Subjective Search Intent Predictions using Customer Reviews

Adrian Boteanu, Emily Dutile, Adam Kiezun, Shay Artzi
Amazon.com
[boteanua,emilyj,akkiezun,artzi]@amazon.com

ABSTRACT

Query intent prediction is a component of information retrieval which improves result relevance through an understanding of latent user intents in addition to explicit query keywords. We target context-of-use intents, such as the activity for which a product is used and the target audience for a product, which are subjective and not usually indexed as product attributes in the catalog. We describe a method to predict latent query intents: we extract intents from product reviews on amazon.com and, using behavioral purchase signals that associate queries with the reviewed products, train query classifiers that label queries with the intents extracted from reviews. For example, we predict the activity "running" for the query "adidas mens pants." We show that our method can predict latent intents not indexed directly in the product catalog.

CCS CONCEPTS

• **Information systems → Query intent**; *Thesauri*.

KEYWORDS

query intent, weak supervision, information retrieval, product reviews

ACM Reference Format:

Adrian Boteanu, Emily Dutile, Adam Kiezun, Shay Artzi. 2020. Subjective Search Intent Predictions using Customer Reviews. In *2020 Conference on Human Information Interaction and Retrieval (CHIIR'20), March 14–18, 2020, Vancouver, BC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Lexical information retrieval systems attempt to identify relevant documents by comparing keywords specified in the query with keywords indexed for the documents [7]. This restricts the set of possible results to products for which a subset of the query keywords are indexed. In contrast, semantic search presents products that abide to the meaning expressed in the query, not only the exact keywords [10, 19]. Understanding query intents is a component of semantic search, which expands the search query not only to semantically-related terms and concepts, but also to implied concepts. For example, customers may search for "waterproof shoes," but could in fact be interested in rugged shoes for summer hiking, or running shoes for cold weather training, or office shoes that are water resistant.

Shopping intents that describe context-of-use are often contextual and subjective, for example, one could go hiking while wearing a variety of footwear depending on personal preference and season. This is in contrast with the traditional view of a product catalog, which is to describe intrinsic properties of the product such as weight or size. Thus, we leverage customer reviews for references of such product properties. In this paper, we target on two classes of context-of-use concepts: Activity, such as "cooking" or "running," and Audience, such as "parent" or "daughter." The main hypothesis of this work is that online shoppers express their context-of-use after making a purchase by writing reviews, while search queries lack this information explicitly. Thus, we can anticipate those latent intents at query time by learning from the reviews written after purchases from previous customers.

We target intents that are present in search queries but are not explicitly modeled as product attributes. We define context-of-use concepts as entities in a knowledge graph; for this work, we use an entity's type and aliases from the graph, which represent a set of shallow taxonomies of trees of type and instances. For example, "teenagers" is a sub-type of Audience and have the alias "teens." These taxonomies were designed by ontologists to model entities present in search traffic. Entities have unique identifiers, but in this work refer to their primary name for the sake of readability.

In regard to predicting latent intents for queries, we evaluate the following hypotheses:

(1) We can predict latent intents in queries by using entity extraction on customer review texts from reviews written for products associated with the query; we evaluate this hypothesis in Section 4.2;
(2) We can leverage weaker behavioral signals, expanding the applicability of our method to less frequent queries and less purchased products; we evaluate this hypothesis in Section 4.3;

We leverage an Amazon-internal dataset, referred here as B, which contains affinity scores between query keywords and product IDs (ASIN. Amazon Standard Identification Number) using previous customer behavior in search, such as clicks, adds to cart, and purchases. The resulting scores are integers between 1 and 15, the higher the score the stronger the affinity between a query and a product.

Our method consists of the following steps:

(1) Extract entities from customer reviews and label products with entities identified in their corresponding reviews (Section 3.1). For example, we match the entity "baking," which is a type of Activity, in the phrase "I like baking on Sundays," which is part of a customer review for ASIN B073P4RPFP;

CHIIR'20, March 14–18, 2020, Vancouver, BC, Canada

ACM ISBN 978-1-4503-6892-6/20/03…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

(2) Join labeled products with search keywords using behavioral data. For example, we label ASIN X with "baking," and the query "muffin tray" is associated with ASIN B073P4RPFP with a score of 12 out of 15;
(3) Train neural network models that learn to predict entities from the query keywords. For example, one of the training inputs is the query "muffin tray," the score 12, and the label "baking."(Section 3.2).

We build classifiers for all query volume, without restricting to a particular index or product category. This is a more challenging task than building category-specific classifiers, but it will lead to experiences in which we can leverage context-of-use intents to present products from multiple categories.

2 RELATED WORK

This paper bridges work from query intent prediction and entity extraction from text.

Intent for shopping search queries has been modeled using features such as query reformulation and click-through data [16], transaction logs [14], search sessions [5], Wikipedia data [13], and query vector representations [12].

Previous work trained query labeling models using catalog data and customer action in response to search queries; in their work, the model learns to label queries with structured attributes of products in the catalog, for example the screen size value for a TV [18]. In this work, we generalize this approach to include entity extraction from unstructured text. This introduces greater uncertainty in the learning process, not only because entity extraction from text is prone to error compared to using structured catalog data, but also because the entities we target are subjective. For example, the screen size of a given TV model is an intrinsic and objective property of that product, while our method targets context-of-use properties expressed in customer opinions. We include weaker associations between queries and products to expand the training set to less frequent queries or for products that result in fewer purchases.

Entity extraction is the task of identifying entities in unstructured text [2–4, 15]. This is related but different from keyword and keyphrase extraction methods [17], in that entities have unique identifiers and may be mentioned in text by multiple aliases. Named entity extraction focuses on identifying people, places, and other distinct concepts [8, 9]. We focus on common but abstract concepts, for example the concept of "hiking" is not unique in the world in the same way the concept of "Planet Earth" is.

3 METHODS

We first describe our method of constructing the training datasets used throughout this work. We then describe the architecture of our models.

3.1 ASIN Entity Labeling from Reviews

We target two types of entities, Activity and Audience. Examples of activities include "biking," with aliases "bicycling" and "cycling," and "child care," with aliases parenting, childcare, child minding, daycare, and preschool. Examples of Audience include "grandfather," with aliases granddad, grandpa, grandpa, grand-father, gramps, grampa, grandsire, and "teacher," with aliases schoolmaster, schoolmistress, educator, professor, school teacher, and "mother, " with aliases mom, mum, mommy.

In total we use 112 Activity intents and 61 Audience intents. We leveraged Wikidata[1] to enrich the entity set with aliases. We used the Wikimedia API to search for entity aliases, selected the top results for page links, and manually selected the results. While this step could be automated with entity linking [11], it is not the focus on our work. For example, we mapped the Audience intent "mother" to Wikidata entity Q7560. We thus added 80 English aliases (and other multi-lingual aliases, not used this work) to the Activity and Audience intents.

We process customer review texts with entities as such:

(1) Remove punctuation, lowercase all words, tokenize and stem unigrams;
(2) Match intent aliases in text, and annotate the review with the corresponding entities;
(3) Take the union of all matches in a ASIN's reviews to label the ASIN (ASIN labels are binary and not weighted by the number of matches).

We then join on the ASIN labels extracted above with query-product affinity scores extracted from B, and output tuples of query keywords, ASIN, and set of intents. We limit the query-ASIN pairs based on the minimum B score, i.e. only pairs for which the score is greater or equal than the threshold are considered; we denote this option as -Threshold, for example -8 for scores with a threshold of 8. The maximum score is 15. For example, if a query "headphones" is associated with ASINs A and B with respective weights 5 and 12, and ASINs A and B are labeled with activities "running" and "working out" respectively, the final outputs are the training tuples (headphones, running, 5) and (headphones, working out, 12). We add a training instance per query-ASIN pair, which means the training dataset may contain multiple rows for the same query keywords. This labeling method is susceptible to noise from text mismatches, for example the activity "running" can be matched in a phrase such as "running nose"; we assume that, because of the volume of training data, our models can mitigate this noise implicitly.

3.2 Query Intent Prediction

We implement all models as convolutional neural networks, with the following architecture of: input one-hot words mapping to pre-trained unigram embeddings, followed by 1D convolutional layers of window sizes of 1 and 3, followed by two dense classification layers with the size of the number of entities to predict. We used Keras with Tensorflow backend [1, 6]. We trained word embeddings computed using the PySpark word2vec implementation on distinct search query keywords that occurred over one month; we tokenize on spaces and do not stem or lemmatize. Differences between the models come from 1) the number of output entities, and 2) the loss function used to train the network.

We evaluate training with two loss functions: 1) binary cross-entropy (BCE), $-\sum_{c=1}^{N} y_{o,c} log(p_{o,c})$ where N is the number of labels (intent entities), y is the binary training label for entity c, and p is the predicted probability for entity c; 2) B-weighted binary cross entropy (B), which we compute as the weighted average of

[1] http://www.wikidata.org

Short Name	B Thr.	# Distinct Labels	# ASINs	# Instances
ACT-15	15	85	1,137,392	11,347,028
ACT-8	8	89	2,042,206	197,261,588
AUD-15	15	44	1,394,858	12,554,462
AUD-8	8	45	2,701,914	223,770,625

Table 1: Datasets used for training. The dataset names indicates the B threshold used to select query-ASIN tuples, see Section 4.1 for details. Note that counts of ASINs and instances are approximate due to large data size (95% confidence interval).

Model	Dataset	Train Acc.	Test Acc.
BCE CNN	ACT-15	0.9566	0.9565
BCE CNN	ACT-8	0.9688	**0.9687**
B CNN	ACT-8	0.9579	0.9579
BCE CNN	AUD-15	0.9007	0.9005
BCE CNN	AUD-8	0.9233	**0.9233**
B CNN	AUD-8	0.9030	0.9030

Table 2: Loss value and binary accuracy after convergence, for training and testing data respectively. Note that loss values are not directly comparable between the BCE-trained CNN model and the one trained using B-weighted loss, because the latter is always less than or equal to the standard loss function. Accuracy values are comparable, since the models were trained on the same datasets.

normalized query-ASIN affinity scores, $-(s_o/S)\sum_{c=1}^{N} y_{o,c} log(p_{o,c})$ where s_o is the query-ASIN affinity score for the training example and $S = 15$ is the maximum possible score. All query-ASIN affinity scores are strictly positive, the higher the score the stronger the affinity, ranging from 1 to 15. The intuition behind the design of the B-weighted loss function is that, when predicting intents, the model should be penalized more for making mistakes on queries that are more strongly associated with an ASIN, and thus with the intents pertaining to that ASIN.

4 RESULTS

We first describe the datasets for training and offline results (Section 4.1). Then, using a separate dataset, we compare our models with a lexical-matching baseline, and include an evaluation on to expert-provided labels (Section 4.2). We then evaluate the impact of using weaker B signals to train, which expands our models' coverage to less frequent queries (Section 4.3).

4.1 Experimental Setup

We extracted entities from a sample of product reviews in the US marketplace. We then used a snapshot of the B dataset for one month to join queries, ASINs, B scores, and matched entities and output training tuples. Table 1 shows statistics for the training datasets. For the rest of this section, we refer to datasets with the convention <Entity Type>-<B minimum score>. For example, AUD-8 is the dataset with Audience labels thresholded at B scores greater than or equal to 8.

Model	Dataset used for training	Recall (vs lexical baseline)
BCE CNN	ACT-15	0.7910
BCE CNN	ACT-8	**0.8517**
B CNN	ACT-8	0.8438
BCE CNN	AUD-15	**0.9510**
BCE CNN	AUD-8	0.9063
B CNN	AUD-8	0.9273

Table 3: Model recall on query keyword data not used for cross-validation training (30,209 distinct queries in total), compared against text-matching baseline in query text.

Model	Training Dataset	Precision	Whole-query Accuracy
BCE CNN	ACT-15	0.7624	0.7472
BCE CNN	ACT-8	**0.8095**	**0.7849**
B CNN	ACT-8	0.7739	0.7494
Lexical Baseline	–	0.7278	0.7228

Table 4: Binary precision and accuracy relative to human annotations. The former computes precision per-label, the latter evaluates accuracy at a query label, i.e. all predicted labels need to match human-provided labels.

We trained by randomly splitting the full training set into 1.2 million partitions that fit in the GPU memory. We used mini-batches of 32 query-label instances, incrementally trained the model on each partition with 90% training data and 10% test data, and stopped training when the loss value would not decrease for 5 consecutive epochs, or up to 50 epochs per sample from the full data. We repeatedly re-sampled partitions and trained on the full data up to five times, or until convergence. Table 2 shows results of loss value and binary accuracy for train and test data after the model had converged. We observe that all models have similar test performance, and fit the Activity datasets better than the Audience data (95% vs 90% test accuracy). We also observe that all B-aware models have lower test accuracy than their corresponding BCE models: 95.79% for B CNN on ACT-8 vs 96.87% for BCE-CNN on the same data.

4.2 Latent Intent Prediction

We compare our models against a baseline that textually matches intent aliases. The baseline uses entities in the graph and their aliases, and counts all entities for which there is a match between the alias and the query tokens. This is the same method we used to extract entities from reviews, but applied to the query keywords. For our models' predictions, we used the same classifications threshold for all entities, because tuning the thresholds per-entity would over fit the baseline (Figure 3. We first observe that recall is competitive for all methods against the respective baselines. With the exception of the BCE CNN model on ACT-15, all methods exceed 80% recall. We observe that the Activity datasets result in lower recall across all methods compared to Audience. The main goal of our work is to predict latent intents which are not explicitly stated in the query. As such, we expect the models to predict more entities that are labeled in the query, thus precision is low for all models compared to the lexical matching baseline; thus, we evaluate precision using

Query Keywords	Entity Type	Predicted Entities
flannel pajamas for men	Activity	reading
bird cage accessories	Activity	cleaning
adidas mens pants	Activity	running
mafia 3 ps4	Activity	gaming
living proof shampoo	Activity	cleaning, showering
crayola super tip markers	Activity	writing
midi pencil skirt	Audience	daughter
adidas wrestling shoes men	Audience	son
lego hogwarts castle	Audience	child, son
versace crystal noir perfume	Audience	woman
toy story alien headbands	Audience	child, daughter
nikon telephoto lens	Audience	professional

Table 5: Examples of labels predicted by our B-aware methods for Activities and Audiences, respectively; these labels are not captured by the lexical matching baseline.

human judgment for the Activity classifiers, which we discuss in Section 4.3.

We evaluated precision using human annotators. We sampled 451 queries from the Activity dataset, and presented them together with predictions made by each model to three expert annotators (each query was evaluated by a single annotator). The annotator's task was to select a subset (at most all) of the labels predicted by the classifier; based on these responses we computed precision, considering as false positives the labels not selected by the experts, i.e. we report binary precision, with each label being considered an independent prediction (Table 4). The table also shows accuracy compared to human annotation, computed as the ration of exact matches, for all predicted labels, compared to human annotation. We observe that the BCE CNN model trained on ACT-8 data preformed the best, corroborating evidence from the recall result. All three machine-learned models outperform the lexical baseline. We discuss the result from B CNN in the section below.

Table 5 shows predictions made by the B CNN model on the two tasks, predicting Activities and Audiences. While these are qualitative data, it is interesting to note that one can hypothesize reasonable connections between the query and labeled activity. For example, often people may read books while wearing pajamas at home, buy branded clothing for sports such as running ("adidas mens pants"), and gift clothing ("midi pencil skirt") and footwear ("adidas wrestling shoes men") to their children.

4.3 Leveraging Behavioral Signals

In this section we explore the effect of using variable query-ASIN affinity scores has on classification performance. In the previous section, 4.2, we trained classifiers using only the highest query-ASIN affinity scores (15 out of 15). We then sampled an equal number of queries, but constructed with B scores varying between 8 and 15 (datasets ACT-8 and AUD-8). Note that, because of this relaxed constrain, the total number of instances in each of these datasets was 20 times larger than for the datasets constructed using only B scores of 15 (i.e. ACT-15 and AUD-15); this shows that a classifier trained on all this data would achieve a higher coverage of distinct queries in search traffic. However, to enable comparisons between models we sampled approximately 11 million instances to match the dataset size used in the previous section.

We compared the BCE CNN and B CNN models trained on the ACT-8 and AUD-8 datasets against the same text matching baseline (Table 3; B CNN models trained on the -15 datasets have identical performance with the BCE CNN models, which is to be expected, and we don't include those results). We observe that the BCE CNN model trained on the ACT-8 dataset achieves better recall that the other versions, but for Audience the BCE CNN classifier trained on ACT-15 outperforms other models. When comparing B CNN with BCE CNN on the -8 datasets, the relative performance differs between Activity and Audience data.

The B CNN variant was second in terms of precision and matched on accuracy with the BCE CNN ACT-15 model when compared to expert annotations (Table 4. We observe that both B CNN and BCE CNN models trained on ACT-8 outperformed the models trained on ACT-15, and the BCE CNN model outperforms the B model, which is consistent with the cross-validation performance shown in Table 2. All models outperform the baseline. This data invalidates our second hypothesis, that leveraging the B affinity score between queries and ASINs, we can inform the classifier which prediction mistakes are more costly when training.

This may indicate that, if the search engine does not index the latent keywords we extract from customer reviews, they do not affect ranking and thus the B score, since customer interaction is affected by position bias. Indeed, using the ACT-8 and AUD-8 datasets, we computed the Spearman correlation between the B score and the number of entities labeled from customer reviews using a uniform sample of 20,000 instances from each dataset. We obtained correlation scores of 0.13 and 0.11 for ACT-8 and AUD-8, respectively, which is close to no correlation at all.

5 CONCLUSION

We explored predicting latent intents in queries, and focus on intents that are not explicitly indexed in the product catalog. We leverage intents that describe context-of-use facets that model customer context, which are expressed in some search queries and in customer reviews, but are not intrinsic product attributes indexed in the catalog. Our method combines entity extraction from unstructured text and training predictive models that leverage query-ASIN affinity scores computed from previous customer behavior. Our results show that we can reliably predict these intents from review queries, but also that there is potential for improving relevance relative to these interests by including these intents in the relevance score. To improve our models' performance, we plan to use a larger scale crowdsourced survey to expand on the human annotations and to tune classification thresholds per-entity. We will also add character tri-grams as input to improve robustness to incorrect spelling. A second improvement we plan is to apply word sense disambiguation before query intent training and prediction in order to separate intents associated with distinct meanings of queries, as presented our method does not model query ambiguity.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard,

Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

[2] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[3] Sanjay Agrawal, Kaushik Chakrabarti, Surajit Chaudhuri, and Venkatesh Ganti. Scalable ad-hoc entity extraction from text collections. *Proceedings of the VLDB Endowment*, 1(1):945–957, 2008.

[4] Alberto Bartoli, Andrea De Lorenzo, Eric Medvet, and Fabiano Tarlao. Active learning of regular expressions for entity extraction. *IEEE transactions on cybernetics*, 48(3):1067–1080, 2018.

[5] Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian-Tao Sun, Enhong Chen, and Qiang Yang. Context-aware query classification. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 3–10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: 10.1145/1571941.1571945. URL http://doi.acm.org/10.1145/1571941.1571945.

[6] François Chollet et al. Keras. https://keras.io, 2015.

[7] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

[8] Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. Neuroner: an easy-to-use program for named-entity recognition based on neural networks. *arXiv preprint arXiv:1705.05487*, 2017.

[9] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165 (1):91–134, 2005.

[10] Ramanathan Guha, Rob McCool, and Eric Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709. ACM, 2003.

[11] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.

[12] Reiner Kraft Homa B. Hashemi, Amir Asiaee. Query intent detection using convolutional neural networks. http://people.cs.pitt.edu/~hashemi/papers/QRUMS2016_HBHashemi.pdf, 2009. (Accessed on 12/18/2018).

[13] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 471–480, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526773. URL http://doi.acm.org/10.1145/1526709.1526773.

[14] Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the user intent of web search engine queries. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1149–1150, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242739. URL http://doi.acm.org/10.1145/1242572.1242739.

[15] Rahul Nair, Killian Levacher, and Martin Stephenson. Towards automated extraction of business constraints from unstructured regulatory text. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 157–160, 2018.

[16] Filip Radlinski, Martin Szummer, and Nick Craswell. Inferring query intent from reformulations and clicks. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1171–1172, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772859. URL http://doi.acm.org/10.1145/1772690.1772859.

[17] Sifatullah Siddiqi and Aditi Sharan. Keyword and keyphrase extraction techniques: a literature review. *International Journal of Computer Applications*, 109 (2), 2015.

[18] Chao-Yuan Wu, Amr Ahmed, Gowtham Ramani Kumar, and Ritendra Datta. Predicting latent structured intents from shopping queries. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1133–1141, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4913-0. doi: 10.1145/3038912.3052704. URL https://doi.org/10.1145/3038912.3052704.

[19] Jiwei Zhong, Haiping Zhu, Jianming Li, and Yong Yu. Conceptual graph matching for semantic search. In *International Conference on Conceptual Structures*, pages 92–106. Springer, 2002.