Smart Query Reformulation for Low Inventory Items in E-Commerce

Ishita Khan
ishikhan@ebay.com
ebay Inc
San Jose, CA

Liyang Hao
liyhao@ebay.com
ebay Inc
San Jose, CA

Raffi Tutundjian
rtutundjian@ebay.com
ebay Inc
San Jose, CA

Zhe Wu
zwu1@ebay.com
ebay Inc
San Jose, CA

ABSTRACT

Understanding user's latent intent from search queries is a long-standing multi-faceted problem in information retrieval. Many of the existing e-commerce search engines rely on token matches between all tokens in the user query (including punctuation, prepositions), more generally tokens and their corresponding lexical and structural expansions, with textual and structural data in documents as the primal retrieval strategy. This strategy makes the retrieval task for low inventory queries particularly challenging; for example, when users paste long titles from other e-commerce sites as their queries or specify the query with natural language syntax. We use the term *Null & Low* (N&L) as an umbrella to cover zero or low recall queries, long queries, over-specified queries, or queries with natural language syntax, a query family where there exists the problem of low inventory when items are retrieved by matching all the query tokens to item titles. Given a N&L query in its raw form (ideally in the tail of the e-commerce search traffic, essentially with low recall) the problem we address in this scope is to find an alternate form of the query by reformulating or dropping tokens in a way that does not alter the query's core intent. The goal of this is to improve relevant recall while making a minimal loss at precision in the search result page (SRP) of the raw user query. We put this problem into a language modeling framework and develop a deep Seq-to-Seq model that is trained on successful query transitions from e-commerce user behavioral logs and performs query reformulation for N&L queries in an online fashion. Upon showing significant improvements of the method in offline evaluation over baseline, an A/B test was conducted in the commercial search engine which achieved statistically significant lifts in conversion metrics for live traffic and millions of buyers who issued a N&L search query. The model has been deployed to serve live traffic in the search ecosystem serving N&L user experience.

KEYWORDS

query reformulation, query relaxation, query recommendation, generative language model, sequence to sequence model, biLSTM, transformer, e-commerce, search, recall, relevance

1 INTRODUCTION

Search in a generic sense aims to optimize for two metrics: precision and recall. Precision is the fraction of retrieved items that are relevant to the search query; recall is the fraction of relevant inventory included in the results. Improving one metric usually has some notion of compromise over the other. In e-commerce, since there are over a billion listings in the inventory, it is intractable to exhaustively score each item listing for every query within the serving latency requirements. Therefore, the first step upon receiving a query is retrieval. The retrieval system returns a short list of items that best match the query using various signals, usually a combination of machine-learned models and human-defined rules. After reducing the candidate pool, the ranking system scores and ranks all items. One effective method to process a given e-Commerce search query is to construct a boolean recall expression that determines the items in the recall set for a query. The boolean recall basis performs token and rewrites matching for every query tokens to item titles to retrieve a relevant inventory set, which consequently is passed to a machine learned ranker.

With this high level query retrieval-ranking pipeline used at an e-commerce engine [31][20], user is bound to encounter negative search experience for the family of queries where the retrieved result size is too limited. The term *Null & Low* query is thus used to address a set of queries with zero or low recall (referred in the rest of the text as N&L queries), due to reasons varying from being overly specific, too long, miss-spelled or differently spelled than the popular form of the queries. For such queries, a default item retrieval approach of matching *all* of the query tokens to document titles is not ideal. In e-Commerce, query recall size follows a power law distribution, meaning that although a majority of queries have abundant matching items and at the same time, a non-trivial number of queries have much smaller recall size or matching inventory. So there is a good opportunity to improve over the low inventory of these by doing smart query reformulation, consequently driving higher search conversions for this family of queries. One straightforward baseline N&L recovery approach is to match an empirically determined percentage of query terms to item titles, since matching all query terms for N&L queries give insufficient results. Although this baseline method has the advantage of retrieving a large recall set, it has to compromise with result precision in terms of query-item relevance, as the query tokens matched to item titles are not learned in a query intent-driven manner. As an example, for the query *3 foot tall doll* the baseline approach will retrieve an item that matches with any two query tokens to item's title, irrespective of word order or linguistic concerns in the matching task. This

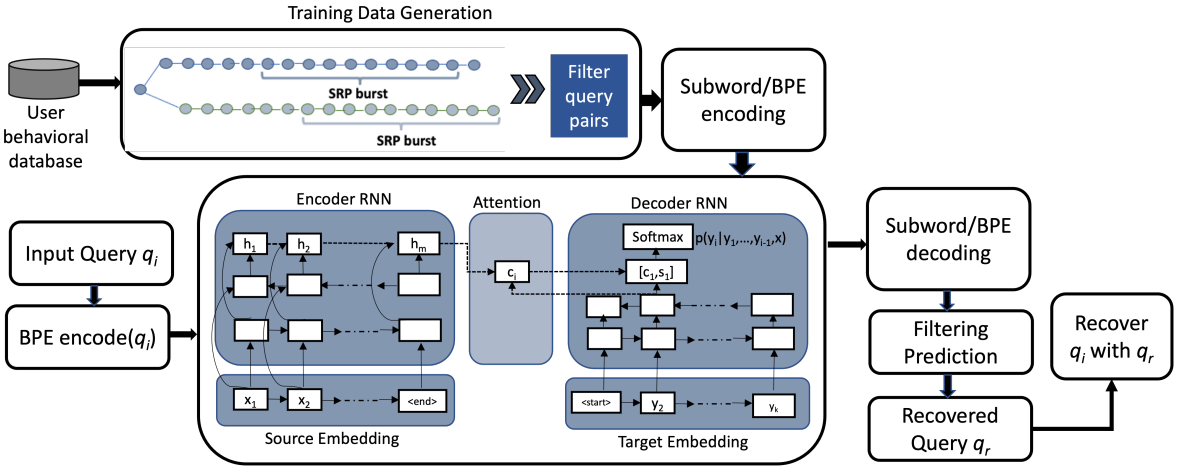

Figure 1: Schematic diagram showing the pipeline for training and query-time inference of over-specified queries with a seq-to-seq model

will return very irrelevant items for the query in many cases not necessarily even a doll, which is the core query intent.

To this end we propose a new model that performs a smarter recovery for N&L queries using query reformulation with the goal of increasing relevant recall. The proposed solution has to tackle two vital concerns in order to prove its merit over the existing baseline: A. the performed query reformulation needs to be intelligent enough to understand user's original intent and e-commerce linguistic perspectives, and B. the run-time performance of the model has to comply with the latency requirements of a live e-commerce search traffic for N&L queries. Note that since the query family we target are mostly in the tail, a cache oriented latency-expensive algorithm is not a feasible solution, since the cache size will be too high to cover a decent N&L traffic share. With this problem scope and the goal, we put this problem into a language modeling framework, specifically into the realm of next-token-prediction algorithms and propose a deep Sequence-to-Sequence (referred as Seq2Seq in the rest of the paper) model that is trained on successful query transitions from user behavioral logs and performs query reformulation for N&L queries in an online fashion. The backbone of the model is user query transition pairs in search sessions that result in successful user engagement and are within our expected range of query intent change, sampled from aggregated behavioral logs in search engine which is one of the largest e-commerce databases in the world. Neural Machine Translation (NMT) models have gained widespread popularity in the past decade due to their simplicity of underlying concept and generalizability specially for long word sequences. The Seq2Seq architecture [29][2] of NMT models has shown to be very effective in a variety of language modeling tasks under the next token or sentence prediction umbrella, such as machine translation, speech recognition, text summarization, and question answering [22]. To the best of our knowledge, this is the first work that puts e-commerce query reformulation/recovery problem into the Seq2Seq modeling framework. For the reformulation, we aim to get a relaxed form of the query by three possible actions: 1. term dropping, 2. term reordering, and 3. term rewording with synonymous terms learned from search user vocabulary. The goal of the reformulation is to improve recall while making a minimal loss at precision in the search result of the original raw user query by using the alternate query form without altering the query's original intent.

We perform evaluation of the Seq2Seq model in three phases: first, we look at offline model level metrics (precision, recall, F-score) in being able to predict the query level and token level reformulation (term dropped or altered), where the target is user issued successful query reformulations in e-commerce user logs. At second phase, we perform an end-to-end evaluation of the Seq2Seq model as a component in a typical e-Commerce query processing workflow and compare the relevance of the result set with a baseline method. The **baseline model** we use for this work uses an empirically learned fraction x and for every query it retrieves an item that matches *any x%* of the of query terms to the item title terms. Essentially this is a random token dropping approach that learns the number of tokens to drop from collected samples. In the third phase, we conduct human judgement as well as an online A/B test with live traffic, exposing the new experience with Seq2Seq query reformulations to millions of live e-commerce users. In all three phases of evaluation, we achieve high model accuracy and significant improvement over the baseline in improving the search result page (SRP) experience for N&L queries by increasing relevant recall. The key novel contributions of this paper are:

- Formulating the query reformulation problem into the generative language modeling framework to improve the N&L search experience in e-commerce.
- Application of a Seq2Seq model for query reformulation to improve N&L buyer experience in an e-commerce search engine at run-time.
- Iterative model augmentation and improvement shown by making changes in training data construction (enhancing local neighborhood of an SRP) and different model architectures (from biLSTM to transformer).

2 RELATED WORK

The e-commerce search domain has been gaining much research attention for the inherently non-trivial problems that comes with it: query expansion/rewriting, query reformulation, query intent detection, personalization, and query-item ranking to name a few. There is extensive research work on query item retrieval with a query token expansion approach in the literature [34], [3], [11], [21], [14], [1], [20] and [24]. Query reformulation is a different problem area than these query expansion approaches as it alters the query keywords issued by the users.

Hirsch et al [10] has an in-depth survey of query reformulation performed by users based on the query logs of search engine. A few different approaches to address the generic query reformulation problem can be studied in literature [6], [19] and a body of work in this area can be found in the web search domain [9], [7], [8], [12], [13], [23]. The actions we perform for the query reformulation (term dropping, term reordering, or term alteration) have been long studied for web session retrieval in studies where the goal was to characterize reformulation types/actions [12], [13], [23]. There is also work on predicting whether a query will be reformulated by the user and what type of reformulation might be performed under the query performance prediction domain [23], [12], [10], [7].

A recent work from Yaxuan et al [33] also uses RNN and LSTMs for query refformulation, however they focus on ambiguous query rewrite which is different from our focus area of N&L queries, which is usually over-specified. A work from Wei at all [17] focuses on ranking relevant queries using a pre-trained encoder-only model for query reformulation problem. Only few research work focus on the query reformulation approaches in low recall queries in e-commerce domain. The only query-log based study of user query reformulations for null queries in e-commerce search that we know of is that of Singh et al [28]. A follow up work from the authors added taxa constraints to sub-query and retrieved items from a pre-built database of historical purchases for null and low queries. The learned distribution of taxa constraints from the historical data are then added to each sub-query for secondary retrieval [27]. Tan et al [30] developed a query rewrite system that reformulate the original query into multiple alternative queries and apply it to the low recall queries to increase recall by performing tagging of several properties of the query keyword (parts-of-speech, entities, unit-of-measure, and phrases).

3 ARCHITECTURE

The full end-to-end design for training and inference of a Seq2Seq model for the purpose of query reformulation to recover Null & Low queries is shown in Figure 1. In the training data generation process we generate candidate (query, reformulation) pairs from past e-commerce user behavioral sessions by looking at consecutive SRPs within 2-hop immediate neighborhood of a SRP query. A (query, reformulation) pair is considered as a valid candidate for training if it results in a successful user engagement in the SRP where the reformulated query is issued. Depending on the type of the Seq2Seq model (term dropping and/or altering), a filtering step based on token comparison between the query and the reformulation is also done on the candidate pairs to graduate them to be part of the model training data. Details of training data generation is described in Section 4.1.

The raw data from user logs are encoded with subword/Byte Pair Encoding(BPE) [25] to enhance the model vocabulary in a robust way, described in Section 4.2. The details of a bi-directional encoder-decoder model with LSTM cell are in Section 4.3.1 and a transformer encoder-decoder model are discussed in 4.3.2. Results for the biLSTM models are in 5.1 and 5.2, and results and discussions for the transformer models are in 5.3. Furthermore, the comparison between the legacy system is discussed in 5.4.

At inference time (second horizontal panel of Figure 1) we feed an input query q_i to the BPE encoder, and pass the encoded form to the trained Seq2Seq model. The model predicts a reformulated output query q_r with a confidence score c_i. We then pass the output q_r through BPE decoder and perform a set of post-processing/filtering to discard bad model predictions. The final filtered query is then used for further query expansion steps [20] to retrieve the recall set for the original user query q_i.

4 METHODOLOGY

4.1 Training Data Generation

For data generation for the Seq2Seq model training, we sample 4 weeks of user behavioral data in query logs to gather successful query transition pairs. In a given user session, consecutive SRPs (ascending order by timestamp) are grouped into *bursts* where transition time inside a given burst is no greater than 10 minutes. Given a burst of SRPs, transitions are generated by connecting consecutive SRPs within 1-hop and 2-hop neighbors of a given query. For a user query q, if (q', q'') are the reformulated queries from q in the same SRP burst immediately after q (i.e., q, q', q'' are in a sequence), then we consider (q, r) as a valid candidate reformulation to be used for model training when:

$$aggregatedEngagement(r) \geq aggregatedEngagement(q),$$
$$aggregatedEngagement(q) > 0, \quad (1)$$

and r is either q' or q''. Equation 1 basically results in a transition with a higher user engagement, and is considered a valid candidate reformulation for generating the training data set. The *aggregatedEngagement* function is used to represent different click and sale based user engagement signals.

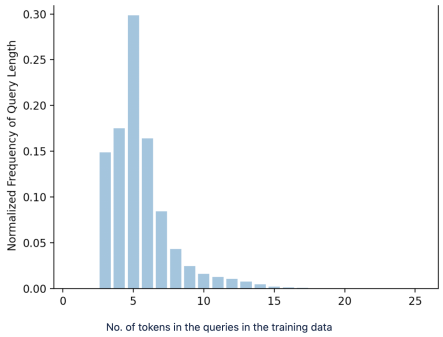

Figure 2: Query Length Distribution in Seq2Seq Training Data.

Query	Reformulated Query
tie-on face mask	face mask
glow up ihome speaker	glow ihome speaker
one shot roulette game	one shot roulette
starter knicks authentic jersey	starter knicks jersey
trek mountain bike	mountain bike
wrapping tapes	wrapping tape
camera sand bagsa	camera sand bag
rare tubeless kit	vintage tubeless kit
mall cookie cutters	mini cookie cutter set
hand card holder	arm card holder

Table 1: Examples of valid query transformations for training the strict term dropping model

model	N	d_{model}	d_{ff}	h	r	n_{param} (10^6)
tfMicro	2	64	64	2	n	13
tfMicroRel	2	64	64	2	y	13
tfMicroRelV2	2	64	64	8	y	13
tfTinyRel	2	128	256	2	y	27
tfTinyRelV2	2	128	256	8	y	27
tfTinyRelV3	2	512	1024	8	y	129
tfMini	4	256	512	4	n	65
tfMiniV2	4	256	512	8	n	65
tfMedi	8	512	1024	8	n	224.5
tfMediV2	8	512	1024	16	n	224.5

Table 2: Variations on the Transformer architecture

We generate two variations of training data to generate two model variants: the first one is a *strict term dropping* model where all query tokens of q' are restricted to be strict subsets of the query tokens in q. A few examples query transition pair (q, q') generated through this variant is shown in Table1 in rows 1-5. The second variant is a *term reformulation* (term dropping as well as term altering) model where all character tri-grams of q' are restricted to be strict subsets of the all character n(3)-grams of q. Other than the strict term dropping pairs that are generated in the super-set term reformulation variant, few examples of additional query pairs generated with the n-gram variant are also shown in Table1 in rows 6-10. From these examples the intuition of the second variant is clear: to be able to recover queries with notions other than just term dropping, such as handling singular plurals, spelling mistakes or white space mistakes, term altering or expansion with synonymous or more popularly used terms in user query logs. The query length distribution for the source query in the training data of term dropping (query, reformulation) pair is shown in Figure 2. Majority of the traffic is under query length 3-10 token bucket, with query length 5 having the highest traffic share.

4.2 Data Normalization and Encoding

We use sub-word/Byte Pair Encoding(BPE) encoding [25] to enhance the model vocabulary in a robust way. BPE brings the perfect balance between character and word-level hybrid representations which makes it capable of managing large corpora. The hybrid representation enables the encoding of any rare words in the vocabulary with appropriate sub-word tokens without introducing an *unknown* token. This especially applies to foreign languages like German where the presence of many compound words can make it hard to learn a rich vocabulary otherwise.

4.3 Seq2Seq Modeling

Neural Machine Translation (NMT) models have gained widespread popularity in the past decade due to their simplicity of underlying concept and generalizability specially for long word sequences. The Seq2Seq architecture [29][2] of NMT models has shown to be very effective in a variety of language modeling tasks under the next token or sentence prediction umbrella, such as machine translation, speech recognition, text summarization, and question answering [22]. A common theme of these problems is dealing with sequence of words/tokens in both input and output sides. Since user queries in e-commerce are collections of tokens where the ordering of the sequence has specific meaning from linguistic standpoint (as an example, *cars for sale* vs *for sale sign*), the reformulated queries are expected to follow the same behavior. In this context, we formulate our query reformulation problem as a Seq2Seq modeling task. A generic and bare-bone Seq2Seq model consists of two components: encoder and a decoder. In our problem scope of query reformulation in the Seq2Seq framework, the encoder finds a representation of the raw user query into a vector encoding the query context which is passed through a decoder that produces a corresponding query reformulation.

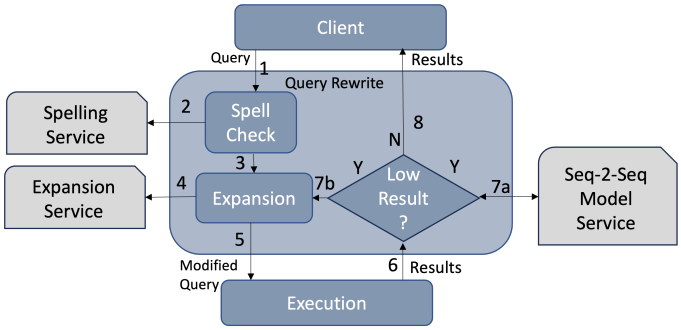

Figure 3: Query Processing Workflow Enhanced with Query Recovery

4.3.1 bi-LSTM Model. We use openNMT package in python [16] for experimentation of different Seq2Seq model architecture. For the first round of model implementation, we use a bi-directional encoder-decoder model with greedy search decoder (beam width 1). LSTM is chosen as the cell type in both encoder and decoder and the number of layers in both is 2. We use dropout layer with 0.3 keep probability in both encoder and decoder. Luong attention head [18] is used that allows the model to capture various components of the input sequence at every stage of the output sequence allowing the context to be preserved from beginning to end. An Adam optimizer [15] is used as an advanced Stochastic Gradient Descent (SGD) approach. Two different total number of units are used in our modeling experiments, namely 128 and 512.

4.3.2 Transformer Model. Proposed by Google in 2017 [32], transformer models have been widely used in different NLP tasks including NMT, mainly overcoming the shortage of missing information between the tokens far away from each other in a biLSTM model. A series of transformer models with different model architectures and parameters have been trained for the query reformulation task. The configurations of different models are shown in Table 2, in which N is the number of layers of the encoder and the decoder, d_{model} is the number of units in each layer of the encoder and the decoder, d_{ff} is the dimension of the inner layers of the feed-forward sub-network, h the number of heads, r is the relation-aware self-attention [26], and n_{param} is the number of trainable parameters in each model. The performance of each model listed in Table 2 will be discussed in Section 5.3.

4.4 RunTime

Figure 3 shows a query processing workflow that is integrated with a run-time Seq2Seq model service. The *Query Rewrite* module is responsible for detecting query intent and modifying the query with structural and lexical expansions to improve recall and precision. It consists of a pipeline of rewrite steps where each step handles a particular aspect of the query [31][20]. The description of the full pipeline is out of the scope for this paper, so only 3 steps are shown. In the spell corrected query is passed down to the expansion step where the query is rewritten to consider synonyms and translation. After all the rewrite steps, the modified query is sent to execution. Once the results come back a check is made and if the number of results is below a threshold, a call is made to the Seq2Seq model service to get the reformulated query. The new query then goes through some of these parts of the rewrite pipeline and gets executed again to get more results. Since the Seq2Seq model service call has a latency overhead we make sure to call the service only when needed for N&L instead of calling it at the beginning for all queries including non-N&L. This has an additional overhead of sending the model-reformulated query to the expansion service again, but that is less compared to the latency of the Seq2Seq model service.

5 EVALUATION

5.1 Model Variants & Performance

After the training data gets generated using the method detailed in Section 4.1, we split the data into training, validation, and test set with a 90%, 9%, and 1% split, respectively. In our experiment we evaluate two different model variations (in terms of modeling behavior: strict term drop and term reformulation models) and two different modeling configurations (in bi-LSTM architecture: models with 512 and 128 total units in the encoder and decoder) on the F1-score, precision, recall and loss at three levels: A. model's performance in predicting the correct query reformulation at the encoded token/subword level, B. model's performance in predicting the correct reformulation at query level; as an example, the query-level precision is calculated as among the queries where model performed a reformulation, the fraction of reformulations that were correct, and C. performance at query token level, where only tokens with sufficiently high frequency are considered from the model vocabulary (e.g., token-level precision is calculated as out of the high frequency tokens in the vocabulary that were included in model's prediction, how many were correctly kept or dropped).

model	$F1^v$	$Prec^v$	$Recall^v$	$F1^t$	$Prec^t$	$Recall^t$
TD-128_{us}	0.8275	0.8272	0.8467	0.7834	0.7852	0.8032
TD-512_{us}	**0.8583**	0.8527	0.8868	**0.7867**	0.7838	0.8197
TD-128_{uk}	0.8487	0.8503	0.8619	0.7897	0.7934	0.8044
TD-512_{uk}	**0.8865**	0.8845	0.9037	**0.7896**	0.7908	0.8134
TD-128_{de}	0.8693	0.8685	0.8843	0.7871	0.7893	0.8032
TD-512_{de}	**0.9116**	0.9110	0.9227	**0.7886**	0.7928	0.8049
TR-128_{us}	0.7929	0.7971	0.8107	0.7732	0.7786	0.7901
TR-128_{uk}	0.8139	0.8152	0.8299	0.7852	0.7875	0.8017
TR-128_{de}	0.8017	0.8046	0.8164	0.7708	0.7749	0.7853

Table 3: Query-level model metrics on decoded data. Legends: TD(Term Dropping), TR(Term Reformulation), v(validation set), t(test set)

Figure 4 shows cross-validation performance of different models described above at the subword/BPE encoded level on the validation set. We build separate models for each of the big three sites in US, UK, and DE/Germany because of the linguistic as well as user traffic (hence training data) differences in these sites. The strict term dropping model shows better performance across the board than the term reformulation model which is intuitive as the latter is a harder problem than dropping from a known set of query terms. Among the two different configurations in the model architecture side, the models with higher number of units (512) show better performances overall.

In Table 3 we show query-level F1-score, precision, and recall in both the validation and the test sets. For query-level behavior, we calculate precision as among the queries where model performed a reformulation, the fraction where reformulations were correct. Similarly, recall is among all the queries in the respective test set, what fraction had a correct reformulation from the model. In the term dropping models, we have 0.85-0.91 F1-score in validation and 0.78 in test set for different sites by the models with 512 total units, which is a better performance group than the models with 128 units. The term reformulation model shows around 0.8 F1-score for validation (> 0.77 for test) set. Table 4 shows model performance computed at query word-level. For query tokens with sufficiently high frequency in the vocabulary, we compute precision as the fraction of tokens that were predicted correctly (kept by the term dropping model) out of the tokens that were included in model's prediction. Recall is computed as the fraction of tokens that were predicted correctly out of the tokens in the vocabulary that made it through the frequency cutoff. Similar to the trend in the previous results, the term dropping models with 512 units show better performance by achieving F1-score ranging between 0.79-0.85 in validation set (0.73 in test set); term reformulation model shows around 0.75 F1-score in both sets.

We also ran an experiment to show the effect of considering more than the immediate SRP neighbors in the user sessions in the training data generation step. Following up from Section 4.1, for a user query q, if (q', q'') are the reformulated queries from q that ended up in a better user engagement over q in the same SRP burst

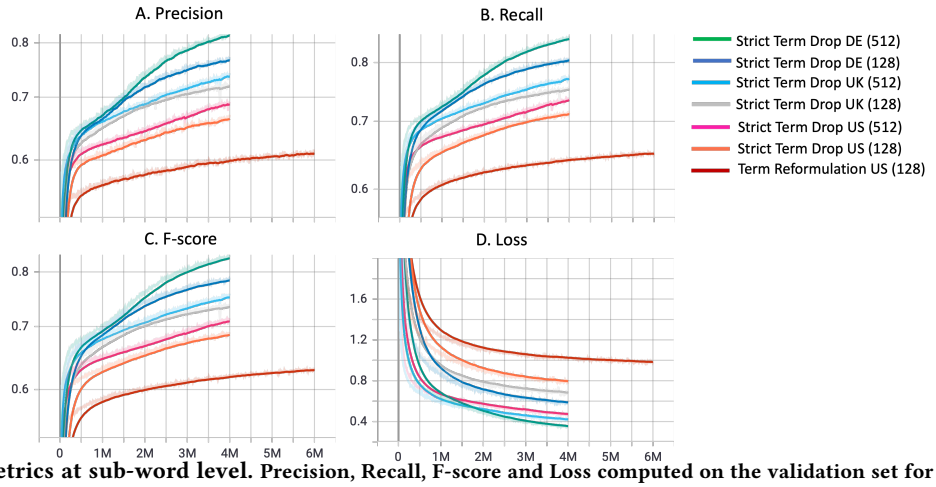

Figure 4: Model metrics at sub-word level. Precision, Recall, F-score and Loss computed on the validation set for the bi-LSTM Seq2Seq models on the encoded/subword data

model	$F1^v$	$Prec^v$	$Recall^v$	$F1^t$	$Prec^t$	$Recall^t$
TD-128_{us}	0.7664	0.7647	0.7681	0.7178	0.7157	0.7201
TD-512_{us}	**0.7869**	0.7868	0.7870	**0.7321**	0.7319	0.7323
TD-128_{uk}	0.7945	0.7929	0.7962	0.7162	0.7139	0.7185
TD-512_{uk}	**0.8107**	0.8104	0.8110	**0.7302**	0.7296	0.7309
TD-128_{de}	0.8079	0.8071	0.8087	0.7226	0.7215	0.7238
TD-512_{de}	**0.8484**	0.8482	0.8485	**0.7268**	0.7265	0.7271
TR-128_{us}	0.7379	0.7435	0.7330	0.7179	0.7256	0.7111
TR-128_{uk}	0.7526	0.7569	0.7486	0.7166	0.7204	0.7132
TR-128_{de}	0.7571	0.7642	0.7509	0.7575	0.7304	0.7233

Table 4: Token-level model metrics on decoded data. Legends: TD(Term Dropping), TR(Term Reformulation), v(validation set), t(test set)

model	$F1^q_{1hop}$	$F1^q_{2hop}$	$\%delta^q$
TD-512^v_{us}	0.8527	0.8621	+1.1
TD-512^t_{us}	0.8010	0.8633	+7.8
TD-512^v_{uk}	0.8680	0.8876	+2.3
TD-512^t_{uk}	0.8023	0.8867	+10.5
TD-512^v_{de}	0.9040	0.8993	-0.5
TD-512^t_{de}	0.7938	0.8995	+13.3

Table 5: Model improvement by enhancing training set from 1-hop to 2-hops. Legends: TD(Term Dropping), v(validation set), t(test set), q(F1 computation at decoded query-level)

model	$F1^{tok}_{1hop}$	$F1^{tok}_{2hop}$	$\%delta^{tok}$
TD-512^v_{us}	0.7868	0.8048	+2.3
TD-512^t_{us}	0.7320	0.8317	+13.6
TD-512^v_{uk}	0.8107	0.8430	+3.9
TD-512^t_{uk}	0.7302	0.8660	+18.6
TD-512^v_{de}	0.8483	0.8440	-0.5
TD-512^t_{de}	0.7268	0.8766	+20.6

Table 6: Model improvement by enhancing training set from 1-hop to 2-hops. Legends: TD(Term Dropping), v(validation set), t(test set), tok(F1 computation at decoded token-level)

immediately after q (i.e., q, q', q'' are in a sequence), then we consider (q, q') as a valid candidate reformulation in the 1-hop model and both (q, q') and (q, q'') as a valid candidate reformulation in the 2-hop model. We get upto 13% improvement in the F1-score in the query-level metric (Table 5), and upto 20% improvement in the token level (Table 6) when the query transitions to consider in the training process is upgraded from 1 hop to 2 hops. Note that as we further extend the number of hops in the training data generation, the reformulations tend to get more tokens dropped from the original query, which eventually leads to loosing more query context.

5.2 Model Behavior Study

We bucket the different types of reformulations we see in model prediction and quantify the behavior for different models. In this way, we make sure the methodology we follow from training data generation to modeling phase (i.e., approaches for generating training data for term dropping vs term reformulation model) are persisted in model predictions. In Table 7 different prediction behavior and examples are shown, and Table 8 shows the percentages for each behavior for three different bi-LSTM models. Most noticeable point here is that the *Subset* bucket is highest in the term dropping model with 512 units (TD-512_us), the variant that also shows better performance than the other models in Section 5.1. Also, since the *Subset* bucket is imitating the term dropping behavior, it is intuitive that the term dropping models in general has much higher percentage in this bucket. The term reformulation model has *Same Length With Modification/SameLWM* as the second highest bucket, which also is interesting as it is following a general nature of how users reformulate their queries.

5.3 Transformer Model Performance

We evaluate different transformers models listed in Table 2 and show the results in Table 9. The query-level recall would be 1 if the

Behavior	Query	Reformulation
Subset	usps priority tape	usps tape
Superset	gold iphone	gold phone iphone
SameLWM	wrapping tapes	wrapping tape
SmallerLWM	engrave photo wood	engraved photo
LongerLWM	it cosmeticslash blowout	it cosmetics slash blowout
Spacing	victorian house photoframe	victorian house photo frame
SubstringDrop	hondalicense plate frame	honda plate frame

Table 7: Examples of Model Behavior. Legends: TD(Term Dropping), LWM(Length With Modification)

Behavior	TD-128_{us}	TD-512_{us}	TR-128_{us}
Subset	96.54	99.51	81.28
Superset	0.00	0.00	0.00
Same Query	0.00	0.00	0.00
SameLWM	0.06	0.09	8.81
SmallerLWM	0.00	0.00	0.09
LongerLWM	0.01	0.00	0.20
Spacing	0.00	0.00	2.17
SubstringDrop	0.04	0.03	3.46
Empty	0.28	0.01	0.33
Others	3.07	0.37	3.86

Table 8: Model behavior statistics (%). Legends: TD(Term Dropping), TR(Term Reformulation), LWM(Length With Modification)

model always outputs long queries that contain all the tokens from q_i. Thus, recall by itself is not a reasonable performance metric when dealing with reformulation where we want to extract the key tokens from an originally long query. Therefore, in addition to recall, we also check whether the model's outputs fall into the subset bucket defined in Section **??** to ensure that a model extracts useful information from the original queries with high recall. We can see from Table 9 that the ratio of reformulated output queries in the subset bucket increases as the transformer model gets bigger in model parameter size. Furthermore, we observe much less percentages of same or longer output query reformulation in deeper transformer models.

Model	Prec	Recall	F1	Subset
tfMicro	75.6	84.0	77.8	97.7
tfMicroRel	75.7	84.1	77.9	98.6
tfMicroRelV2	76.1	83.4	77.8	99.0
tfTinyRel	76.6	82.8	77.8	99.7
tfTinyRelV2	76.5	83.5	78.2	99.4
tfTinyRelV3	76.9	80.5	77.0	99.8
tfMini	76.8	81.2	77.2	99.6
tfMiniV2	76.6	82	77.6	99.7
tfMedi	77.1	**79.9**	76.8	**99.9**
tfMediV2	**77.2**	79.3	76.5	99.8

Table 9: Offline Evaluation on Transformer Models. Details of the each model is shown in Table 2

	$Delta_{latency}$
50th percentile	-2ms
95th percentile	+14ms
99th percentile	+34ms

Table 10: Latency Changes from biLSTM to Transformer models in milliseconds (ms)

In order to understand how the model behaves on different query lengths, we also compare in Table 11 the four metrics (prf, subset ratio) on different query length buckets. We observe that for queries with less than 12 tokens, the transformer model has a better performance in terms of precision, recall and subset ratio. However, since very long queries (more than 12 tokens) are rare (only 5%) in the evaluation set, the performance difference is negligible.

For search, besides precision and recall, the latency of the query also impacts the user experience since users might abandon the search session if the query processing takes too long. The difference in the end-to-end inference between the biLSTM and the transformer model is shown in Table10. The transformer model, although showing better performance in different model evaluations, does show a degradation over the biLSTM model due to the nature of heavier model architecture and multiple attention heads. One of the future tasks is to optimize the model latency for this model with GPU cluster optimization as well model parameter tuning.

Length	transformer				biLSTM			
	Prec	Rec	F1	Subset	Prec	Rec	F1	Subset
1-3	0.83	0.83	0.83	1.00	0.82	0.82	0.82	0.99
4-6	0.80	0.80	0.79	1.00	0.79	0.80	0.79	1.00
7-12	0.76	0.76	0.73	1.00	0.76	0.76	0.72	0.99
13-25	0.74	0.83	0.73	0.93	0.77	0.79	0.73	0.97

Table 11: Comparison on different query length buckets between biLSTM and transformer models.

5.4 Baseline Comparison of Seq2Seq model on Query-Item Relevance

In this section we measure how much divergence was introduced to the original query intent by query reformulations predicted by the Seq2Seq models compared to a **baseline model**. Since for N&L queries the approach of matching every query term to the item titles will result in insufficient results, one possible baseline model is to use an empirically learned fraction x and for every query retrieve an item that matches *any x%* of the of query terms to the item title terms. For example, for the query *3 foot tall doll*, let's say that the learned fraction x is 0.6. Then the baseline model will retrieve an item if it matches with any 2 of the query tokens, which could be token pairs such as (3, tall), (tall, doll), and (3, foot). To evaluate the impact of relevance of retrieved items by different methods, we construct two Null & Low query sets in US site: a set of 3500 queries where all queries have 6 or more tokens, (referred as Long Query Null Low or LongQ_NL for short later in the paper), and a set of 8000 queries of all lengths (referred as N_L in short). For a query q in

these sets, we run any variant of the Seq2Seq model and get the predicted reformulation r. Then we construct two initial sets of items: set A (baseline), a random sample of 500 items from the top 10,000 items returned by for q by the baseline model, where the items are sorted by the default best match ranker [5], and set B (Seq2Seq), random sample of 500 items from the top 10,000 items returned for the reformulation r by doing all query token matching to item titles. Using these two sets, we compare the average relevance score of items for each query, and also compute average relevance across queries in the two query sets mentioned above. For comparing the relevance, we leverage three GBM models that are trained on human judged query-item relevance data (referred as Rel_nonNL, Rel_All, Rel_NL for three versions of the models that are trained on non N&L queries, both non-N&L and N&L queries, and N&L queries only, respectively), and eBERT (BERT based model [4] with additional item and query data from our company's systems).

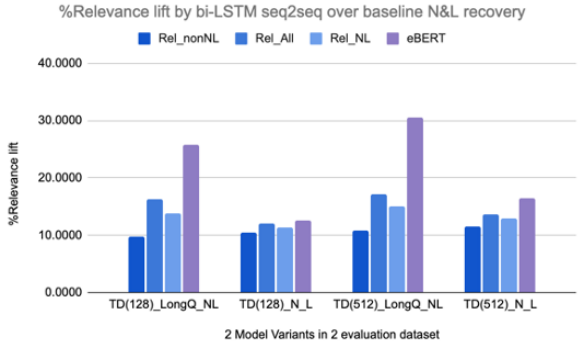

Figure 5: Query-item relevance evaluation of bi-LSTM Seq2Seq model. Legends: TD(Term Dropping), LongQ(Long Queries), NL(Null & Low)

Figure 5 shows the result of relevance evaluation with the method described above for the biLSTM model. The y-axis is percentage of improvement in relevance of retrieved items to the original query, when the seq2seq reformulation is used to perform the retrival (referred as *%Relevance lift* in Figure 5). We show percentage of improvement we see compared to the baseline model using two N&L datasets, two different biLSTM model configurations, and four different relevance models. Overall, with the biLSTM term dropping model with 512 units on the LongQ_NL we achieve between 10-30% improvement over baseline (11-16% on the N_L data set), which is a significant relevance improvement over baseline. We also compare recall and relevance between transformer and biLSTM model and observe higher recall (delta +4, +45, +551 at 25th, 50th, and 75th percentile) with neutral relevance impact for the transformer model, which is promising and desired from the model architecture upgrade.

Figure 6 in Section 7 shows an anecdotal example on a long N&L query *25 used tennis balls grade a free fast ship support our non profit mission.* The recovery by the baseline matches any item title with x% query term match, and retrieves a lot of irrelevant items shown in the left panel that are not tennis balls. The biLSTM model reformulates the query as *tennis balls*, and in the middle panel all the top retrieved items are tennis balls. The transformer model reformulates the query as *used tennis balls*, and as shown in the right panel all top items are *used* tennis ball items.

5.5 Human Judgement with Baseline Comparison & A/B test

We conducted an offline human judgement task on set of 500 N&L queries where the human annotators were provided with a data set quadruples, each quadruple consisting of a query q, its reformulation r by the Seq2Seq biLSTM model, and two e-Commerce SRP page urls u_1 and u_2 using q and r used as search keywords, respectively. Note that in u_1, the baseline model described in Section 5.4 is used to retrieve the SRP items as the query is N&L. The task for the annotators was to label each quadruple into one of three classes: class A (u_1 better than u_2), class B (u_2 better than u_1), class C (neutral). 79% of the quadruples in the judgement set got a B or C vote, with class B getting 45% vote, leading to the conclusion that the SRP experience where the biLSTM reformulation was used as the search keyword for item retrieval results in a significantly better experience compared to the baseline.

We conducted an online A/B test with live traffic in eBay search engine, exposing the new experience to millions of users that fall into a N&L experience. The users in the test variant with a N&L query were exposed to SRP using the query reformulation by the biLSTM model as the search keyword, and those in the control variant were exposed to the baseline model performing the item retrieval in the SRP. We ran this experiment for a period of 2 weeks in the three big markets (US, UK and Germany). We measured both financial and engagement metrics in the two variants. For the treated traffic, we observed statistically significant lift in search conversion and sale metrics, drop in search abandonment rate, and decrease in search re-query rate. The results demonstrated clearly that the proposed biLSTM Seq2Seq model provides a much better N&L recovery and as a result an elevated user experience.

5.6 Shortcoming of the existing model

Both the bi-LSTM and transformer models that were A/B tested in live user traffic are term dropping models that finds an optimal reformulated query by dropping one or more tokens of the original query. Such models does not perform well for short queries where the number of query tokens are limited (typically between 2-3), where term replace or augmentation mechanisms are needed for query reformulation. In this paper we showed some preliminary experiments with term reformulation models which were underperforming compared to the term drop models (Table 3, Table 4). This is also clearly observed in Figure 5 where the %relevance lift in the long N&L query set is much better in the dataset specific to long N&L queries (*LongQ* in Figure 5) compared to others. One of the next improvement steps here is to develop a dedicated model for short N&L queries that has term reformulation capabilities with the support of inherent relations among different entities in the query itself from a knowledge graph.

6 DISCUSSION AND FUTURE WORK

The query reformulation models developed to improve the e-commerce search experience for N&L queries have shown great promise and

opens up new possibilities for the NMT framework based query reformulation models to be applicable to many other search use cases in the e-commerce domain. The proposed model can be greatly benefited from many core structural properties of e-commerce query and items that can be encoded into a knowledge graph and incorporated into the training data or the encoding layer of the model architecture itself. Named entity recognition (NER) is also another dimension worth exploring for the user query reformulation problem where NER can be used to identify the item's structural entities in a query keyword, which will eventually help in deciding whether to keep or drop a query token. When gathering data, we filtered out not-safe-to-work data. However, another hurdle we faced in this endeavor was mitigating the risk of generating toxic output from the model. These are some of the future challenges for the proposed work.

REFERENCES

[1] Ricardo Baeza-Yates and Alessandro Tiberi. 2007. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 76–85.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web*. ACM, 325–332.

[4] eBay Inc. 2020. PyKrylov: Accelerating Machine Learning Research at eBay. https://tech.ebayinc.com/engineering/pykrylov-accelerating-machine-learning-research-at-ebay/.

[5] ebay Inc. 2022. *Optimizing your listings for Best Match.* //www.ebay.com/help/selling/listings/listing-tips/optimising-listings-best-match?id=4166#section1

[6] Sreenivas Gollapudi, Samuel Ieong, and Anitha Kannan. 2012. Structured Query Reformulations in Commerce Search *(CIKM '12)*.

[7] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. 2013. Beyond clicks: query reformulation as a predictor of search satisfaction. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2019–2028.

[8] Ahmed Hassan, Ryen W White, Susan T Dumais, and Yi-Min Wang. 2014. Struggling or exploring? Disambiguating long search sessions. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 53–62.

[9] Ahmed Hassan Awadallah, Ranjitha Gurunath Kulkarni, Umut Ozertem, and Rosie Jones. 2015. Characterizing and predicting voice query reformulation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 543–552.

[10] Sharon Hirsch, Ido Guy, Alexander Nus, Arnon Dagan, and Oren Kurland. 2020. *Query Reformulation in E-Commerce Search.* 1319–1328.

[11] Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. 2003. Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology* 54, 7 (2003), 638–649.

[12] Jeff Huang and Efthimis N Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 77–86.

[13] Bernard J Jansen, Danielle L Booth, and Amanda Spink. 2009. Patterns of query reformulation during web searching. *Journal of the american society for information science and technology* 60, 7 (2009), 1358–1371.

[14] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*. ACM, 387–396.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810* (2017).

[17] Sen Li, Fuyu Lv, Taiwei Jin, Guiyang Li, Yukun Zheng, Tao Zhuang, Qingwen Liu, Xiaoyi Zeng, James Kwok, and Qianli Ma. 2022. Query Rewriting in TaoBao Search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3262–3271.

[18] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[19] Saurav Manchanda, Mohit Sharma, and George Karypis. 2019. Intent Term Weighting in E-Commerce Queries. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*.

[20] Aritra Mandal, Ishita K. Khan, and Prathyusha Senthil Kumar. 2019. Query Rewriting using Automatic Synonym Extraction for E-commerce Search. In *eCOM@SIGIR*.

[21] Roberto Navigli and Paola Velardi. 2003. An analysis of ontology-based query expansion strategies. In *Proceedings of the 14th European Conference on Machine Learning, Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia*. Citeseer, 42–49.

[22] Graham Neubig. 2017. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619* (2017).

[23] Daan Odijk, Ryen W White, Ahmed Hassan Awadallah, and Susan T Dumais. 2015. Struggling and success in web search. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1551–1560.

[24] Stefan Riezler and Yi Liu. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics* 36, 3 (2010), 569–582.

[25] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* (2015).

[26] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155* (2018).

[27] Gyanit Singh, Nish Parikh, and Neel Sundaresan. 2012. Rewriting null e-commerce queries to recommend products. In *Proceedings of the 21st International Conference on World Wide Web*. 73–82.

[28] Gyanit Singh, Nish Parikh, and Neel Sundaresn. 2011. User behavior in zero-recall ecommerce queries. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 75–84.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[30] Zehong Tan, Canran Xu, Mengjie Jiang, Hua Yang, and Xiaoyuan Wu. 2017. Query rewrite for null and low search results in eCommerce. In *eCOM@ SIGIR*.

[31] Andrew Trotman, Jon Degenhardt, and Surya Kallumadi. 2017. The Architecture of eBay Search. In *eCOM@SIGIR*.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[33] Yaxuan Wang, Hanqing Lu, Yunwen Xu, Rahul Goutam, Yiwei Song, and Bing Yin. 2021. QUEEN: Neural query rewriting in e-commerce. In *The Web Conference 2021*. https://www.amazon.science/publications/queen-neural-query-rewriting-in-e-commerce

[34] Jinxi Xu and W. Bruce Croft. 1996. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Zurich, Switzerland) *(SIGIR '96)*. ACM, New York, NY, USA, 4–11. https://doi.org/10.1145/243199.243202

7 APPENDIX

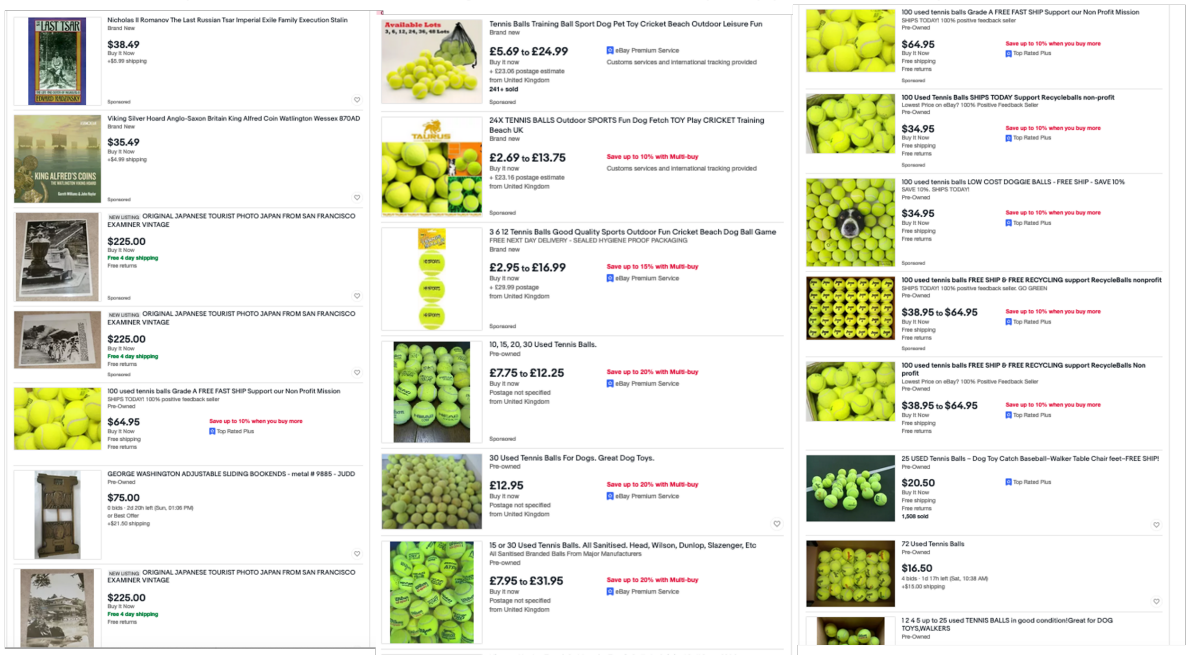

Figure 6: Side-by-side example of Null & Low recovery by legacy baseline (left), middle (biLSTM Seq2Seq), right (transformer Seq2Seq)