3]https://pages.ebay.com/sell/lots/

Table 1: Examples of lot offering titles. The lot size (highlighted for emphasis) is often included somewhere in the title in sometimes colorful shorthand.

Lot Size	Title	Category
3	Lot of 3 Vtg. 1974 ENESCO IMPORT Rustic Metal Sculptures Wagon Telephone MailBox	Collectibles
5	5 PACKETS EPIL-STOP PERFECT FINISH NEUTRALIZING AFTER WASH FREE SHIPPING USA NEW	Health & Beauty
20,000	Antique German Doubled Baked Ceramic Bricks 20000 pcs	Antiques
1000	(1000) CD Disc Jewel Case Bin Divider Cards - 5-5/8"x6" - White HEAVY DUTY 30mil	Music
2	Genuine OEM 2 Pack Canon PG-220 Black PGI-220BK Ink Tank NEW	Computers/Tablets & Networking
22	ALL BRAND NEW...LOT OF 22 KIDS GIFT ITEMS	Toys & Hobbies
50	Varian 1210-2046 Analytichem Bond Elut box of 50 SEALED BOX	Business & Industrial
28	LOT 28x 459512-002 375863-010 HP 146GB 3G SAS 10K SFF 2.5" HDD HARD DRIVE NR	Computers/Tablets & Networking
2	Pier 1 Curtain Panels (set of 2) gold, burgundy, green with geo design 84" long	Home & Garden

and Sweden. They discussed the different benefits, mostly related to production costs, packaging, storage, distribution, and increased sales due to the discounted prices. In their work on matching offerings to catalog products, Shah et al. (Shah et al., 2018) note that lots make data ambiguous, since, for example, "*a number in a product description could refer to a lot quantity or variation of product edition*". They state that such product offerings exhibit another level of complexity and require special treatment or a separate model to identify, but do not further explore this task. Zentes et al. (Zentes et al., 2017) mention multipacks as one of the main strategies for price reductions, but do not further characterize it compared to other promotion approaches, such as coupons or price packs.

A key research challenge in the e-commerce domain is the extraction of structured key-value attributes, such as brand, model, size, or color, from the titles of products or offerings.Techniques to approach this general problem vary from using attribute-specific gazetteers to applying sequence labeling for named entity recognition, as well as applying ideas from search and question answering (Ghani et al., 2006; More, 2016; Putthividhya and Hu, 2011; Xu et al., 2019; Wang et al., 2020). The lot identification task could be modeled as the extraction of a binary attribute. One of the main studies in the area mentions "package quantity" as an example attribute (More, 2016), but does not further explore its extraction.

Related areas of study in the commerce literature are "bundling" (Adams and Yellen, 1976; Hanson and Martin, 1990; Yadav, 1994; Tzaban et al., 2020), tying together multiple distinct products, and "price packs" (Kwok and Uncles, 2005; Tellis, 1998), which are monetary promotions that offer savings by combining multiple items.

2.2 Text Representation and Classification

The literature on representation and classification of text data spans many disciplines and several decades. For a recent general survey on text classification the reader is referred to (Kowsari et al., 2019). Specific applications of these methods include document retrieval (Schütze et al., 2008), document categorization (Sebastiani, 2002) , question answering (Rajpurkar et al., 2018), and sentiment analysis (et al., 2002). The research area of *text representation* is devoted to methods for encoding a passage of text data in a machine-interpretable way. Most methods involve tokenization (Manning et al., 2014), breaking up a document into a collection of substrings, often corresponding to the words or word combinations in the document.

Word embedding has been an area of study that produces models, such as word2vec (Le and Mikolov, 2014; Mikolov et al., 2013), which include a distributed representation of words as part of their learned output. Other popular embedding models include dependency-based embedding (Levy and Goldberg, 2014) and GloVe (Pennington et al., 2014).

Language Modeling considers the problem of predicting unseen texts from context. Early language models were based on word and n-gram frequency (Jelinek and Mercer, 1980; Katz, 1987). Neural language modeling (Bengio et al., 2000) uses fully connected neural nets to predict the next word in a sentence. Other works propose models that use word-embedding resemblance in a similar setup (Le and Mikolov, 2014; Mikolov et al., 2013). Language models applying recurrent neural architectures are proposed in (Graves, 2013) (RNN) and (Merity et al., 2018a) (LSTM). These approaches also learn word embeddings as a component of their network architecture. A more recent architecture for language modeling that has gained much popularity is the transformer (Vaswani et al., 2017),

which uses neural attention mechanism instead of recurrence to encode the relevant context. BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is a variant of a transformer, which allows bi-directional training by masking random words in the training set, rather than trying to predict the next word in a sequence.

The upper layer output of language models, such as transformers and recurrent networks, can be used as a sentence embedding. This leads to the idea of *fine-tuning* (Howard and Ruder, 2018). The idea is to train large models in domains where a large volume of text data exists (e.g. Wikipedia). The parameters in the lower layers of the resulting models are then frozen and the upper layers are trained in a specialized domain where only a small amount of data is available (e.g., travel tips). The resulting model yields a useful representation of text in the specialized domain, without having to collect vast amounts of data.

3 Datasets

Recall from our discussion in the introduction, many lot offerings are not explicitly designated as lots by the seller, and thus this explicit signal cannot be relied upon. As such, we employed human agents to manually label offerings sampled at random.

We collected a number of examples of lot and non-lot offering titles from across several categories on eBay, one of the world's largest online marketplaces. The label was acquired by allowing a human evaluator to look at the entire offering page, which includes the title, but also additional structured information on the offering attributes and possibly an image and description. The evaluator then provided the lot-size label for each offering under consideration.

3.1 Lot Dataset

Table 2 describes the datasets used in our experiments. Each dataset is named for the category of e-commerce from which the offering title examples are taken. The *Heterogeneous* dataset is the largest dataset and contains examples from a number of different e-commerce categories. For the categories we considered, the lot class was severely underrepresented in the labeled data. Thus, we create a balanced evaluation set, containing roughly equal numbers of lot and non-lot offerings.[4]

Table 2: Datasets

Category	Training Size
Health & Beauty	4,370 (40.7% Lots)
Business & Industrial	1,754 (38.8% Lots)
Heterogeneous	18,742 (14.5% Lots)

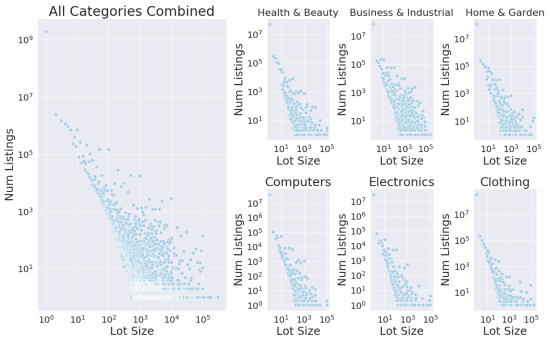

Figure 2: Distribution of lot size across categories. The distribution displays a classic power-law behavior across all categories (note the log scale in the axes.)

3.2 Lot Characteristics

We present additional empirical analysis of lot offerings in e-commerce with the hope of providing additional insight into their properties. To this end, we used the offerings explicitly designated as lots by the seller. While this is a noisier signal, it allows us to analyze many millions of offerings.

Figure 2 displays the the lot-size distribution of these same offerings, across several categories. The figure shows that the majority of lot offerings have a small lot size. In fact, the distribution displays a classic power-law behavior (note the log scale in the axes.)

To provide a sense of how titles of lot offerings differ from title of non-lot offerings, we set out to explore the most characterizing terms of lot titles versus non-lot titles. To this end, we used Kullback-Leibler (KL) divergence, which is a non-symmetric distance measure between two given distributions (Berger and Lafferty, 1999). Specifically, we calculated the unigrams and bigrams that contribute the most to the KL divergence between the language model of the lot titles versus the language model of the non-lot titles in our dataset. Table 3 presents the results. It can be seen that the

[4]The heterogeneous dataset will be publicly shared if the paper is accepted; a sample of the data is available at https://git.io/JJQCb

Table 3: Most distinctive unigrams and bigrams according to KL divergence over a sample of 4 million lot versus 4 Million non-lot titles. xxnum is a special token added by the tokenizer ahead of any numeric quantity. In addition, the portion of lot titles containing the unigram/bigram and the non-lot titles containing the unigram/bigram are presented.

Unigram	%lot	%non-lot	Bigram	%lot	%non-lot
lot	35.30%	2.73%	of xxnum	25.39%	2.18%
of	31.62%	7.32%	lot of	21.23%	1.27%
xxnum	90.04%	66.30%	lot xxnum	4.21%	0.38%
pcs	8.51%	1.21%	" lot	3.09%	0.16%
x	15.00%	5.48%	pack of	2.64%	0.16%
pack	5.80%	0.98%	- pcs	1.38%	0.01%
)	14.86%	8.93%	(pack	1.97%	0.05%
&	11.25%	6.14%	(xxnum	7.67%	3.20%
(	14.79%	9.37%	- xxnum	15.54%	10.48%
set	7.19%	3.44%	set of	3.13%	0.71%

top unigrams and bigrams represent diverse language, with very substantial differences between their occurrence in lot versus non-lot titles. As we will later show, due to this diverse language, a rule-based approach using regular expressions is not effective enough, and more advanced supervised approaches are required.

4 Methods

In this section, we formalize our research problem, and propose an approach for identifying lot offerings and lot size. In order to conserve space and focus the discussion, we defer the details of our novel tokenization (Section A.1) , training procedure (Section A.2), and model architectures (Section A.3) to the appendix.

4.1 Problem Definition

We formalize two variants of the lot classification task. Both accept only the offering title as input and are distinguished by their output.

1. *Binary Classification* – the decision function determines whether the title represents a lot offering or not. That is, are multiple identical items for sale in this offering?

2. *Lot Size Prediction* - in this, more challenging, formulation, the decision function outputs the *lot size*, the number of identical products for sale in the offering described by the title. This is a generalization of the first formulation, as non-lot offerings will have a lot-size of one.

Table 4: Binary Accuracy across datasets. * indicates statistical significance at 0.05 level.

	Health&Beauty	Business&Industrial	Heterogeneous
`RegExp_FC`	0.600	0.696	0.544
`NGram_FC`	0.843	0.845	0.815
`FastText_FC`	0.845	0.861	0.785
`LSTM_Basic_SZ`	0.889	0.881	0.872
`ENC_LSTM_BIN`	0.889	**0.928**	0.917
`ENC_LSTM_SZ`	0.915	0.897	0.898
`TRANS_ENC_SZ`	**0.944***	**0.933**	**0.945***

4.2 Identifying Lot Offerings

The problem definition above suggests using natural language processing techniques that given offering titles would output either the classification or the lot size prediction. However, in this work we propose several innovations specifically tailored to the problem of identifying lot offerings.

4.2.1 Lot size prediction as sequence labeling

While the lot size prediction problem is ostensibly a regression problem in that its output is a quantity, lot sizes are positive (more accurately ≥ 2), integer-valued, and distributed across a wide range of possible values (see Figure 2). Further, our defined business objective is exact lot-size accuracy. That is, an error in predicted lot-size of magnitude 1 should have equal cost to an error of magnitude 100 (which is very different from common regression objectives like squared error). We also note that, the lot-size information is very often present in the offer title exactly, and can (often) be made to be contained in a single token with sufficiently clever tokenization (see above).

For these reasons, rather than formalize the lot-size prediction problem as a naïve regression, with continuous output, we propose formalizing the approach as a sequence labeling problem. That is, the model output is a sequence of binary predictions. Each decision in the output sequence corresponds to a token in the input sequence, and encodes the probability that the corresponding token describes the lot size of the offering. Note that this objective is different from the eventual goal we measure in our experiments of predicting the lot size. We describe how to convert a per-token binary prediction to a lot-size prediction in Section A.3.2.

5 Experiments and Results

In our empirical evaluation, we examine the performance of the various model architectures described in Section A.3 on the lot classification problem vari-

Table 5: Lot Size Accuracy across datasets. * indicates statistical significance at 0.05 level.

	Health & Beauty	Business & Industrial	Heterogeneous
LSTM_Basic_SZ	0.870	0.820	0.845
ENC_LSTM_SZ	0.905	0.840	0.874
TRANS_ENC_SZ	**0.932***	**0.892***	**0.922***

ants defined in Section 4.1: Binary Classification and Lot Size Prediction.

To this end, we consider the following metrics:

1. Binary Accuracy (**BAcc**) - The number of times a title was classified Lot/ Not Lot correctly as a fraction of the evaluation set.

2. Lot Size Accuracy (**LAcc**) - The number of times the lot size was predicted (exactly) correctly as a fraction of the evaluation set.

When considering Binary Accuracy, we evaluate the Binary Classification Model architectures as well as the Binary Sequence Model architectures, which, as previously described, can be post-processed in a straightforward manner to yield a binary classification decision. We further evaluate the accuracy of the binary sequence model architectures in the Lot Size Prediction Problem. Recall that we approach this problem as a token classification problem. The token with the highest score is parsed for a numeric quantity, and this quantity is considered the predicted lot size. In this formulation, small errors in the lot size prediction are weighted equally to large errors. We computed statistical significance using a two-proportion z-test (Sprinthall and Fisk, 1990), with a significance level of 0.05.

Table 4 examines binary accuracy of the various models across the datasets, while Table 5 examines the lot size accuracy of the relevant models.

Examining Tables 4 and 5, we observe that the TRANS_ENC_SZ model achieves the best performance across all datasets. This may be because for this family of tasks, the transformer encoder architecture, which only considers word ordering indirectly, is more appropriate than the recurrent encoder architecture (ENC_LSTM_SZ), which explicitly models the word ordering. In other words, local word structure is more important than global word structure for this class of problem.

Furthermore, the results indicate that the pre-trained class of models (TRANS_ENC_SZ, ENC_LSTM_BIN, ENC_LSTM_SZ) leverage their indirect access to much larger general-purpose datasets to achieve better performance than models that were trained "from scratch" with random weight initialization. We can also observe that modeling the binary task directly does not improve binary performance and in fact, 3 of the top 4 performers on the Lot Classification task are sequence models, whose output is post-processed to reach a binary decision. This indicates that the value of the additional information (the lot size) and structure used during training the sequence model outweighs the cost of additional complexity incurred by expanding the decision space.

Another, somewhat surprising, result is that lot size prediction accuracy for all sequence models is quite close in magnitude to the binary classification accuracy (e.g., 0.932 compared to 0.944 for the *Health and Beauty* dataset and TRANS_ENC_SZ model). Thus, the models are able to predict the precise lot size correctly almost exactly as well as they are able to classify the offer as *Lot or Not*.

Generalizing from the results a bit, we observe that a large improvement is gained by modeling all n-grams (NGram_FC) and/or subwords (FastText_FC) over a simple collection of heuristic features (RegExp_FC). A smaller additional gain is made by using a recurrent architecture to explicitly model the temporal dynamics of the offer title (LSTM_Basic_SZ). Finally, an additional gain is achieved by introducing pre-trained high-capacity encoder architectures (ENC_LSTM_SZ, TRANS_ENC_SZ).

5.1 Complexity vs Accuracy

An additional analysis we carried out considers the complexity–accuracy tradeoff that exists in the models we considered. The reader of Section A.3 will no doubt observe that some of the architectures are significantly more complex than others. The more complex models generally achieve better quantitative performance in our empirical evaluation. However, how much of this complexity is needed is an important practical question, as often very complicated models are difficult to deploy and maintain in a production environment. In such cases, if a simpler model only slightly underperforms the more complicated model, in many cases it is preferred. To quantify this question of "bang for the buck" we plot the **BAcc** metric of experiments with different architectures against the "com-

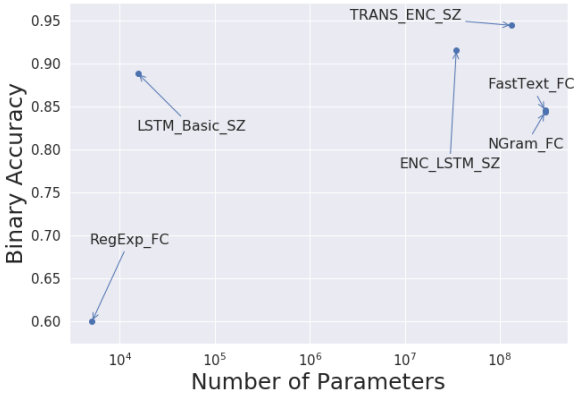

Figure 3: Comparing model complexity and binary accuracy on the *Health and Beauty* dataset.

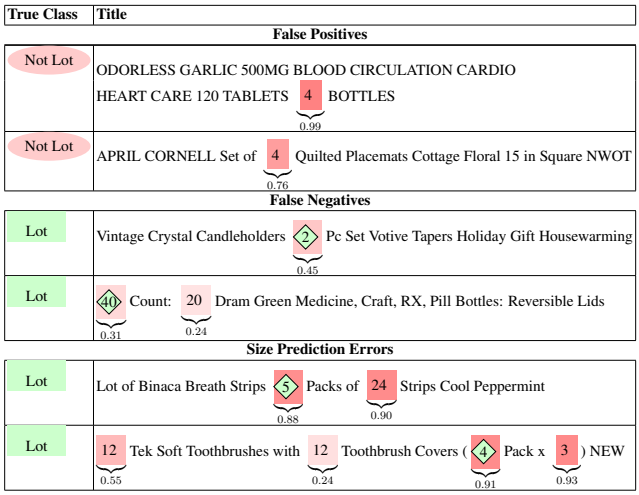

Figure 4: Error Analysis. ◇ indicates the lot-size token for each *Lot* title. ■ indicates the most likely lot-size tokens according to the model. The model score associated with each token is indicated below the token (when non-negligible).

plexity" of the method as measured by the number of learnable parameters in the architecture.[5] Figure 3 shows a plot of this tradeoff across several different training runs of different architectures on the *Health & Beauty* dataset (similar relative performance is observed on other datasets.)

The plot accentuates the benefit of the relatively simple LSTM architecture, `LSTM_Basic_SZ`. This architecture, while among those with the least learnable parameters, achieves performance (on both the binary and size prediction tasks) within 10% of the leading approach, while performing better than several other approaches with more parameters. Further, this architecture is much more flexible to augmentation as it does not rely on any pre-training. Thus, for a practical production scenario, which emphasizes "bang for buck", `LSTM_Basic_SZ` may be preferable to the other more complex alternatives.

5.2 Error Analysis

To gain additional insights into the performance, we present examples where the model disagrees with the ground truth labels in Figure 4. We focus on three types of such disagreements. "*false positive*" disagreements occur when an offering title is labeled as a lot incorrectly by the model. Examining the top rows of the figure, we observe that these types of mistakes often occur on titles that include phrases and language often associated with lot offerings. In many cases, using only the title information, a human evaluator may tend to agree with the model. Thus, we conjecture that these kinds of mistakes are largely due to the gap between the information available to the model and information available to the human labelers (which includes multiple modalities such as offering image, description, and more).

"*False negative*" mistakes occur when the model incorrectly labels an offering title as "not a lot". As Figure 4 demonstrates, the model often detects the "lot-size token" with non-negligible probability. However, this probability does not rise above the threshold needed to classify the offer as a lot. We used a threshold of 0.5 for this purpose, but this hyper-parameter can be tuned lower in order to correctly classify the examples in the figure. This type of tuning represents an opportunity to trade off false positive errors for false negative errors, as appropriate for the particular business scenario.

The third type of mistake we consider is "*size prediction error*". This type of error occurs when the model correctly identified an offering as a lot, but gets the lot size wrong. Examining the figure we can observe that this type of error occurs when the offer title is very complex, and specifically contains many numbers. It may be possible to detect this situation by considering the relative scores of different tokens.

We present the different types of errors for analysis, however, one should note that the different types do not occur with the same frequency. In our evaluation, false negative errors were more common then the other error types. Specifically, in the

[5] This method is not without faults, e.g., fastText uses a hashmap of 1M vectors to represent all possible sub-words, so technically has 300 (embedding size) times 1M learnable parameters, even though much fewer are updated in practice during training.

Table 6: Lot size prediction accuracy over the heterogeneous dataset across different architecture depths and tokenization methods. Boldfaced results are statistically tied best models at significance level of 0.05.

Number of layers	Simple Tokenization	BPE tokenization
6	**0.937**	0.901
12	**0.930**	0.918
24	**0.928**	0.909

Heterogeneous test set, the top performing model had 36 false positive, 71 false negatives, and 7 size prediction errors (out of 2,082 test examples).

5.3 Impact of Tokenization

Tables 4 and 5 show that `TRANS_ENC_SZ` outperforms all baseline architectures over all datasets. In additional experiments (not described), we observed that this architecture also outperforms the original BERT model (Devlin et al., 2019) pretrained on orders of magnitude more documents. One reason for this performance gap is the difference in tokenization. `TRANS_ENC_SZ` uses the custom tokenization described in Section A.1, while the original BERT tokenization is based on a trainable WordPiece tokenizer (Al., 2016), which uses sub-word level tokens. However, a confounding factor could be that the corpus used to train our model, a collection of 10 Million English language e-commerce titles (about 150M words), is more appropriate for our task than BERT's corpus of general natural language ($\sim$ 3B words).

To isolate the impact of the choice of tokenizer, we pre-trained language models with different tokenization variants: 1) the "Simple" tokenization is a plain rule-based tokenization that splits on punctuation and white spaces (see Section A.1); 2) the "BPE" tokenization is a "BERT-style" byte-pair-encoding scheme that tokenizes text into sub-word tokens based on the frequency statistics of bytes in a corpus. We also hypothesized that the depth (number of layers) of the language model is related to the performance of a tokenization approach. To evaluate this hypothesis we varied the number of layers along with the tokenizer.

Table 6 shows the result of this comparison for lot size prediction accuracy on the *Heterogeneous* dataset. Simple tokenization outperforms BPE tokenization by statistically significant margins. Notably, the depth of the transformer language model does not play a role, with all network depths achieving similar performance.

We conjecture that the reason for this performance increase is that sub-word tokenization is inappropriate for the lot classification task (at least for English text), as the important tokens are usually discovered by simple rules, and complex tokenization schemes, such as BPE, without this foreknowledge of the application, can potentially break an important "lot-size token" into multiple tokens, making a successful lot-size prediction impossible.

The table also shows that a "shallow" 6-layer transformer with Simple tokenization can perform just as well as much deeper models for this task. This combination is also more efficient computationally, due to the fewer tokens and layers.

6 Conclusions and Future Work

In this work, we consider the task of identifying *lots*, e-commerce offerings that contain multiple identical items. This application has the potential to improve the online e-commerce experience for millions of users. In our experiments, we apply a number of state-of-the-art natural language processing approaches to analyze the offering titles. We show that binary sequence models, which are aimed at identifying the lot-size token within the title, are especially effective for achieving high accuracy on both Lot Classification and Prediction tasks across multiple e-commerce domains.

Our models reach high performance based on title only, which is an advantage since almost all offers contain a valid title (as opposed to image, description, or key-value attributes). That said, the ability to detect lot offerings can potentially be further improved by using additional signals available for each offering beyond its title. The offer's price may also help achieve a further performance gain.

The methods developed herein rely on the availability of data to perform effectively. A large amount of unlabeled domain title data is necessary to build the language model, and a smaller amount of labeled data is required to fine-tune the model to the lot identification task. In other areas, where such data is available, specifically e-commerce data in languages other than English, we conjecture that this approach can generalize well.

Finally, the methods developed for analyzing titles in the pursuit of lot identification are useful in other problems that arise in the curation of a large and heterogeneous e-commerce catalog, including matching offerings to products and enabling product search.

References

William James Adams and Janet L Yellen. 1976. Commodity bundling and the burden of monopoly. *The quarterly journal of economics*, pages 475–498.

Yonghui Wu Et Al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2000. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proc. of SIGIR*, pages 222–229.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Bo Pang et al. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Thomas Wolf et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 8(1):41–48.

Anjan Goswami, Sung H Chung, Naren Chittar, and Atiq Islam. 2012. Assessing product image quality for online shopping. In *Image Quality and System Performance IX*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *ArXiv*, abs/1308.0850.

Ward Hanson and R Kipp Martin. 1990. Optimal bundle pricing. *Management Science*, 36(2):155–174.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8).

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the ACL*, Melbourne, Australia.

Fred Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In Edzard S. Gelsema and Laveen N. Kanal, editors, *Proceedings, Workshop on Pattern Recognition in Practice*, pages 381–397. North Holland, Amsterdam.

M. Jimenez, C. Maxime, Y. Le Traon, and M. Papadakis. 2018. On the impact of tokenizer and parameters on n-gram based code analysis. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*.

Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 35:400–401.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Simon Kwok and Mark Uncles. 2005. Sales promotion effectiveness: the impact of consumer differences at an ethnic-group level. *Journal of Product & Brand Management*, 14(3).

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Johan Lindskog and Jessica Lundgren. 1996. Multipack - a growing packaging concept. an analysis of the market, the distributions/handling & cost.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Sgdr: Stochastic gradient descent with warm restarts. In *Proc. of ICLR*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proc. of ACL*, pages 55–60.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018a. Regularizing and optimizing lstm language models. In *International Conference on Learning Representations*.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018b. Regularizing and optimizing lstm language models. In *International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. *arXiv preprint*, abs/1608.04670.

Slava Novgorodov, Ido Guy, Guy Elad, and Kira Radinsky. 2019. Generating product descriptions from user reviews. In *Proc. of WWW*, pages 1354–1364.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Duangmanee Pew Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proc. of EMNLP*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*, volume 4.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Kashif Shah, Selcuk Kopru, and Jean David Ruvini. 2018. Neural network based extreme classification and similarity models for product matching. In *Proc. of NAACL:HLT*.

Leslie N. Smith. 2017. Cyclical learning rates for training neural networks. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Richard C Sprinthall and Stephen T Fisk. 1990. *Basic statistical analysis*. Prentice Hall Englewood Cliffs, NJ.

Gerard J Tellis. 1998. *Advertising and sales promotion strategy*. Prentice Hall.

Hen Tzaban, Ido Guy, Asnat Greenstein-Messica, Arnon Dagan, Lior Rokach, and Bracha Shapira. 2020. Product bundle identification using semi-supervised learning. In *Proc. of SIGIR*, page 791–800.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. *Learning to Extract Attribute Value from Product via Question Answering: A Multi-Task Approach*, page 47–55. Association for Computing Machinery, New York, NY, USA.

Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *Proc. of ACL*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.

Manjit S Yadav. 1994. How buyers evaluate product bundles: A model of anchoring and adjustment. *Journal of Consumer Research*, 21(2):342–353.

Joachim Zentes, Dirk Morschett, and Hanna Schramm-Klein. 2017. Pricing. In *Strategic Retail Management*, pages 279–306. Springer.

A Appendix

A.1 Tokenization

An important preprocessing step in many natural language processing approaches is tokenization, transforming the raw text input into an ordered sequence of discrete tokens (often mapped to a finite-size dictionary). The choice of tokenization method can have significant impact on results in the downstream task (Jimenez et al., 2018). Specifically, applying tokenization that is catered to the downstream task may improve the overall performance. We therefore devise a unique tokenization scheme tailored to our scenario, processing offering titles in a general e-commerce marketplace. These titles (see Table 1) contain their own set of rules and idiosyncrasies, and can be quite different than English natural language text. As such, using general-purpose English language tokenization may be less desirable.

To understand our tokenization approach consider the following example title corresponding to a lot offering:

```
BOBBIN WINDER TIRE (2pk) Brother PC8895 ...
```

Clearly the numerical tokens are important to separate, but as the example illustrates, many numbers appear in lot offering titles that have nothing to do with the lot size, usually model numbers or various specification quantities. Further, the lot

quantity often appears in important context, such as adjacent to specific punctuation (e.g., within parentheses) or close to one or more context tokens (e.g. `Lot of` or `pcs`). These may or may not be separated by a whitespace token.

To deal with these phenomena, we developed several unique approaches to tokenization. First, we separate all punctuation into its own token. Then, we separate tokens with a numeric **prefix** into two separate tokens. Finally, we add a special token to indicate a numeric quantity. Note that this special token is added only when another token contains just digit characters. Thus, the token `pc8895`, for example, would not trigger a special token. This token does not replace the original numeric quantity, but rather is added next to it. This strategy is designed to allow generalizing from patterns often seen in offering titles.

With the application of such principles, the title above may be tokenized as follows:

```
bobbin | winder | tire | ( |
xxnum | 2 | pk | ) | brother | pc8895
```

where `xxnum` is the special token adding numerical context. Note that, in the example, the token `pc8895` is not split, as it does not begin with a digit character.

A.2 Techniques for Training Lot Models

Before describing specific neural architectures, we discuss key techniques that we applied in training our models that helped achieve the performance reported in the experiments section. While not all techniques are applicable to all model architectures, they play a key role in allowing our models to be trained to high performance.

A.2.1 Dynamic Learning Rates

We use a stochastic gradient descent variant (specifically, Adam (Kingma and Ba, 2015)) to optimize the parameters of the model architectures considered in this paper. Following (Bengio, 2012) , we employ several techniques to determine good values for the learning-rate parameter. The first of these is using *differential learning rates*, i.e. each parameter layer has a different learning rate. Another innovation that yields improved results is working with cyclical learning rates (Smith, 2017) combined with "cycle restarts" (Loshchilov and Hutter, 2017). That is, at the beginning of each epoch the learning rate is relatively large and begins to decay with each update. Another technique we employ is an interactive approach to finding the base learning rate called *The Learning Rate Finder* (Smith, 2017), a technique in which several batches of training are run with increasing learning rate, until the training error begins to increase. The process described above for training our models is interactive, and based on the performance of the network on such metrics as training and validation error.

A.2.2 Pre-Training and Fine Tuning

A well-accepted practice for improving the performance of neural models for natural language is the use of pre-trained language models (Howard and Ruder, 2018). These models are often quite large in terms of the number of parameters they contain, as well as the amount of training data they were trained on. Given a supervised text classification task, especially one where the amount of labeled training data is limited, we can use the language model to generate a useful representation of the text. This is often achieved by "chopping off" the top layer of the language model and using the continuous values of the activations in the second-to-last layer as the representation of the text. Using this representation, which is assumed to encode universal properties of word tokens, allows the primary task to utilize significantly less data.

The technique known as *Fine Tuning* introduces another step in this process. Essentially, the pre-trained language model is used as a language model on an additional corpus of text data, usually more relevant to the task of interest than the original corpus the model was pre-trained on (which is general in nature). Once this step is complete, the fine-tuned language model is used as before for the primary supervised task. Fine-tuning is especially useful when a large corpus of task-specific unlabeled data is available alongside the (often small) task-specific labeled data.

A.3 Model Architectures

We evaluated a number of different model architectures for the two problems described in Section 4.1. Each of the architectures, applied some subset of the techniques described above. Table 7 specifies the precise correspondence between models and techniques applied.

The approaches can be divided into two logical groups (1) binary classification models, and (2) binary sequence models. These groups are named according to their output type. The former group

Table 7: Techniques used by our different models.

Model	Dynamic Learning Rate	Pre-Training	Fine Tuning	Tokenization
RegExp_FC	✓	✗	✗	✗
NGram_FC	✓	✗	✗	✓
FastText_FC	✓	✗	✗	✓
ENC_LSTM_BIN	✓	✓	✓	✓
LSTM_Basic_SZ	✓	✗	✗	✓
ENC_LSTM_SZ	✓	✓	✓	✓
TRANS_ENC_SZ	✓	✓	✗	✓

outputs a single quantity corresponding to the probability that a title corresponds to a lot offering. The latter group outputs multiple quantities, each corresponding to a token in the input sentence. An important note is that binary classification models can only be used for the binary Lot Classification task and cannot address the Lot Size Prediction task. On the other hand, binary sequence models can be used for both the Lot Size Prediction task and the binary Lot Classification task.

A.3.1 Binary Classification Models

1. `RegExp_FC` – this model corresponds to a naïve baseline for the Lot Classification task. We represented the text as a set of binary features. Each such feature corresponded to whether we were able to match with a regular expression designed to fit important patterns pertaining to lots in the offering titles. For example, one such regular expression was the following : `pack of \d+` (where \d+ represents one or more digit characters). We made use of 15 such regular expressions. We fed this representation into a fully-connected neural network with a single hidden layer (size 300).

2. `NGram_FC` – here we represented the text as a bag of n-grams (using n=2 or n=3). Each n-gram corresponds to a binary feature (does the n-gram appear in the title). Although the space of possible n-grams is very large, in practice only a small sub-set appears. However, in order to enable unseen n-grams and keep the model size consistent, we used a hashmap of size 1M to map between each n-gram and its corresponding feature. That is, potentially multiple n-grams will map to the same binary feature, although such collisions rarely occur in practice. For each title, only a few n-grams of the many possible will be active. Thus, a sparse vector of size 1M represents each title. This representation was fed into a fully-connected neural network with one hidden layer (of size 300). We applied the lot-specific tokenization and dynamic learning rate techniques when learning the parameters of this architecture.

3. `FastText_FC` – in this model, we represented each word token as a vector of size 300, which is computed as a sum of its sub-word embeddings, which are learned separately. A sub-word is essentially a sub-string that can be constructed by only considering a subset of the characters composing the token. Sub-word information can be useful for generalizing tokens with similar roots that appear in different forms (e.g. the tokens `lot` and `lots`). Since there are many possible sub-words, as above in the n-grams model, we used a hashtable of size 2M to keep the model size fixed and allow generalization to sub-words that are unseen during the training phase. Each title is represented as a simple average of its word tokens. This representation was then processed by a fully connected linear layer. The architecture is equivalent to the fastText approach described in (Bojanowski et al., 2017) , although we used our own tokenization and training procedure.

4. `ENC_LSTM_BIN` – in this approach, we employed an LSTM-based encoder (specifically we employed the bi-directional multi-layered architecture described in (Merity et al., 2018b)), which yields a representation of the text using the sequence information explicitly. This approach uses pre-training a language model on a large corpus of text (specifically, the *WikiText 103* (Merity et al., 2016) dataset of English text) and then fine-tuning the learned representation on available e-commerce offering title data (not necessarily those offerings with known lot labels). We then attached a linear layer to the final layer of this architecture (which is a concatenation of the representation at each token), and trained the model on the available supervised data, to obtain the final binary classification model. We applied our own tokenization of the text before pre-training. During training, we made use of dynamic learning rate techniques described in Section A.2.1.

A.3.2 Binary Sequence Models

As discussed in Section 4.2.1, we address the lot size prediction problem with models that output a

sequence of binary decisions (one for each token in the input). To obtain the final prediction from such output, we apply the heuristic of choosing the maximum output value (assuming it passes some threshold) in the sequence and parsing the corresponding input token for a quantity. If no such token exists then the title does not represent a lot offering (and the predicted lot size is 1).

1. `LSTM_Basic_SZ` – in this approach, we used a basic LSTM model (Hochreiter and Schmidhuber, 1997), which takes into account the token ordering. The LSTM learns its own embedding for each word token. The final state vector is processed by a linear layer that outputs a binary decision per token. This method makes use of our custom tokenization (Section A.1) and dynamic learning rate (Section A.2.1).

2. `ENC_LSTM_SZ` – in this approach, we used the same encoder architecture as described for `ENC_LSTM_BIN` above. That is, we applied custom tokenization, pre-trained the encoder component of the model on a large corpus of general English text, and then fine-tuned using in-domain text data. However, instead of a binary classification head, this architecture attaches a binary sequence head on top of the encoder, which provides a binary decision for each of the tokens in the sequence.

3. `TRANS_ENC_SZ` – in this approach, we used the well-known BERT (Devlin et al., 2019) transformer architecture, and specifically its RoBERTa variant(Liu et al., 2019). The innovation of BERT over classical transformers is the combination of multiple self-supervision tasks, Masked Language Model and Next Sentence Prediction when training the encoder. The version we made use of is consistent with the common "base" architecture of BERT (et al., 2019), which is composed of a 12-layer encoder with 768 hidden nodes and 12 attention heads per layer, for a total of approx 132 million parameters. The model uses our custom tokenization scheme, which we believe is more appropriate for our research problem. Our model is pre-trained on 10 million English language e-commerce offering titles. As the pre-training is done on in-domain data, no additional fine-tuning step was performed.