Weakly supervised hierarchical multi-task classification of customer questions

Jitenkumar Rana
Amazon
jitenkra@amazon.com

Promod Yenigalla
Amazon
promy@amazon.com

Chetan Aggarwal
Amazon
caggar@amazon.com

Sandeep Mukku
Amazon
smukku@amazon.com

Manan Soni
Amazon
sonmanav@amazon.com

Rashmi Patange
Amazon
rpatang@amazon.com

Abstract

Identifying granular and actionable topics from customer questions (CQ) posted on e-commerce websites helps surface the missing information on the product detail page expected by customers before making a purchase. Insights on missing information on product page helps brands and sellers enrich the catalog quality to improve the overall customer experience (CX). In this paper, we propose a weakly supervised Hierarchical Multi-task Classification Framework (HMCF) to identify topics from customer questions at various granularities. Complexity lies in creating a list of granular topics (taxonomy) for thousands of product categories and building a scalable classification system. To this end, we introduce a clustering based Taxonomy Creation and Data Labeling (TCDL) module for creating taxonomy and labelled data with minimal supervision. Using the TCDL module, taxonomy and labelled data creation effort by subject matter expert reduces to 2 hours as compared to 2 weeks . For classification, we propose a two level HMCF that performs multi-class classification to identify coarse level-1 topic and leverages NLI based label-aware approach to identify granular level-2 topic. We showcase that HMCF (based on BERT and NLI) a) achieves an absolute improvement of 13% in Top-1 accuracy over single-task non-hierarchical baselines b) learns a generic domain invariant function that can adapt to a constantly evolving taxonomy (open label set) without need of re-training. c) reduces model deployment efforts significantly since it needs only one model that caters to thousands of product categories.

1 Introduction

Having correct, complete, and consistent information on the detail page is very important to ensure a world-class customer experience. E-commerce customers often refer to the "Customer Questions and Answers" (CQA) section to seek the information that they deem important before buying. World wide, a leading e-commerce website customers ask and questions in the order of millions[1] per week before buying. Customers refer to the CQA section primarily to find information that is a) not present on product page or b) is inconsistent across various sections of the product page. Therefore, identifying information gaps on product page can help catalog owners improve the quality of catalog, create new attributes to enrich product page. It also helps product owners design better products, understand customer preferences, and improve the overall customer experience.

In this paper, we aim to build a scalable solution that extracts topics from customer questions (CQ). *Note:* In this problem, we want to extract topics customers are interested in and not the answers to their questions since the objective is to identify information gap on product page using CQs. Further, we wish to identify granular and "actionable" topics. An actionable topic clearly conveys the intent behind the question. Please refer to Table 5 for topic action-ability examples.

Identifying topics for CQ poses several practical challenges, especially, at the scale of an e-commerce giant where products are spread across thousands of categories.

Constantly evolving taxonomy (open label set): We wish to extract diverse topics from various product categories. The topics are dynamic and keep evolving over time because new products keep launching. Hence, traditional text classification approaches are not applicable for our use case since they work with a fixed and limited set of labels. Further, we usually don't have a pre-defined taxonomy to start with, and we need to define a separate taxonomy for each product category for two reasons: a) we observe that CQs are generally very specific to product categories b) we also observe that granular topic overlap amongst similar or re-

[1] We are not revealing exact numbers to comply with company legal policy.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 786–793
July 10-12, 2023 ©2023 Association for Computational Linguistics

Customer question	Topic	Action-ability	Actions
What is seat height?	size	×	can not take action
Does this sofa set include ottoman too?	pack content	×	can not take action
Can I place lounger on right of the sofa?	usage	×	can not take action
What is seat height?	seat height	✓	Update "size chart"
Does this sofa set include ottoman too?	includes ottoman	✓	Add pack content info in bullet points
Can I place lounger on right of the sofa?	right-left placement	✓	Update placement images

Table 1: Examples of action-ability of topics

lated product categories such as Table, Chair, Sofa is less than 30%. However, it is complex to manually create taxonomies across thousands of product categories from scratch.

Labelled data scarcity: Next, it is infeasible to obtain a large amount of manually annotated dataset for thousands of product categories typically required for training deep learning models.

High cardinality of label space: We observe that there are hundreds of granular topics per product category in our datasets. We observe (in Table 3) that performance of metric learning or softmax based multi-class classifier degrades with such a high number of classes.

To tackle the challenges mentioned above, we introduce two main novel ideas. We introduce a clustering based Taxonomy Creation and Data Labeling (TCDL) module to create taxonomy and labelled data efficiently with very minimal manual supervision. Using this module, taxonomy creation effort by a subject matter expert reduces to 2 hours as compared to 2 weeks.

For topic identification, we introduce a novel Hierarchical Multi-task Classification Framework (HMCF), which performs multi-class classification to predict level-1 topic and leverages Natural Language Inference (NLI) to identify granular level-2 topic from question. We show that a model trained using NLI based HMCF a) achieves an absolute improvement of 13% Top-1 accuracy over a single-task non-hierarchical architecture baseline, and b) learns a generic function that can adapt to new product categories and topics with high accuracy without retraining.

Rest of the paper is organized as follows. We discuss related work in section 2. In section 3, we explain TCDL module in detail. We discuss HMCF in detail in section 4. Following that, we discuss experiments and results in section 5 and section 6, respectively. Finally, we conclude the paper with our findings in section 7.

2 Related work

The lack of availability of training data and a pre-defined taxonomy is a big challenge in industry. LDA ([Blei et al., 2003]) can be helpful in mining topics from corpus of text and creating taxonomy. Taxogen ([Zhang et al., 2018]) proposed an unsupervised method to derive taxonomy from a large corpus of text. However, these approaches model the text corpus as Bag of Words (BoW) and do not take into account the context of the keywords. There are many models available that model multi-class text classification as a hierarchical classification problem [Tsochantaridis et al., 2005, Sinha et al., 2018]. However, they do not offer flexibility in adding or removing labels without adding extra parameters and retraining on entire dataset. Also, they use deep learning based models which require large amounts of labelled data for training. Recently, few-shot learning based approaches have gained popularity in the NLP domain, particularly to address the challenge of large scale label data availability ([Nichol et al., 2018], [Snell et al., 2017], [Zhang et al., 2020]). However, these approaches do not take label information into account. Recently, BERT ([Devlin et al., 2018]) has shown state-of-the-art performance on many NLP tasks. We use it for both level-1 and level-2 tasks.

3 Weakly supervised taxonomy and labelled data creation

In this section, we describe the taxonomy and label creation approach in detail (refer to Figure 1).

3.1 Hierarchical taxonomy

We organize topics into a two level hierarchy for following reasons: a) using hierarchical taxonomy, sellers can consume insights at two different levels of granularity b) topic classification model performance improves with hierarchical taxonomy as compared to flat taxonomy (section 6).

Level-1 in taxonomy contains total of 9 generic

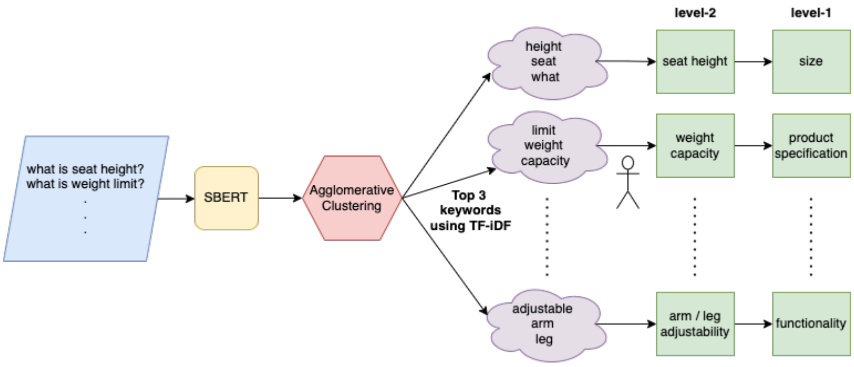

Figure 1: Illustration of taxonomy creation and data labelling module

topics such as "size", "health and safety", etc. They are common for all the product categories. Whereas, level-2 topics are granular and different for each product category. For example, some of the "size" related level-2 topics for "CHAIR" product categories are "arm height", "seat depth" whereas the same for "TABLE" product categories are "table top dimensions", "drawer dimensions". Please refer to Table 6 for more examples.

3.2 Taxonomy creation and data labeling module (TCDL)

We propose a scalable two step approach for defining taxonomy. The steps are: a) cluster CQs b) assign names to each clusters. First, we cluster CQs using their Sentence-BERT embedding [Reimers and Gurevych, 2019]. We perform "agglomerative clustering" primarily because we don't know the number of clusters beforehand. We choose "cosine similarity" with "average" linkage as a similarity criterion and a threshold of 0.2 for merging the clusters. We observe that the clusters obtained using the criteria mentioned are coherent and granular.

To assign a topic name to each cluster, we take the top k keywords for each cluster based on their TF-IDF scores. Then, with minimal manual supervision, we can derive granular topic names with guidance from top keywords for each cluster. We organize all the granular topic names at level-2 and manually map it to an appropriate level-1 topic. With this approach, it requires only $\sim 1 - 2$ hours of manual supervision as opposed to 2 weeks to come up with taxonomy per product category.

At the end of taxonomy creation step, we have a hierarchical taxonomy and level-1, level-2 labels for each cluster. Cluster labels serve as level-1 and level-2 labels for each question within the cluster. This way, labelled data generation for the classification model is totally automated. Finally, to obtain product category p specific taxonomy T^p, a granular topic t is added to T^p only if there is a question q from p with label t present in the TCDL output. Please refer to Table 7 for the output of the clustering and labeling.

4 Hierarchical Multi-task Classification Framework (HMCF)

4.1 Problem definition and formulation

Given a question q from the product category p, we want to map it to l_1 and l_2^p. Here, l_1 is a level-1 topic generic across all product categories. Whereas, l_2^p is a level-2 topic specific to product category p.

With HMCF described in the paper, we can map q to 1-class at each level because CQs typically talk about one topic. However, HMCF is generic and can be extended to multi-label classification at every level with minor modifications.

4.2 Framework details

Figure 2 shows the details of the proposed HMCF. We use BERT as the shared input encoder for both tasks. Individual task specific fully connected networks are added on top of BERT. [CLS] token embedding from BERT is used as input to each task specific network.

Level-1 topic identification problem is modelled as multi-class classification task since level-1 topics a) are high level, b) don't vary with time, and c) remain the same across product categories. Level-1 network is a fully connected network with 9 output nodes and softmax activation.

Level-2 classes change over time and are different across product categories. To handle such challenges, we leverage NLI based architecture for level-2 topic identification task. Given a question q

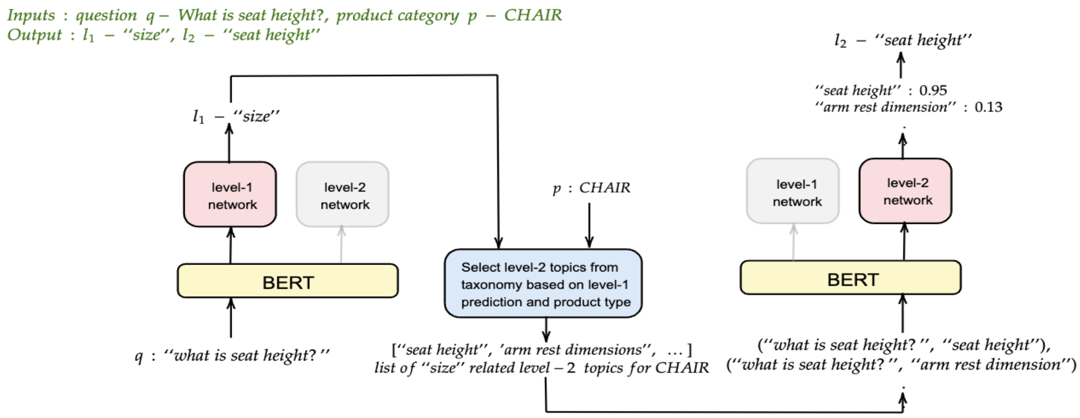
Inputs : question q – What is seat height?, product category p – CHAIR
Output : l1 – "size", l2 – "seat height"
l1 – "size"
level-1 network
level-2 network
BERT
q : "what is seat height?"
p : CHAIR
Select level-2 topics from taxonomy based on level-1 prediction and product type
["seat height", 'arm rest dimensions", …]
list of "size" related level – 2 topics for CHAIR
l2 – "seat height"
"seat height" : 0.95
"arm rest dimension" : 0.13
level-1 network
level-2 network
BERT
("what is seat height?", "seat height"),
("what is seat height?", "arm rest dimension")

Figure 2: Inference in Hierarchical Multi-task Classification Framework

and a granular topic t, the level-2 network predicts 1 if t is an appropriate topic for q and 0 otherwise. Level-2 network is again a fully connected network with one output node and sigmoid activation. We refer to the proposed architecture as BERT-HMCF-1-model-NLI throughout the rest of the paper.

4.3 Input preparation

We use the BERT tokenizer to prepare input for both tasks. Input for level-1 task is constructed as a concatenation of [CLS], τ_q, [SEP]. Whereas, input for level-2 task is constructed as a concatenation of [CLS], τ_q, [SEP], τ_t, [SEP]. Here, τ_q and τ_t are the lists of tokens of q and t obtained from the BERT tokenizer, respectively. Whereas, [CLS] and [SEP] are special tokens from the BERT vocabulary.

4.4 HMCF inference

In HMCF, inference happens in two stages (refer to equation[1]). At first, given a question q, level-1 network is used to predict high level topic l_1. Then, given product category p and l_1, level-2 network is used to predict p specific granular topic l_2^p that maps to l_1. To do so, q is paired with t_i^{p,l_1} where $t_i^{p,l_1} \in T^{p,l_1}$, the set of all p specific granular topics that maps to l_1. Then, BERT is used to encode (q, t_i^{p,l_1}) pairs, and level-2 network is used to obtain s_i^{p,l_1}. Here, s_i^{p,l_1} can be thought of as score of appropriateness for the (q, t_i^{p,l_1}) pair. Finally, l_2^p is chosen as the t_j^{p,l_1} for which s_j^{p,l_1} is the maximum.

In the equation 1, e^x_{CLS} is the [CLS] token embedding from BERT for input x, L is the set of all level-1 topics, M_1 and M_2 are the level-1 and level-2 task networks, respectively, l_1 is level-1 prediction, T^{p,l_1} is the set of all p specific level-2 topics that map to l_1, n^{p,l_1} is the number of topics in T^{p,l_1} and l_2^p is the final level-2 prediction.

$$
\begin{aligned}
e^q_{CLS} &= \mathrm{BERT(q)} \\
l_1 &= \mathrm{argmax}(\mathrm{M}_1(e^q_{CLS})) \quad \text{where } l_1 \in L \\
e^{q,t_i^{p,l_1}}_{CLS} &= \mathrm{BERT}(q, t_i^{p,l_1}) \quad \text{where } t_i^{p,l_1} \in T^{p,l_1} \\
S &= \left[s_i^{p,l_1}\right]_{i=0}^{i=n^{p,l_1}} \quad \text{where } s_i^{p,l_1} = \mathrm{M}_2(e^{q,t_i^{p,l_1}}_{CLS}) \\
l_2^p &= t_j^{p,l_1} \quad \text{where } j = \mathrm{argmax}(S)
\end{aligned} \tag{1}
$$

4.5 HMCF training

4.5.1 Training data preparation

To prepare training data, we use the output of the TCDL module that contains level-1 topic l_1, level-2 topic l_2^p and product category p for every question q. Training data for level-1 task can be obtained by setting l_1 as the label for question q. We need to create positive and negative (q, t) pairs for the level-2 task. Given q, p, l_1 and l_2^p, positive training samples are obtained by pairing q with l_2^p. We create easy negatives by pairing q with a randomly sampled level-2 topic from the set of all level-2 topics available. To generate hard negatives, we sample level-2 topic t' from T^{p,l_1} such that $t' \neq l_2^p$. Here, T^{p,l_1} is the set of all p specific level-2 topics that map to l_1.

4.5.2 Batch sharing for training

We train networks for both tasks simultaneously. However, each batch contains data only for one task. At every step of training, we randomly sample a task and a batch for the same task for optimization. We optimize task-1 network using standard cross-entropy loss and task-2 network using binary cross-entropy loss. If a batch corresponds to task k, then only task k specific parameters are optimized.

5 Experiments

5.1 Training setup and experiments

We conduct experiments to evaluate HMCF on various aspects such as a) hierarchical vs. non-hierarchical architecture; b) multi-task vs. single-task modelling; c) suitability of NLI tasks for constantly changing label space; d) zeroshot capabilities; and e) impact of level-1 task performance on overall performance.

We use a dataset of customer questions posted on an e-commerce website for our experiments. This dataset contains questions from 122 product categories. Taxonomy for all 122 product categories and training data is prepared using the TCDL module. BERT based HMCF and all the other baselines are trained only on the data from 100 product categories. We denote this dataset as CQ100PCTrain.

We use the remaining 22 product categories to test the few-shot capabilities of HMCF. To do so, we separately train models on CQ100PCTrain combined with only $k(= 5, 10, 20)$ samples per level-2 topic from the remaining 22 product categories.

To reduce the impact of level-1 errors on overall system performance, we also experiment with following inference strategy. We pair q with all the level-2 topics that map to any of the top-k ($k = 1, 2, 3, \ldots$) level-1 topics by output probabilities. Then, topic t with the highest output score for the (q, t) pair from level-2 model is chosen as the final level-2 prediction.

Since topic identification from CQs is a multi-class classification problem, we choose Top-1 accuracy as a performance criterion for both tasks. We measure the performance of the model on two separate datasets: a) CQ100PCTest and b) CQ22PCTest. CQ100PCTest is a full-shot dataset that contains 15000 manually annotated questions from same 100 product categories that are used in training. CQ22PCTest is a zero-shot dataset that contains 5000 manually annotated questions from the remaining 22 product categories that are not used for training. Refer to Table 2 for detailed data statistics.

We use pre-trained bert-base-uncased[2] as the base encoder to maintain parameter parity among every framework. We fine-tune all models on single Nvidia V100 GPU for 2 epochs using Adam optimizer, learning rate of $2e^{-5}$, batch size of 32.

Dataset	# questions	# unique level-2 labels
CQ100PCTrain	81,042	2,031
CQ100PCTest	15,000	2,031
CQ22PCTest	5,000	246

Table 2: Data statistics

5.2 Baselines

BERT-Softmax is a single-task, non-hierarchical BERT model with a linear layer and softmax activation on top of it. Output layer contains 2031 nodes (same as no. of level-2 topics in the dataset).
BERT-cos is a single-task, non-hierarchical model. In this model, we encode question q and topic t separately using BERT and obtain 64 dimensional projection using a learnable linear layer. CosineLoss [eq. 2] is used to train the network.

$$CosineLoss(x, y) = \begin{cases} 1 - cos(x, y), & \text{if } y = 1 \\ max(0, cos(x, y)) & \text{if } y = -1 \end{cases} \tag{2}$$

BERT-NLI is a single-task, non-hierarchical NLI model. It takes a question q and topic t as input. Expected output is 1 if t is an appropriate topic for q and 0 otherwise. During inference, we don't use level-1 information and pair q with all the level-2 topics for the product category p.
BERT-HCF-2-model-NLI differs from BERT-HMCF-1-model-NLI in the following aspects: HCF stands for hierarchical classification framework. For HCF, we train two different models for each task: one BERT model as a multi-class classifier for level-1 and another BERT model as an NLI based binary classification for level-2. Note that BERT-HCF-2-model-NLI is a hierarchical architecture with single-task models for each task.
BERT-HMCF-1-model-cos and BERT-HCF-2-model-cos are the architectures equivalent to BERT-HMCF-1-model-NLI and BERT-HCF-2-model-NLI, respectively. The only difference is that level-2 task is trained using CosineLoss between 64 dimensional linear projections of individual [CLS] embeddings of question and topic.

6 Results

In this section, we discuss the results of our experiments in detail. We use the Top-1 accuracy metric in all the experiments. Below is a summary of key observations made from the experiment results.

6.1 Key observations

We capture the detailed results of all experiments in Table 3 and make the following observations.

[2] https://huggingface.co/bert-base-uncased

			CQ100PCTest		CQ22PCTest	
		Architecture	level-1	level-2	zero-shot level-1	zero-shot level-2
Non-hierarchical	single-task	BERT-Softmax	-	0.59	-	-
		BERT-cos	-	0.59	-	0.58
		BERT-NLI	-	0.70	-	0.67
Hierarchical	single-task	BERT-HCF-cos-2-model	0.88	0.68	0.84	0.63
		BERT-HCF-NLI-2-model	0.89	0.79	0.83	0.75
	multi-task	BERT-HMCF-cos-1-model	0.9	0.69	0.86	0.59
		BERT-HMCF-NLI-1-model	**0.92**	**0.83**	**0.87**	**0.78**

Table 3: Comparison of the performance of various architectures

Hierarchical framework performs better than non-hierarchical framework. We observe from Table 3 that models trained in hierarchical framework tend to perform better than equivalent non-hierarchical models. For example, BERT-HMCF-1-model-NLI yields 13% and 11% absolute improvement in Top-1 accuracy (for level-2 task) over BERT-NLI on the CQ100PCTest and CQ22PCTest datasets, respectively. Superior performance of hierarchical framework can be attributed to the fact that a) level-1 prediction helps level-2 model narrow down focus on limited set of classes, leading to better performance b) models can be trained on hard negative examples under hierarchical framework because of the availability of level-1 information.

Multi-task model outperforms single-task model. We can see from Table 3 that, BERT-HMCF-1-model-NLI achieves a 4% absolute accuracy improvement over BERT-HCF-2-model-NLI for level-2 task on the CQ100PCTest dataset. This suggests that weight sharing between tasks is helpful in learning more general representations that are useful across both tasks.

NLI framework is suitable for level-2 identification. Again, we can see from Table 3 that NLI based architectures outperform equivalent cosine similarity based architectures. The primary reason for this finding can be attributed to the fact that [CLS] embeddings in NLI architecture are obtained by computing attention over both question and topic tokens together. Hence, the [CLS] embeddings obtained are rich in representation as compared to individual question and topic embeddings computed in a cosine similarity based framework.

NLI based architecture demonstrates excellent generalization capabilities. We observe from Table 4 that BERT-HMCF-1-model-NLI achieves 78% accuracy on the dataset of 22 product categories on which the model was not trained. Further, with just 10 samples per label, accuracy reaches 82% which is almost at par with full-shot model performance. We conclude that NLI architecture can be thought of as computing a general similarity metric between topic and question. Since this task is domain agnostic, the model can easily adapt to out-of-domain data.

k-shot setting	level-1	level-2
0-shot	0.87	0.78
5-shot	0.89	0.81
10-shot	0.89	0.82
20-shot	0.91	0.84

Table 4: BERT-HMCF-1-model-NLI performance on CQ22PCTest (22 product category dataset) in various few-shot scenarios

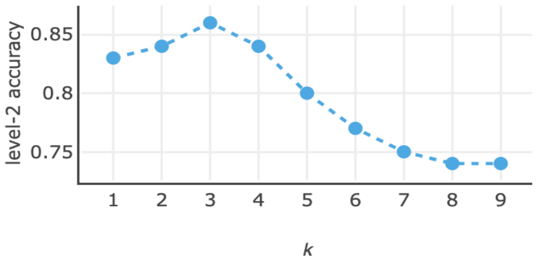

Figure 3: Level-2 accuracy when top-k level-1 predictions are considered for level-2 inference

Using Top-k level-1 predictions for level-2 inference reduces the impact of level-1 error on overall performance. Figure 3 demonstrates level-2 accuracy with respect to k, where k is the number of top-k level-1 predictions used for level-2 inference. Level-2 performance increases from $k = 1$ to $k = 3$ monotonically. The performance reduces from $k = 4$ onwards, mainly because the model has to differentiate between more topics as k increases.

7 Conclusion

In this paper, we present HMCF, a hierarchical multi-task classification framework to identify granular topics from customer questions. Through systematic studies, we showcase that NLI based HMCF is more appropriate for our problem as compared to single-task or non-hierarchical architectures and yields 13% absolute improvement in Top-1 accuracy over single-task non-hierarchical baselines. We also demonstrate that NLI based HMCF generalizes well on other domains since it learns a domain invariant topic and question similarity metric. We also propose a top-k level-1 predictions based inference strategy for level-2 task to reduce the impact of level-1 model errors on overall performance. Further, with the TCDL module proposed in the paper, taxonomy and labelled data creation efforts reduce significantly. We deployed single BERT-HMCF-1-model-NLI to production for 600 product categories and use it to provide actionable insights to the selling partners. Selling partners and brand owners make corrections to the product page or create new attributes to enrich the detail page. Our business teams consume the output to measure the impact of enrichment on product page using the purchase inquiry rate (PIR) metric, the % questions asked WoW.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3 (null):993–1022, mar 2003. ISSN 1532-4435.

Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian M. Sadler, Michelle Vanni, and Jiawei Han. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. *CoRR*, abs/1812.09551, 2018. URL http://arxiv.org/abs/1812.09551.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6: 1453–1484, 09 2005.

Koustuv Sinha, Yue Dong, Jackie Chi Kit Cheung, and Derek Ruths. A hierarchical neural attention-based text classifier. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 817–823, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1094. URL https://aclanthology.org/D18-1094.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018. URL https://arxiv.org/abs/1803.02999.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017. URL https://arxiv.org/abs/1703.05175.

Jian-Guo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference, 2020. URL https://arxiv.org/abs/2010.13009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

A Discussion on output of BERT-HMCF-NLI-1-model

We can observe from Table 5 that the level-2 topics predicted are very granular and actionable. We also did error analysis to understand error patterns. The major error contribution comes from errors made by level-1 model. If the level-1 model makes an error, the level-2 prediction will be incorrect with probability 1. We observe that errors made by level-1 contribute to $\sim$ 60% of overall errors. Next, we also observe that, $\sim$ 25% of the errors are contributed by questions with multiple intents (example 7,8 in Table 5). In such cases, even though the model predicts one correct topic, it misses predicting another topic, and we count it as an error. This error occurs due to an architectural limitation because our architecture only predicts one topic. There are very few cases where we observe that the model learns to put undue emphasis on certain keywords ("back" in topic and question, example 9 in Table 5) and gives the highest score to the wrong topic.

B Taxonomy and TCDL module output

id	Customer question	Level-1	Level-2	is level-2 correct?	correct topic
1	What is seat height?	size	seat height	yes	-
2	Does this sofa set include ottoman too?	pack/quantity	includes ottoman	yes	-
3	Can I use it outdoor?	usage	indoor or outdoor usage	yes	-
4	Can this chair serve as a study chair for children?	compatibility	suitable for studying	yes	-
5	does it rust?	usage	rustproof	yes	-
6	is this a set of 2?	pack/quantity	quantity	yes	-
7	What is the height? what is the width?	size	seat height	no	(“size”, “seat height”), (“size”, “seat width”)
8	Is the table green color? Also, what is the size of white table?	size	table dimensions	no	(“product specification”, “color”), (“size”, “table dimensions”)
9	what is the inscription on the back?	product specification	front/back specification	no	(“product specication”, “back inscription”)
10	can I place 10kg oven on this table?	usage	placement	no	(“product specification”, “weight limit”)

Table 5: Example of customer questions and BERT-HMCF-1-model-NLI output

Level-1	Level-2
size	[arm height, foot rest height, depth without cushion, . . .]
product specification	[color, reclining angle, rocking feature, . . .]
usage	[adjustable lumber, foldable, can be used outdoors, . . .]
material	[cushion material, leather type, arm material, . . .]
compatibility	[suitable for certain heights, suitable for kids, suitable for studying, . . .]

Table 6: Example of taxonomy for CHAIR category

Questions in a cluster	Top-3 keywords	Level-2 topic	Level-1 topic
what is the height of seat from the floor?, how high is the seat from the ground?, what is seat height?	seat, height, floor	seat height	size
Can this be used by men that weight 250pds...Top and bottom?, What is the weight limit?, what’s the weight capacity for this? I’m an adult of about 300 pounds., What is the weight limit? Can an adult use it?	weight, limit, capacity	weight limit	product specification
Can its height be adjusted?, How high can this adjust to?, Is it adjustable to height or is it one height? limit?, Does it have different adjustable hights and what is the lowest setting?, Is height adjustable	adjust, height, limit	height adjustability	usgae

Table 7: Example of output of taxonomy creation and data labeling module