A Regularised Intent Model for Discovering Multiple Intents in E-Commerce Tail Queries

Subhadeep Maji[1](✉), Priyank Patel[1], Bharat Thakarar[1], Mohit Kumar[2], and Krishna Azad Tripathi[1]

[1] Flipkart Internet Private Limited, Bangalore, India
{subhadeep.m,priyank.patel,bharat.thakarar,krishna.tripathi}@flipkart.com
[2] Udaan.com, Bangalore, India
mohitkum@udaan.com

Abstract. A substantial portion of the query volume for e-commerce search engines consists of infrequent queries and identifying user intent in such *tail* queries is critical in retrieving relevant products. The intent of a query is defined as a labelling of its tokens with the product attributes whose values are matched against the query tokens during retrieval. Tail queries in e-commerce search tend to have multiple correct attribute labels for their tokens due to multiple valid matches in the product catalog. In this paper, we propose a latent variable generative model along with a novel data dependent regularisation technique for identifying multiple intents in such queries. We demonstrate the superior performance of our proposed model against several strong baseline models on an editorially labelled data set as well as in a large scale online A/B experiment at Flipkart, a major Indian e-commerce company.

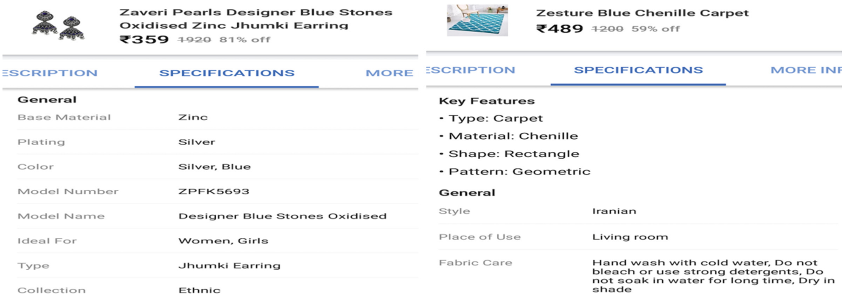
1 Introduction

E-commerce companies offer a wide selection of products from many categories and the number of unique queries submitted to their search engines can be of the order of millions per month. A substantial portion of these queries are infrequent; we observed that approximately 35% of the unique queries at Flipkart, a major Indian e-commerce company occur less than 50 times a month. Such *tail* queries [11,24] lack sufficient click-through data and tend to have poor retrieval performance [11,14,17]. Improving performance on these queries has a large business impact from the long term benefits of greater customer satisfaction [2,7].

E-commerce search is a faceted search on a structured catalog of products defined by a set of specifications represented as key-value pairs. Two products from the Jewellery and Home Furnishing categories at Flipkart are shown in Fig. 1 along with some of their specifications. Specifications like 'plating' and 'shape' are product attributes that take values 'silver' and 'rectangle' respectively. The intent of a search query is defined as a labelling of its tokens with the

M. Kumar—This work was done while the author was at Flipkart.

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12035, pp. 651–665, 2020.
https://doi.org/10.1007/978-3-030-45439-5_43



Zaveri Pearls Designer Blue Stones Oxidised Zinc Jhumki Earring
₹359 ~~1920~~ 81% off

General

Base Material	Zinc
Plating	Silver
Color	Silver, Blue
Model Number	ZPFK5693
Model Name	Designer Blue Stones Oxidised
Ideal For	Women, Girls
Type	Jhumki Earring
Collection	Ethnic

Zesture Blue Chenille Carpet
₹489 ~~1200~~ 59% off

Key Features

- Type: Carpet
- Material: Chenille
- Shape: Rectangle
- Pattern: Geometric

General

Style	Iranian
Place of Use	Living room
Fabric Care	Hand wash with cold water, Do not bleach or use strong detergents, Do not soak in water for long time, Dry in shade

Fig. 1. Specifications of products from Jewellery and Home Furnishing.

product attributes whose values are matched against the query tokens during retrieval. The intent of two search queries is illustrated in Table 1.

Queries in e-commerce search can have multiple correct intents due to multiple valid matches between their tokens and the values of product attributes. An example of this is shown in Table 1 where the attributes 'color', 'plating', and 'base material' are all correct labels for the token 'silver' in the query 'silver oxidised earring'. This phenomenon is particularly prevalent in tail queries; an analysis of an editorially labelled sample of tail queries at Flipkart revealed that approximately 42% of tail queries had multiple correct intents. Existing techniques for identifying user intent in search queries are either supervised [17,19] or semi-supervised [11,19] and require labelled or partially labelled queries. Extending them to identify multiple intents in tail queries is difficult due to a lack of sufficient click-through data from which labels can be derived [14,17,25]. We address this shortcoming of existing techniques in our current work.

We start with an empirical study of the product catalog and search query logs at Flipkart and base our current work on its conclusions. We propose a latent variable generative model for the observed ordered pairs of query tokens that has the corresponding ordered pairs of attribute labels as the latent variables. This addresses the lack of labelled data for tail queries. We observed that tail queries tend to have multiple intents due to multiple attributes having similar high

Table 1. Labellings of multi-intent queries 'silver oxidised earring' and 'rectangle room mat' by the baselines and our proposed model (RIM) which identifies all correct intents.

	silver	oxidised	earring	rectangle	room	mat
Correct intent	color, plating, material	model	store	shape, pattern	place-of-use	store
LR	color, store	model	store	type	type	store
CRF	color, store, model	model	store	store	place-of-use	store
Bi-LSTM-1	model	model	store	key-features	model	model
Bi-LSTM-2	color	model	store	key-features	place-of-use	store
UMM	color, store	model	store	type	place-of-use	store
RIM	color, plating, material	model	store	shape, pattern	place-of-use	store

empirical probabilities of generating the same tokens. We propose a similarity measure between attribute pairs and use it to regularize our model in a way that the learnt posterior distributions have similar probabilities for similar attribute pairs. This addresses the problem of identifying multiple intents in tail queries. We finally demonstrate the superior performance of our proposed model against several strong baselines on an editorially labelled data set and in a large scale online A/B experiment at Flipkart where we achieved statistically significant improvements of 3.03% in click-through rate and 15.45% in add-to-cart ratio.

2 Definitions and Preliminaries

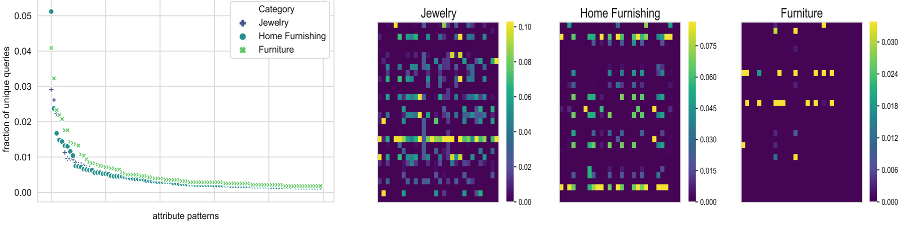
E-commerce product catalogs are typically divided into various categories where every product belongs to a single category. Examples of such categories are Jewellery, Furniture, and Home Furnishing (bed sheets, table covers, curtains, etc.). Sample products from Jewellery and Home Furnishing are shown in Fig. 1. We define *tail* queries as queries that occur less than 50 times a month.

The attributes that describe the products within a category are denoted by $\mathcal{A}$ and the values these attributes can take are denoted by $\mathcal{V}$. Every product can thus be represented by a set of attribute-value pairs (a, v) where v may consist of multiple tokens. For example, some of the values that the attribute 'material' can take in the Jewellery category are 'rose gold', 'silver', 'bronze', 'stainless steel', etc. The vocabulary of tokens that constitute all the attribute values is denoted by $\mathcal{W}$. A query is denoted by $\mathbf{x}$ and is defined as a sequence of n tokens $(x_1, x_2, \ldots, x_n)$. The intent of this query is denoted by $\mathbf{z}$ and is defined as a corresponding assignment of n attribute sets $(z_1, z_2, \ldots, z_n)$, where $z_i \subseteq \mathcal{A}$. We let z_i be a set so that a query can have multiple intents. In our current work, we focus on intent identification within a category and assume a query to category mapping is available; a fairly standard assumption in vertical search engines [3].

Constructing Intent Labels from Click Logs: Manual intent labelling of queries is a laborious task requiring significant domain expertise. However, for queries that occur sufficiently often in the click logs, matches between the query tokens and the attribute-values of the clicked products provide a natural means of obtaining the attribute labels. Following [19], for a particular query we find matches between its tokens and the tokens of the attribute-values of every product that is clicked for this query. We then aggregate these matches across attributes to construct intent labels for every token in the query. This process is applied to queries that occur at least 500 times in a month with a click-through rate of at least 40%. Using such frequent queries with high click-through rates lets us construct reliable and fairly noise-free attribute labels for them. Applying this process to tail queries will result in fairly noisy attribute labels [11,14,17]. The labelled data set thus constructed is denoted by $\mathcal{D}^L$ and is referred to as the *click-log labelled data* in this paper. The average number of such labelled queries $\mathcal{D}^L$ for the Jewellery, Furniture, and Home Furnishing categories is ≈5k while the average number of unique queries $\mathcal{D}$ that occur at least 10 times a month in these categories is ≈50k. The labelled queries are much fewer than the unique queries which shows the limitations of constructing intent labels from click logs.

3 Empirical Data Analysis

Query intent understanding on a large scale product catalog presents unique challenges and we discuss two distinct characteristics here. The fraction of unique queries with a particular attribute pattern in the click-log labelled data has a long-tailed distribution as shown in Fig. 2a. Two example attribute patterns for the query 'silver oxidised earring' are 'color, model name, store name' and 'plating, model name, store name' as shown in Fig. 1. From Fig. 2a, it is noteworthy that the most frequent attribute pattern represents on average only 5% of the unique queries in the three categories. This makes supervised learning difficult since most attribute patterns have very few example queries. Moreover, this analysis is for the relatively frequent queries $\mathcal{D}^L$ and we expect this distribution to have an even longer tail for tail queries.



(a) The proportion of unique queries in 100 most frequent attribute patterns in the labelled data.

(b) Each point in the heat map is the normalized overlap between the vocabularies of a pair of attributes. Each graph visualises a random set of 30 attribute pairs.

Fig. 2. Empirical data analysis

The average number of attributes $\mathcal{A}$ for the three categories is ≈130 while the average size of the vocabulary $\mathcal{W}$ is ≈20k. Many pairs of attributes have a significant degree of overlap between their vocabularies. We illustrate this in Fig. 2b where the non-zero entries in the heat map indicate an overlap between the vocabularies of a particular pair of attributes. For example, the attributes 'plating' and 'base material' in the Jewellery category have an overlap of ≈30% in their vocabularies. This overlap indicates the possibility of multiple attributes being the correct labels for a token in a query and thus the query having multiple correct intents. We use this characteristic to develop a regularisation technique that improves our model's ability to capture multiple intents in queries.

4 The Latent Variable Generative Model

Tail queries have very few clicks and thus the click log mining technique of Sect. 2 can not be used to derive labels for them. Generative models are naturally suited to an unsupervised setting where labels are absent. The authors of [5] propose a simple generative process for queries which generates query tokens independently by first sampling an attribute and then sampling a token from that attribute's vocabulary. However, modelling dependence is important since the attribute label for a token depends on the other tokens in a query. For example, consider the queries 'cotton sofa cushion' and 'cotton bed sheets double bed'. The correct label for 'cotton' in the first query is 'filling material' while in the second query is 'fabric'. This highlights the need for a richer generative model that captures token interactions and attribute co-occurrences in a query.

We propose a latent variable generative model for queries where the observed variables are ordered pairs of tokens and the latent variables are the corresponding ordered pairs of attribute labels. The generative process is defined over all ordered pairs of tokens in a query and not just the adjacent ones. For example, there are 3 ordered pairs of tokens in the query 'silver oxidised earring': ('silver', 'oxidised'), ('oxidised', 'earring'), and ('silver', 'earring').

Let $c_{\mathbf{x}}$ be the set of all ordered pairs of tokens in a query $\mathbf{x}$. We define ψ as a $|\mathcal{A}| \times |\mathcal{A}|$ matrix of parameters specifying the attribute co-occurrence probabilities, i.e., $\sum_a \psi_{a,a'} = 1$ for each a'. We similarly define ϕ as a $|\mathcal{W}| \times |\mathcal{A}|$ matrix of parameters specifying the probability of generating a token from an attribute, i.e., $\sum_w \phi_{w,a} = 1$ for each a. We assume that the ith ordered token pair $x_i = (x_{i1}, x_{i2})$ is generated from a corresponding ordered attribute pair $z_i = (z_{i1}, z_{i2})$ as follows: Sample an attribute z_{i1} uniformly at random and then sample the attribute $z_{i2} \sim \text{Mult}(\psi_{\cdot, z_{i1}})$ conditioned on z_{i1}. The token pair x_i is then generated by sampling $x_{i1} \sim \text{Mult}(\phi_{\cdot, z_{i1}})$ and $x_{i2} \sim \text{Mult}(\phi_{\cdot, z_{i2}})$. The joint probability of x_i and z_i is thus given by

$$p(x_i, z_i) = p(x_{i1}|z_{i1})\, p(x_{i2}|z_{i2})\, p(z_{i2}|z_{i1})\, p(z_{i1}) = \phi_{x_{i1},z_{i1}}\, \phi_{x_{i2},z_{i2}}\, \psi_{z_{i2},z_{i1}}\, \tfrac{1}{|\mathcal{A}|}.$$

Therefore, our model represents queries as a set of all ordered pairs of its tokens and we assume all pairs to be independent to get $p(\mathbf{x}) \approx p(c_{\mathbf{x}}) = \prod_i \sum_{z_i} p(x_i, z_i)$. This assumption is critical for computational tractability while still capturing the interactions between the tokens as well as the co-occurrences between the attributes. The observed log-likelihood which we optimize using the standard Expectation Maximization (EM) algorithm is

$$l_o(q, \phi, \psi) = \sum_i \left[\mathbb{E}_{q_i}\left[\log\left(\frac{p(x_i, z_i)}{q_i(z_i)} \right) \right] + \text{KL}(q_i \,\|\, p(z_i|x_i)) \right]. \tag{1}$$

Via a standard derivation, the E-Step update for the token pair x_i is given by

$$q_i((z_{i1} = a, z_{i2} = a')) = \frac{\phi_{x_{i1},a}\phi_{x_{i2},a'}\psi_{a,a'}}{\sum_{(a,a')} \phi_{x_{i1},a}\phi_{x_{i2},a'}\psi_{a,a'}}, \tag{2}$$

where $q_i((z_{i1} = a, z_{i2} = a'))$ is the posterior probability of the attribute pair (z_{i1}, z_{i2}) being (a, a') given the token pair (x_{i1}, x_{i2}). Via a standard derivation, the M-Step updates for the parameters ϕ and ψ are given by

$$\phi^{(o)}_{w,a} = \frac{\sum_i \Big[\mathbb{1}[x_{i1}{=}w]\, q_i((a,\cdot)) + \mathbb{1}[x_{i2}{=}w]\, q_i((\cdot,a))\Big]}{\sum_i \Big[q_i((a,\cdot)) + q_i((\cdot,a))\Big]}, \quad \psi^{(o)}_{a,a'} = \frac{\sum_i q_i((a,a'))}{\sum_i q_i((\cdot,a'))}, \quad (3)$$

where $q_i((\cdot, a)) = \sum_{a'} q_i((z_{i1} = a', z_{i2} = a))$ and $q_i((a, \cdot))$ is defined similarly.

Since our model is defined over pairs of tokens, computing the attribute assignments for each token in a query during posterior inference requires an approximation. We follow [18] and approximate the posterior distribution of the attribute assignments by decomposing it over pairs of tokens as follows

$$p(\mathbf{z}|\mathbf{x}) \approx p((z_1, z_2)|(x_1, x_2)) \prod_{i=3}^{n-1} p((z_{i-1}, z_i)|(x_{i-1}, x_i)).$$

We compute multiple attribute assignments at each position in the query using a standard forward-backward algorithm to obtain multiple intents per token.

5 Regularisation for Learning Multiple Intents

Queries with multiple intents have multiple attribute labels for one or more of their tokens, for example, the token ‘silver’ in the query ‘silver oxidised earring’ shown in Table 1. As illustrated in Fig. 2b, certain attributes have a significant overlap between their vocabularies. We use this observation to define a similarity measure between attributes using background estimates of the generative model’s parameters. We then use this similarity measure to devise a data dependent regularisation technique that distributes the generative model’s posterior across attributes with significantly overlapping vocabularies which improves its ability to detect multiple intents.

5.1 Background Parameter Estimates

We use the product catalog and the click-log labelled data to derive background estimates for the generative model’s parameters. To derive the estimates for ϕ, we first iterate over all products in a category and construct the set $\{(a, v, \kappa_{v,a})\}$, where a is an attribute, v is an attribute value and $\kappa_{v,a}$ is the number of products with v as the attribute-value for the attribute a. We then define the estimate

$$\widetilde{\phi}_{w,a} = \frac{C(w,a) + \varphi_a}{\sum_w C(w,a) + \varphi_a |\mathcal{W}|},$$

where $C(w,a) = \frac{C^U(w,a) + C^L(w,a)}{2}$, $C^U(w,a) = \sum_v \frac{\mathbb{1}_{[w \in v]} \log \kappa_{v,a}}{|v|}$, $C^L(w,a)$ is the number of times the token w is labelled with the attribute a in the click-log labelled data set $\mathcal{D}^L$, and $\varphi_a = \frac{\varphi}{\max_w C(w,a)}$ is a smoothing factor with $\varphi > 0$ being a hyper-parameter.

To derive the estimates for ψ, we first iterate over all products in a category and construct the set $\{(a, a', \kappa_{a,a'})\}$, where a and a' are attributes and $\kappa_{a,a'}$ is the number of products having both attributes a and a'. We then define the estimate

$$\widetilde{\psi}_{a,a'} = \frac{C(a,a') + \omega_{a'}}{\sum_a C(a,a') + \omega_{a'}|\mathcal{A}|},$$

where $C(a,a') = \frac{C^U(a,a')+C^L(a,a')}{2}$, $C^U(a,a') = \log \kappa_{a,a'}$, $C^L(a,a')$ is the number of times the attribute pair (a,a') co-occur in the click-log labelled data set $\mathcal{D}^L$, and $\omega_{a'} = \frac{\omega}{\max_a C(a,a')}$ is a smoothing factor with $\omega > 0$ being a hyper-parameter.

5.2 Attribute Similarity Regularisation

The probability of the model generating the token w from the attribute a is given by the model parameter $\phi_{w,a}$ and its background estimate is $\widetilde{\phi}_{w,a}$. Thus, if $\widetilde{\phi}_{w,a} \approx \widetilde{\phi}_{w,b}$ and both background estimates are high, then the model should pick both attributes a and b as relevant labels for the token w. Analogously, if two attribute pairs (a,a') and (b,b') have similar high background estimated probabilities of generating the token pair (w,w'), then the model should pick both attribute pairs (a,a') and (b,b') as relevant labels for the token pair (w,w'). We quantify this notion by defining

$$g_{(w,w')}((a,a'),(b,b')) = (\widetilde{\phi}_{w,a}\widetilde{\phi}_{w,b})^2(\widetilde{\phi}_{w',a'}\widetilde{\phi}_{w',b'})^2.$$

Note that g is high when $\widetilde{\phi}_{w,a} \approx \widetilde{\phi}_{w,b}$, $\widetilde{\phi}_{w',a'} \approx \widetilde{\phi}_{w',b'}$ and the individual $\widetilde{\phi}$'s are high. We use this notion of attribute similarity to define a regularisation term that distributes the generative model's posterior across attribute pairs with similar vocabularies. Let $g_{(w,w')}$ denote a square positive matrix of size $|\mathcal{A}|^2 \times |\mathcal{A}|^2$ over the attribute pairs. Alternating normalization of the rows and columns (the Sinkhorn-Knopp algorithm [23]) of $g_{(w,w')}$ will generate a doubly stochastic matrix $\bar{g}_{(w,w')}$ that we will use instead of $g_{(w,w')}$ as the measure of similarity. For a token pair x, the regularisation term penalizes large differences in the posterior probabilities $p((a,a')|x)$ and $p((b,b')|x)$ if $\bar{g}_x((a,a'),(b,b'))$ is high and is given in the following regularised log-likelihood

$$l_o(q,\phi,\psi) - \alpha \sum_i \left[\frac{1}{2}\sum_{z,\bar{z}} \bar{g}_{x_i}(z,\bar{z})\big(p(z|x_i) - p(\bar{z}|x_i)\big)^2\right], \tag{4}$$

where $x_i = (x_{i1}, x_{i2})$ is the ith token pair, z and $\bar{z}$ are attribute pairs, and $\alpha \in (0,1)$ is a hyper-parameter. Unfortunately, maximizing the above regularised log-likelihood becomes intractable due to a coupling of the model parameters in the M-step optimization. So we establish the following upper bound on the regularisation term in (4) that gives us a lower bound on the regularised log-likelihood that is tractable to maximize.

Theorem 1. *Let* $\mathbf{z}$ *and* $\mathbf{x}$ *be discrete random variables and* $\bar{g}_{\mathbf{x}}$ *be a* $|\mathbf{z}| \times |\mathbf{z}|$ *doubly stochastic matrix. Then, for any distribution* $q_{\mathbf{x}}$*, we have*

$$\tfrac{1}{2}\sum_{z,\bar{z}} \bar{g}_{\mathbf{x}}(z,\bar{z})\big(p(z|\mathbf{x}) - p(\bar{z}|\mathbf{x})\big)^2 \leq \tfrac{1}{2}\sum_{z,\bar{z}} \bar{g}_{\mathbf{x}}(z,\bar{z})\big(q_{\mathbf{x}}(z) - q_{\mathbf{x}}(\bar{z})\big)^2 + \min\Big[1, 5\sqrt{2\mathrm{KL}(q_{\mathbf{x}} \,\|\, p(\mathbf{z}|\mathbf{x}))}\Big]$$

Applying this bound on the posterior distribution $p(z_i|x_i)$ *and the approximate posterior distribution* q_i *gives the following lower bound on* (4)

$$\mathbb{E}_{q_i}\left[\log\left(\frac{p(x_i, z_i)}{q_i(z_i)}\right)\right] - \alpha\left[\frac{1}{2}\sum_{z,\bar{z}} \bar{g}_{x_i}(z,\bar{z})\big(q_i(z) - q_i(\bar{z})\big)^2\right] - \frac{5\alpha}{4}. \tag{5}$$

Proof. See Online Supplementary Material.

Thus, the regularised E-step optimization is

$$\max_{q_i} \mathbb{E}_{q_i}\left[\log\left(\frac{p(x_i, z_i)}{q_i(z_i)}\right)\right] - \alpha\left[\frac{1}{2}\sum_{z,\bar{z}} \bar{g}_{x_i}(z,\bar{z})\big(q_i(z) - q_i(\bar{z})\big)^2\right], \tag{6}$$

subject to $\sum_{z_i} q_i(z_i) = 1$, where we have dropped the constant term involving α. The optimization in (6) can be done via projected gradient descent [21]. In our experiments, we observed that 3 to 4 iterations were usually sufficient for convergence and that our method results in the posterior distribution being distributed over similar attribute pairs instead of being concentrated on one of them. The M-step updates for this model are exactly the same as in (3).

6 Experiments and Analysis

We evaluated our proposed model against several strong baseline models on data sets proprietary to Flipkart. To the best of our knowledge, there are no publicly available data sets for evaluating query intent algorithms for e-commerce search or similar domains and all previous related work [11,16,18,19,24] has been evaluated on such proprietary data sets. We selected the Jewellery, Home Furnishing, and Furniture categories for experimental evaluation. These categories at Flipkart have a high business value in spite of low query volume and thus very sparse click data leading to more tail queries as compared to more popular categories like Electronics or Lifestyle. The click-log labelled data set $\mathcal{D}^L$ and the unlabelled data set $\mathcal{D}$ used to train all models were obtained from one month of query logs. We restricted $\mathcal{D}$ to queries with at least 10 occurrences over that month to filter out queries with misspellings.

6.1 Baseline Models

There is little prior work on understanding the intent of e-commerce search queries, especially in our setting where we have access to labelled as well as unlabelled query logs in addition to data from the product catalog. Prior work on intent understanding can be broadly classified into supervised and unsupervised methods. The unsupervised baseline model we compare against is UMM [5] described in Sect. 4. The supervised baseline models we compare against are Multinomial Logistic Regression (LR), the Linear Chain CRF from the query intent understanding work in [11,19], and the Bi-LSTM-CRF from [10]. The recent work [26] on understanding intent in Google shopping queries is not applicable in our setting since it focuses on a different problem of understanding overall query intent and not token level attribute labelling as ours. These supervised baseline models were trained on the click-log labelled data set $\mathcal{D}^L$ with elastic-net regularisation whose hyper-parameters were selected by 3-fold cross-validation with F_1 score as the performance metric.

Multinomial LR and Linear Chain CRF: Each training instance consisted of a query token x_i at position i and its attribute label z_i taken from the queries in $\mathcal{D}^L$. We extended the features from [19] by defining additional catalog features in terms of matches between the query tokens and the catalog attributes and additional syntactic features in terms of the surface form of the tokens. The catalog features were unigram and bigram TF-IDF matches with the vocabulary of each attribute in a category. The syntactic features were whether a unigram is a stopword, is a short word with less than 4 characters, or is alphanumeric.

Bi-LSTM-CRF: We implemented two variants of the Bi-LSTM-CRF from [10]. The first, Bi-LSTM-1, used 100-dimensional word embeddings trained on the product descriptions from the catalog (using fastText [13]) as its features. The second, Bi-LSTM-2, additionally used the catalog features described above. It is important to note that we have a much stronger set of features compared to the standard implementations of a Bi-LSTM-CRF since we incorporate where a unigram or a bigram matches in the attribute space.

We evaluated all baseline models against the all pairs mixture model (PMM) described in Sect. 4 and the all pairs mixture model with attribute similarity regularisation (RIM) described in Sect. 5.2.

6.2 Evaluation of Intent Labellings

A team of search quality experts at Flipkart labelled a random sample of tail queries from the query logs using their domain expertise. We randomly selected 900 queries with multiple intents (300 queries per category) from this labelled set on which to evaluate all models and refer to it as the *golden set*. We further created 5 randomized 80/20 splits of the golden set to get multiple test and validation sets. We computed marginal distributions at each token position in a query for all models and considered only those labellings that were above a threshold tuned on a validation set. We chose F_1 score as the performance metric and since we are interested in queries with multiple intents, we follow [8] and get the overall F_1 score per query by micro-averaging the F_1 score per query token. We used the same validation and test sets for all models in each run.

The performance of all models on the test sets is summarized in Fig. 3 and Table 2. RIM outperforms PMM as well as all baseline models with an average improvement of 12.5% in F_1 score over UMM, the best performing baseline model. RIM achieves an average improvement of 13.4%, 15.2%, and 8.6% in F_1 score over UMM for the Furniture, Home Furnishing, and Jewellery categories. Moreover, RIM and PMM together outperform all baseline models which demonstrates the effectiveness of modelling pairwise dependencies between the query tokens. All the supervised baseline models including Bi-LSTM-CRF, a

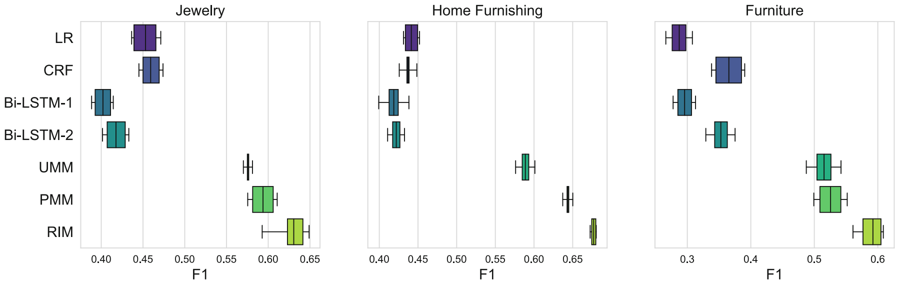


Fig. 3. Box plots of F_1 scores on the held-out test splits of the golden set for all models.

state-of-the-art model for slot-tagging problems, perform much worse than the unsupervised baseline model UMM due to a lack of sufficient labelled data. RIM's performance improvements over PMM demonstrate the effectiveness of our data dependent attribute similarity regularisation for queries with multiple intents. An example of this is illustrated in Table 3 where RIM's posterior is distributed over the correct attribute labels whereas that of PMM is distributed over the correct and incorrect attribute labels.

Table 2. Average F_1 scores on the held-out test splits of the golden set for all models. The results for RIM are statistically significant against all baselines with p-value < 0.01.

	LR	CRF [11,19]	Bi-LSTM-1 [10]	Bi-LSTM-2 [10]	UMM [5]	PMM	RIM
Jewellery	0.45	0.46	0.40	0.42	0.58	0.59	**0.63**
Home Furnishing	0.44	0.44	0.42	0.42	0.59	0.64	**0.68**
Furniture	0.29	0.36	0.29	0.35	0.52	0.53	**0.59**
Average	0.39	0.42	0.37	0.39	0.56	0.59	**0.63**

Table 3. The marginal posterior distributions for the token 'silver' in the query 'silver oxidised earring' returned by PMM and RIM. Here, $\delta < 10^{-4}$ and the correct attribute labels are color, plating, and base material.

	color	plating	base material	store	model	ideal for	body material
PMM	0.381	0.148	0.121	0.143	0.059	0.033	0.015
RIM	0.693	0.135	0.081	δ	δ	δ	δ

6.3 Performance in an Online A/B Experiment

The intent inferred for a search query plays a major role in determining and retrieving the most relevant products for that query at Flipkart as is standard in e-commerce search [15]. Thus, the quality of the inferred intent very strongly influences a user's propensity to click and add-to-cart the products retrieved for a search query. Hence, we measure the click-through rate (CTR) and the add-to-cart ratio as the relevant metrics in the online A/B experiment. The add-to-cart ratio (i.e., search conversion) is defined as the fraction of searches leading to a product being added to the shopping cart. We deployed RIM and UMM in the production search system at Flipkart and compared the performance of the models against each other in a standard A/B experiment configuration where we treated UMM as the control condition. More than 10 million users visit Flipkart daily and we randomly assigned 15% of the users to each condition and conducted the test over 10 days. Since the models were trained for the Jewellery, Home Furnishing, and Furniture categories, only those queries belonging to these categories were considered for comparison. The query to category mapping was obtained by a separate production system at Flipkart. We would have ideally liked to restrict the experiment to tail queries with multiple intents only in order to better demonstrate the capabilities of RIM. However, in practice it is difficult to determine on the fly if a query has multiple intents. Thus, we conducted the experiment on all tail queries. The query volume affected by the experiment was ≈75k tail queries (with ≈ 36k unique queries). The results of this online A/B experiment are summarized in Table 4.

Table 4. Results of the online A/B experiment comparing RIM against the best baseline model UMM. Statistical significance with p-value < 0.01 is denoted by $*$ and that with p-value < 0.001 is denoted by $\wedge$.

	Tail CTR (%)	Tail Add-to-Cart (%)
Jewellery	$+2.78^*$	$+10.22^*$
Home Furnishing	$+2.13^\wedge$	$+13.46^\wedge$
Furniture	$+4.19^\wedge$	$+22.67^\wedge$
Average	+3.03	+15.45

RIM significantly improves both the CTR and the add-to-cart ratio for tail queries across all categories. The average improvement in CTR is 3.03% while that in add-to-cart ratio is 15.45%. The results for all categories were statistically significant as measured by a paired sample t-test with p-value < 0.01. The much larger improvement in add-to-cart ratio as compared to CTR is noteworthy. On further analysis, we found that most tail queries express a very specific product need and when the search system is able to infer the correct query intent and retrieve the relevant products, the customers are satisfied, as indicated by add-to-cart, with fewer clicks. We are thus able to demonstrate the effectiveness of the proposed model in a large scale real world setting. We finally illustrate the retrieval quality with intents inferred by RIM compared to the existing production system for two queries in Figs. 4 and 5. Both queries are tail queries drawn from the online A/B experiment and RIM correctly identifies their intents.

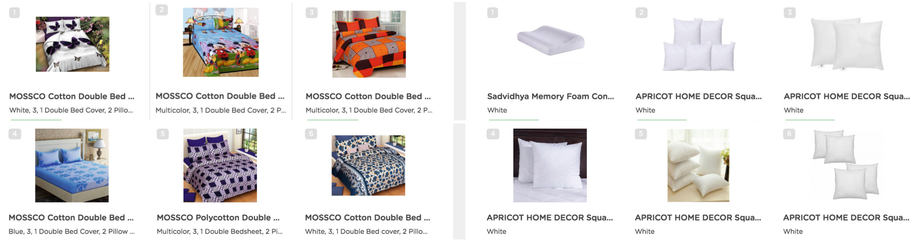

1
MOSSCO Cotton Double Bed ...
White, 3, 1 Double Bed Cover, 2 Pillo...
2
MOSSCO Cotton Double Bed ...
Multicolor, 3, 1 Double Bed Cover, 2 P...
3
MOSSCO Cotton Double Bed ...
Multicolor, 3, 1 Double Bed Cover, 2 P...
4
MOSSCO Cotton Double Bed ...
Blue, 3, 1 Double Bed Cover, 2 Pillow ...
5
MOSSCO Polycotton Double ...
Multicolor, 3, 1 Double Bedsheet, 2 Pi...
6
MOSSCO Cotton Double Bed ...
White, 3, 1 Double Bed cover, 2 Pillo...
1
Sadvidhya Memory Foam Con...
White
2
APRICOT HOME DECOR Squa...
White
3
APRICOT HOME DECOR Squa...
White
4
APRICOT HOME DECOR Squa...
White
5
APRICOT HOME DECOR Squa...
White
6
APRICOT HOME DECOR Squa...
White

Fig. 4. Top retrieved products for the query 'small pillow cover pack' by the existing production system (left) and with intents inferred by RIM (right). The production system retrieves irrelevant bed sheets. RIM correctly identifies 'pillow cover' as 'store/model' and 'small' as 'size/shape'.

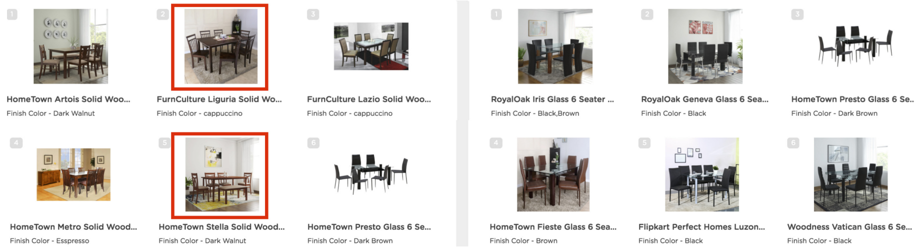

1
HomeTown Artois Solid Woo...
Finish Color - Dark Walnut
2
FurnCulture Liguria Solid Wo...
Finish Color - cappuccino
3
FurnCulture Lazio Solid Woo...
Finish Color - cappuccino
4
HomeTown Metro Solid Wood...
Finish Color - Espresso
5
HomeTown Stella Solid Wood...
Finish Color - Dark Walnut
6
HomeTown Presto Glass 6 Se...
Finish Color - Dark Brown
1
RoyalOak Iris Glass 6 Seater ...
Finish Color - Black,Brown
2
RoyalOak Geneva Glass 6 Sea...
Finish Color - Black
3
HomeTown Presto Glass 6 Se..
Finish Color - Dark Brown
4
HomeTown Fieste Glass 6 Sea...
Finish Color - Brown
5
Flipkart Perfect Homes Luzon...
Finish Color - Black
6
Woodness Vatican Glass 6 Se...
Finish Color - Black

Fig. 5. Top retrieved products for the query 'glass top wooden dining table 6 seater' by the existing production system (left) and with intents inferred by RIM (right). The products highlighted in red are wooden top tables and thus irrelevant for the query. RIM correctly identifies 'glass' as 'top material'. (Color figure online)

7 Related Work

The existing work on query understanding has mainly focused on learning query intent in a supervised manner by using click-through data [6,9,12,22] and this restricts their generalization to combinations of frequent attribute patterns only. However, tail queries exhibit tail attribute patterns and this is the focus of our current work. The existing methods for understanding intent of tail queries can be broadly divided into two major types: (a) Those that identify a mapping between tail queries and similar frequent queries [11,24], and (b) Those that learn query intent from partially labelled queries [16,17,19]. Fusing the results of a tail query with those of a similar frequent query as a way of improving retrieval metrics is suggested in [11]. However, the underlying assumption that a tail query is a frequent query that is expressed differently does not hold in our case. Transferring the intent of frequent queries to tail queries using an external knowledge base is studied in [24]. However, building domain specific knowledge bases is difficult. Learning query intent from partially labelled queries along with side-supervision in the form of derived attributes for some query tokens is studied in [19]. However, it is difficult to obtain partial labellings for tail queries because most tokens in tail queries will be marked as 'unknown' due to the sparsity of the click-through data as observed in [17]. A hidden-unit linear-chain CRF that allows for non-linearities is introduced in [16]. However, its formulation too requires partially labelled queries. The availability of derived labelled data by performing rule-based labelling of unlabelled sequences is assumed in [4]. However, the rule-based labelling is domain specific and is difficult to extend. The CRF auto-encoder [1] and its application to tasks like POS tagging [20] is promising especially since it does not require labelled data. However, the CRF auto-encoder has difficulty scaling to the label space for query intent understanding that is much larger than that for POS tagging. The most recent work on understanding intent of e-commerce search queries is described in [26] for Google shopping. However, it is not applicable in our setting since it focuses on a different problem of understanding overall query intent and not token level attribute labelling.

8 Conclusion and Future Work

In this paper, we investigated the problem of discovering multiple intents in tail queries for e-commerce search. We introduced a latent variable generative model for queries to overcome the lack of sufficient labelled data. To improve this model's ability to identify multiple intents, we then introduced a novel data dependent regularisation technique derived from empirical evidence of overlap in attribute vocabularies. We finally demonstrated the superior performance of our regularised intent model against several strong baseline models on an editorially labelled data set as well as in a large scale online A/B experiment at Flipkart, a major Indian e-commerce company. In the future, we plan to investigate deep generative intent models and knowledge graph representation of the product catalog to further improve intent understanding.

References

1. Ammar, W., Dyer, C., Smith, N.A.: Conditional random field autoencoders for unsupervised structured prediction. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 2, pp. 3311–3319. MIT Press, Cambridge (2014)
2. Anderson, C.: The Long Tail: Why the Future of Business is Selling Less of More. Hyperion (2006)
3. Arguello, J., Diaz, F., Callan, J., Crespo, J.F.: Sources of evidence for vertical selection. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 2009, pp. 315–322. ACM, New York (2009). https://doi.org/10.1145/1571941.1571997
4. Dredze, M., Talukdar, P.P., Crammer, K.: Sequence learning from data with multiple labels. In: Learning from Multi-Label Data at ECML PKDD, 2009, vol. 39 (2009)
5. Duan, H., Zhai, C., Cheng, J., Gattani, A.: Supporting keyword search in product database: a probabilistic approach. In: Proceedings of the VLDB Endowment, vol. 6, no. 14, pp. 1786–1797 (2013)

6. Gao, J., He, X., Nie, J.Y.: Clickthrough-based translation models for web search: from word models to phrase models. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010, pp. 1139–1148. ACM, New York (2010)
7. Goel, S., Broder, A., Gabrilovich, E., Pang, B.: Anatomy of the long tail: ordinary people with extraordinary tastes. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM 2010, pp. 201–210. ACM, New York (2010)
8. Han, J., Fan, J., Zhou, L.: Crowdsourcing-assisted query structure interpretation. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI 2013, pp. 2092–2098. AAAI Press (2013)
9. Hashemi, H.B., Asiaee, A., Kraft, R.: Query intent detection using convolutional neural networks. In: International Conference on Web Search and Data Mining, Workshop on Query Understanding (2016)
10. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991 (2015). http://arxiv.org/abs/1508.01991
11. Huo, S., Zhang, M., Liu, Y., Ma, S.: Improving tail query performance by fusion model. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. CIKM 2014, pp. 559–568. ACM, New York (2014)
12. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 133–142. ACM, New York (2002)
13. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427–431. Association for Computational Linguistics (2017)
14. Kang, C., Lin, X., Wang, X., Chang, Y., Tseng, B.L.: Modeling perceived relevance for tail queries without click-through data. CoRR abs/1110.1112 (2011)
15. Karmaker Santu, S.K., Sondhi, P., Zhai, C.: On application of learning to rank for e-commerce search. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017, pp. 475–484. ACM, New York (2017)
16. Kim, Y.B., Jeong, M., Stratos, K., Sarikaya, R.: Weakly supervised slot tagging with partially labeled sequences from web search click logs. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 84–92. Association for Computational Linguistics (2015)
17. Kiseleva, J., Agichtein, E., Billsus, D.: Mining query structure from click data: A case study of product queries. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 2217–2220. ACM, New York (2011). https://doi.org/10.1145/2063576.2063930
18. Konishi, T., Ohwa, T., Fujita, S., Ikeda, K., Hayashi, K.: Extracting search query patterns via the pairwise coupled topic model. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM 2016, pp. 655–664. ACM, New York (2016)
19. Li, X., Wang, Y.Y., Acero, A.: Extracting structured information from user queries with semi-supervised conditional random fields. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, pp. 572–579. ACM, New York (2009)

20. Lin, C.C., Ammar, W., Dyer, C., Levin, L.: Unsupervised POS induction with word embeddings. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1311–1316. Association for Computational Linguistics (2015)
21. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. The MIT Press, Cambridge (2012)
22. Radlinski, F., Szummer, M., Craswell, N.: Inferring query intent from reformulations and clicks. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 1171–1172. ACM, New York (2010). https://doi.org/10.1145/1772690.1772859
23. Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. Ann. Math. Stat. **35**(2), 876–879 (1964)
24. Song, Y., Wang, H., Chen, W., Wang, S.: Transfer understanding from head queries to tail queries. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, pp. 1299–1308. ACM, New York (2014)
25. Szpektor, I., Gionis, A., Maarek, Y.: Improving recommendation for long-tail queries via templates. In: Proceedings of the 20th International Conference on World Wide Web. WWW 2011, pp. 47–56. ACM, New York (2011). https://doi.org/10.1145/1963405.1963416
26. Wu, C.Y., Ahmed, A., Kumar, G.R., Datta, R.: Predicting latent structured intents from shopping queries. In: Proceedings of the 26th International Conference on World Wide Web. WWW 2017, pp. 1133–1141, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017)