The Last Mile: Taking Query Language Identification from Model Ready to Production

Tracy Holloway King
tking@adobe.com
Adobe Inc.
San Jose, CA, USA

Chirag Arora
charora@adobe.com
Adobe Inc.
San Jose, CA, USA

Francois Guerin
guerin@adobe.com
Adobe Inc.
San Francisco, CA, USA

Sachin Kelkar
kelkar@adobe.com
Adobe Inc.
San Jose, CA, USA

Judy Massuda
massuda@adobe.com
Adobe Inc.
San Francisco, CA, USA

ABSTRACT

Taking a custom-made machine-learned model from model-ready to a user-facing feature requires significant effort: This last mile can seem like half the trip. This paper describes the many product and applied science and engineering decisions that had to be made in order to integrate query language identification (QLI) into Adobe Stock's asset search capabilities, even after the custom-trained QLI model was available. We hope that the detailed discussion of the decision making process and the decisions made for this particular feature will help others when integrating different types of machine learned models to improve the eCommerce user experience.

KEYWORDS

language identification, query understanding, eCommerce search

ACM Reference Format:
Tracy Holloway King, Chirag Arora, Francois Guerin, Sachin Kelkar, and Judy Massuda. 2021. The Last Mile: Taking Query Language Identification from Model Ready to Production. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'21).* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

In order to rapidly find information, search engine users have become accustomed to conveying their information needs as short queries. These queries are generally between one and five words long. With natural language processing (NLP) models traditionally targeting sentences, e.g. the newspaper texts in the LDC's Penn Treebank [8], the need to process 1-5 word queries has given rise to a subdomain of NLP: short text understanding [1, 12, 15]. Queries are not just shorter than sentences, they also follow different syntactic rules, often dropping function words (e.g. *dog beach*), defaulting to noun phrases instead of sentences (e.g. *woman running*), and having unusual word order as users refine their queries by adding to the end of the query (e.g. *dress* ⟶ *dress red*).

Search has become a key way for eCommerce customers to find what they are looking for [14]. The domain-specific nature of eCommerce queries means that NLP models have to be adapted both for short text and for the specific domain. NLP has a long history of research on domain adaptation, with a recent resurgence in interest driven by the availability of large-scale, pre-trained language models [3, 5] that can then be fine-tuned for specific domains and tasks.

This paper focuses on a different aspect of integrating NLP models into eCommerce use cases: What types of business logic have to be considered when integrating a model into production once a domain-specific, short-text model has been trained for the task? We discuss the integration of a query language identification (QLI) model into Adobe Stock's search system.[1] [6] discuss the importance of language identification in query processing for eCommerce, showing how adding language identification to Amazon product search improved key user metrics. The integration described here for Adobe Stock QLI involved three major product decisions and three major applied science and engineering decisions (section 4). Although the focus of this paper is on a specific type of model, QLI, for a specific eCommerce product, Adobe Stock, we hope that discussing the process around the model integration and the decisions made will help others to more quickly move from initial ML model development to product integration.

2 QLI PRODUCT FEATURE

ECommerce companies often have different sites (e.g. as indicated by urls) for different locales. These locales' settings govern decisions such as what items are available, what currency is displayed for pricing, and, of primary interest here, what language is expected for queries. [6] demonstrate how customers do not always enter queries in the expected, default language of the site and that treating these queries as the default language results in fewer or irrelevant search results. Adobe Stock is a marketplace where contributors upload content such as photographs, illustrations, and short videos that buyers purchase to, for example, create marketing materials. Although the assets themselves are largely language independent,

SIGIR eCom'21, July 15, 2021, Virtual Event, Montreal, Canada

[1] We would like to thank Ajinkya Kale and Ritiz Tambi who designed and trained the custom QLI model [13], Alex Filipkowski and Kate Sousa whose data design expertise was crucial for evaluating the model, and Madhura Das for detailed pre/post analysis.

For reasons of Adobe proprietary data, we are unable to provide exact statistics for some analyses. Given the focus of the paper on the process of integration, we believe that the reader will benefit from the paper even without these details.

the buyers' search queries are in a particular language.[2] The Adobe Stock search engine processes the query with an NLP pipeline that includes tokenization, lemmatization, stop word identification, and mapping from words to language-independent concepts.

An analysis of queries issued in non-English locales showed that these sites often had 20% of queries in English and in some cases, such as in Korea, over 50% of the queries were in English. Comparison of the results for English queries in non-English locales (e.g. DE, KO) to the same queries in English locales (e.g. US, UK) showed that the result sets were much smaller or even empty. This was due to the queries not being processed as English text. In addition, non-English queries were responsible for many of the null queries in English language locales. Based on this analysis, we decided to integrate language identification into the query processing pipeline.

3 QLI MODEL

The first step was to create a machine-learned QLI model. This was reported in detail in [13] and is summarized in this section. [6] and [10] also custom-train QLI models for eCommerce sites. Language identification systems are generally trained on well-edited text. They use signals such as: known words including closed class words; character n-grams; punctuation, spaces, and upper vs. lower casing. These general purpose language identification models work well for longer, similar style texts. However, their performance drops significantly on shorter texts such as search queries and shorter tweets [2, 4, 11], as do many NLP components [1]. The degradation is often severe because short text is missing many of the signals that are seen in the training text [1, 12, 15]. In search queries, capitalization is meaningless, there are frequent misspellings, punctuation is missing or used differently, and closed class words are dropped.

We only wanted to identify languages where we can use the language identification to improve the customer experience, e.g. by applying the correct NLP. So, we wanted the QLI to focus on a handful (<20; currently 8: EN, DE, FR, IT, ES, PT, JA, KO) languages. Identifying only 8 languages also helps to overcome the issue of having less signal due to the short text in the queries.

To train the model we automatically created a large, weak-labeled training set for the 8 languages and then a smaller, human-annotated evaluation set. The training data was created by taking a seed dictionary of known language terms and finding their nearest-neighbors in the query logs. We then found the nearest-neighbors to that larger set of labeled query terms and used a voting schema to determine their language. The result is a ~664K query set with weak labels for each language. We then trained a CatBoost language identification model. To evaluate the QLI model, we manually annotated ~65K queries through a crowd-sourcing task.

4 INTEGRATION DECISIONS

As is often the case in integration ML models, once we had the model available, our task of improving search results via QLI had just begun. First, we had to make key product decisions. Then we had to integrate the QLI model into the query NLP pipeline, requiring applied science and engineering decisions. This section discusses these decisions in detail since they highlight the complexity of going from having a custom-trained model to having a customer-facing feature in an eCommerce search experience.

4.1 Product Decisions

There were three product decisions to be made. Importantly, these decisions were made by the product manager, using data provided by the applied science and engineering team.

Should QLI be used just in non-English locales or also in English locales? The largest degradation in user experience occurred in non-English locales because these locales had from 20% to over 50% English queries and most of those queries had degraded (e.g. only a small subset of the relevant assets) result sets due to incorrect language processing. For the first version, we decided to only apply QLI in non-English locales. Detecting non-English in English locales was left for future versions.

Should QLI apply to all queries or only to ones with few results (e.g. <100)? Comparison of the results for English queries in non-English locales to the same queries in English locales showed a drop in recall even though this drop was not always to less than one page (100 results). This indicates that users were only seeing a fraction of the relevant assets due to the incorrect language processing. So, we decided to use QLI on all queries regardless of result size. The low latency (<5ms) of the QLI model makes this feasible.

Should we identify only English as an alternative language or any of the QLI languages? Examination of the queries identified as not in the language of the locales showed that the vast majority (high 90 percent) were in English. To keep the logic simple with maximum benefit to the users, we decided to only look for English queries.

Thus, our product specifications were to apply QLI to all queries in non-English locales and determine whether the query was in English, in which case the English NLP pipeline was applied, or not, in which case the locale language NLP pipeline was applied.

4.2 Applied Science and Engineering Decisions

Once we had the product specifications, we had to modify the query NLP pipeline to create the new user experience.

Logic for Calling QLI The first decision involved determining the overall logic for the NLP pipeline. This is shown in Figure 2 and directly reflects the product requirements. Steps 4–6 prefer the locale language over English if the QLI model had high confidence for both languages. If the locale language is known to the QLI model, a high confidence result is treated as the locale language regardless of the score for English (step 4). If this fails, we check for high confidence English (step 5). If neither of those conditions is met (step 6), the query is treated as the locale language. This helps ensure no degradation in user experience.

Error Compensation The second decision involved determining what processing, if any, was needed to compensate for errors in the QLI model. Since the QLI model was trained on queries that had been lower-cased and tokenized with the same NLP pipeline as used in production, no changes had to be made to those steps. We ran a stratified sample of queries from ~10 non-English locales and examined the queries which were identified as English. There were three classes of queries that the QLI model incorrectly identified as English. The first errors were queries with digits (e.g. *happy new*

[2]Buyers can also find assets through image-similarity search which uses image processing models [7, 9]. Image similarity can be used in conjunction with textual queries. The image-similarity search feature is orthogonal for the QLI feature described here.

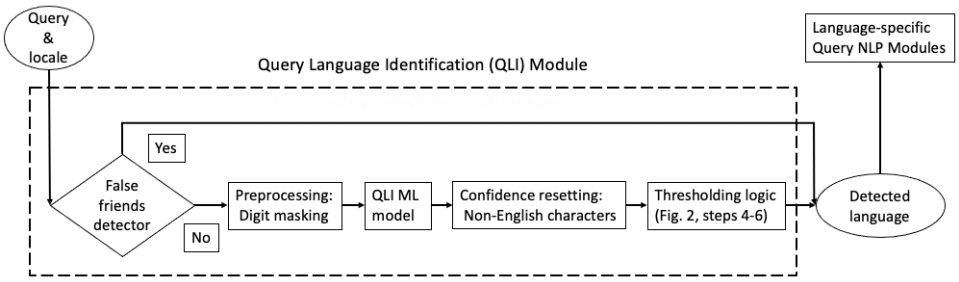

Figure 1: QLI Architecture: Queries with locales are passed to the QLI module. False friend queries are skipped. Digits are masked. The QLI ML model provides confidence scores for 8 languages. Adjustments are made based on special characters. The thresholding logic is run. The detected language is passed to the query NLP modules (e.g. lemmatization, stop words, mapping to language-independent concepts).

1. let *loc_lang* be the default language of the locale and *qry* be the user query
2. let *thresh_loc_lang* & *thresh_en* be confidence thresholds for QLI set on a per language basis
3. pass *qry* and *loc_lang* to QLI
4. if *loc_lang* ∈ { FR DE IT ES PT KO JA } && the confidence for *loc_lang* > *thresh_loc_lang*, analyze *qry* as if it is *loc_lang*
5. elsif the confidence for EN > *thresh_en*, analyze *qry* as if it is EN
6. else analyze *qry* as if it is *loc_lang*

Figure 2: Algorithm for high level QLI thresholding logic

year 2020). To alleviate this, all digits were removed before calling QLI. The second errors were queries with a mix of unaccented Latin characters with accented Latin or non-Latin characters. These were sometimes identified as English because the QLI model training data (section 3) from the Adobe Stock US site had some non-English in it. To alleviate this, we introduced a post-processing module that resets the confidence for English to 0 if the query contains non-Latin characters (e.g. Cyrillic, Hangul) or select Latin characters very rarely used in English (e.g. łżąśęńß).[3] The third errors were false friend terms. Even with the high thresholds set for English and the locale language, there are terms that belong both to the locale language and English but with different meanings. These are referred to as "false friends" in the linguistic literature. For example, *pain* in French means 'bread' and *gift* in German means 'poison'. In the context of longer queries (e.g. French *pain étalage* 'bread display', English *back pain*), the QLI model correctly identifies the language, but as single word queries, which are extremely common, the QLI model has difficulty. To avoid misanalyzing single-word queries with these terms, we implemented an allow list that specified a small (<<100) list of terms for each locale as being in the locale language. This list was seeded with lists of such terms from language web sites and then updated based on feedback from the manual annotation task (section 5).

[3]The current system aims to treat multi-language, code-switched queries as being in the language of the locale. Splitting these into words from multiple languages is left as future work. See [16] on ways to approach this.

Thresholds The third decision involved setting the confidence thresholds for the QLI model to be used in the algorithm in Figure 2. The goal was to have high precision identification of English so that basically no queries in the locale language were identified as English. To determine the thresholds we took a moderate (low thousands) sample of queries from ~10 non-English locales and calculated the QLI model probability for the locale language (*thresh_loc_lang*) and for English (*thresh_en*). We then frequency sorted them and examined them from most to least frequent to determine a threshold. For all languages identified by the QLI model, a threshold of 0.8 was set for *thresh_loc_lang*. For the English threshold, Latin character languages (e.g. DE, FR, ES) had a relatively high threshold of 0.8 for *thresh_en*, while for non-Latin character languages (e.g. JA, KO, RU) the *thresh_en* threshold was a much more aggressive 0.5 because the character set differences resulted in the QLI model being able to easily differentiate the languages. For English locales (e.g. US, UK), *thresh_loc_lang* = *thresh_en* by definition and these were set to 0; this effectively disables QLI for English locales per the product requirements (section 4.1).

The resulting system architecture is shown in Figure 1.

5 EVALUATION, RESULTS & FUTURE WORK

Due to relatively low traffic in some of the non-English locales combined with the fact that only a subset of the queries would be affected by the QLI feature, the QLI feature was not AB tested but instead launched to production and then analyzed via pre/post analysis. This meant that we had to be certain the feature would not cause any degradation in search result quality for users. We focused on extrinsic evaluation, i.e. evaluation of the effect of QLI on the search results. Lightweight intrinsic evaluation of the QLI output was done as part of the threshold setting (section 4.2) and extensive intrinsic evaluation was done on the underlying QLI model [13].

5.1 Human Annotation Task

To increase our confidence in the quality, we ran a human annotation task.[4] We restricted the annotators to countries with the relevant languages and wrote the instructions in those languages. For single-word queries that occurred more than 10 times and were

[4]We used the Appen platform with a general crowd restricted to "level 2".

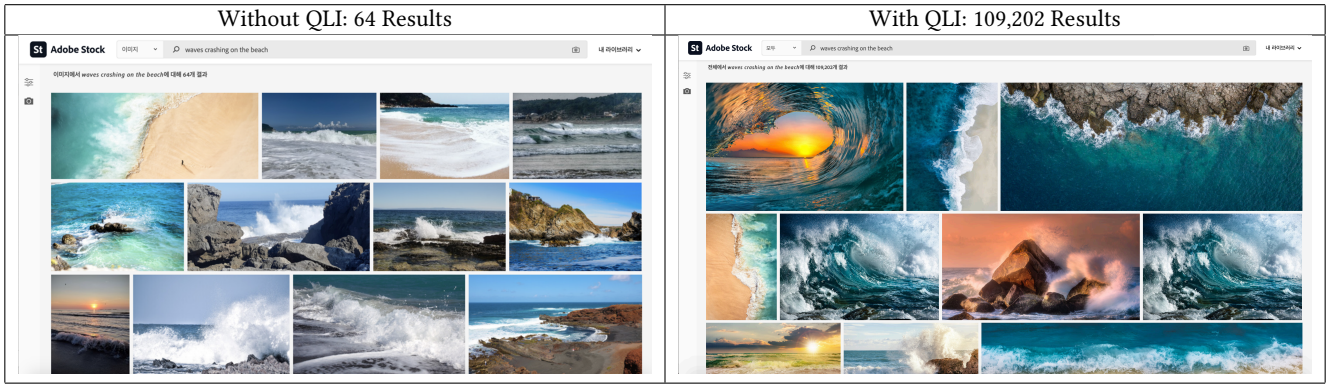

Figure 3: Results for English query *waves crashing on the beach* on the Adobe Stock Korean site. Both result sets show relevant images. Without QLI there are only 64 results. With QLI there are 109,202 results.

detected as English, we took the top 20 Adobe Stock results with the QLI feature and without it. If these results varied by more than 20%, we had the results human annotated. The task was a side-by-side comparison of the two images at a given rank (e.g. the first result with QLI compared to the first result without it) returned for a query. Each task (query + 2 images) was judged by three annotators. The comparison was a 5-point scale from *left image is much better* to *right image is much better* with an option to declare the query to be uninterpretable. We judged 1.9K tasks for FR, 1.6K for ES, and 4K for DE. Table 1 shows that for very few queries was QLI incorrectly identifying the query as English. All queries where the results were better without QLI were examined. Most of these were false friends and were added to the allow list (section 4.2) so that they would be treated as the locale language in the final version (Figure 1). The human annotation task results are likely a lower limit on the improvement users experience because it only included single word queries, which are harder to detect language on.

Table 1: Human annotation results comparing results without QLI (previous production) to ones with QLI applied for single-word queries identified as English.

Locale language	Better without QLI	Better with QLI	Equally good
French	1.2%	54.5%	44.4%
German	0.3%	50.4%	49.3%
Spanish	12.7%	44.9%	42.4%

5.2 Pre-/Post-Analysis

After the QLI feature had been in production for a month, we ran an analysis of the impact. We looked at three metrics:

(1) The percentage of queries with no and low (<100) results (hypothesis: there should be fewer null and low result queries because of the improved NLP processing of English queries)
(2) Forward action rate (hypothesis: if the new results are good, the forward action rate, including clicking on and downloading assets shown in the search results, should be the same or better than before)
(3) Percentage of queries detected as English (hypothesis: this should be similar to our offline analysis and is a sanity check that the QLI system is working properly).

All three metrics supported the hypotheses and confirmed the decision to launch the QLI feature. As expected, there was a modest decrease in the null result rate and a more significant one in the low result rate, the forward action rate remained steady, and the percentage of queries detected as English was in line with the initial offline analysis. An example of the change in results is shown in Figure 3 for the English query *waves crashing on the beach* on the Korean Adobe Stock site. Although the search results are relevant both with and without QLI, there are many more results to choose from with QLI (64 vs. 109,202).

5.3 Conclusions and Future Work

This paper described the multiple product, applied science and engineering decisions that had to be made in order to integrate QLI into Adobe Stock's asset search capabilities. Even after the custom-trained QLI model [13] was available, decisions had to be made and evaluations conducted. This is a common state of affairs in integrating machine learned models into eCommerce production applications and we hope that the detailed discussion of the decision making process and the decisions made for this particular feature will help others when integrating models to improve the eCommerce user experience.

Future work includes integrating the model into other Adobe search products: The pattern of having English language queries in non-English locales extends to other Adobe products, such as search over the general Adobe help content. We also plan to improve the English locale experiences for Adobe Stock by identifying non-English queries, such as Spanish queries on the US site.

The QLI model integration into Adobe Stock highlighted several areas for improvement of the machine-learned QLI model (section 3). These include making the model more robust to language-specific characters and the presence of digits in the queries. In addition, we plan to extend beyond the current 8 languages to include other frequently queried languages such as Russian and Chinese. Finally, we hope to tackle multi-lingual, code-switched queries [16], which are particularly common in the Japanese and Korean locales.

REFERENCES

[1] Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How Noisy Social Media Text, How Diffrnt Social Media Sources?. In *International Joint Conference on Natural Language Processing*. 356–364.

[2] Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Proceedings of NAACL HLT 2010*. 229–237.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901.

[4] William B. Cavnar and John M. Trenkle. 1994. Ngram-based text categorization. In *Proceedings of the Third Symposium on Document Analysis and Information Retrieval*.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805

[6] Qie Hu, Hsiang-Fu Yu, Vishnu Narayanan, Ivan Davchev, Rahul Bhagat, and Inderjit S. Dhillon. 2020. Query Transformation for Multi-Lingual Product Search. In *ECOM20: The SIGIR 2020 Workshop on eCommerce*. CEUR Workshop Proceedings.

[7] Adobe Inc. 2020. Find Assets Similar to Another Asset. User help documentation for Adobe Stock image similarity search https://helpx.adobe.com/stock/help/find-similar-assets.html.

[8] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating Predicate Argument Structure. In *ARPA Human Language Technology Workshop*.

[9] Aashish Kumar Misraa, Ajinkya Kale, Pranav Aggarwal, and Ali Aminian. 2020. Multi-Modal Retrieval using Graph Neural Networks. *CoRR* abs/2010.01666 (2020). arXiv:2010.01666

[10] Sweta Sharma, Vijay Huddar, Ishita Aggarwal, Namrata Khoriya, Vishnu Narayanan, Atul Saroop, and Rahul Bhagat. 2021. Query language identification with weak supervision and noisy label pruning. In *The Web Conference 2021 Workshop on Multilingual Search*.

[11] Penelope Sibun and Jeffrey C. Reynar. 1996. Language determination: Examining the issues. In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*. 125–135.

[12] Xiangyan Sun, Haixun Wang, Yanghua Xiao, and Zhongyuan Wang. 2016. Syntactic Parsing of Web Queries. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1787–1796.

[13] Ritiz Tambi, Ajinkya Kale, and Tracy Holloway King. 2020. Search Query Language Identification Using Weak Labeling. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, 3520–3527.

[14] Manos Tsagkias, Tracy Holloway King, Surya Kallumadi, Vanessa Murdock, and Maarten de Rijke. 2020. Challenges and Research Opportunities in eCommerce Search and Recommendations. *SIGIR Forum* 54, 1 (June 2020), 1–23.

[15] Zhongyuan Wang, Haixun Wang, and Zhirui Hu. 2014. Head, modifier, and constraint detection in short texts. In *Proceedings of the International Conference on Data Engineering*. 280–291.

[16] Yuan Zhang, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A Fast, Compact, Accurate Model for Language Identification of Codemixed Text. arXiv:1810.04142v1.