Intent-Driven Similarity in E-Commerce Listings

Gilad Fuchs, Yoni Acriche, Idan Hasson, Pavel Petrov
{gfuchs,yacriche,ihasson,ppetrov}@ebay.com
eBay Research

ABSTRACT

Discovering similarities between online listings is a common back-end task being used across different downstream experiences in eBay. Our baseline unstructured listing similarity method relies on measuring the semantic textual similarity between the embedding vectors of listing titles. However, we discovered that even with the latest contextualized embedding methods, our similarity fails to give the proper weight to the key tokens in the title that matter. This often results in identifying listing similarities that are not sufficient, which later hurts the downstream experiences. In this paper we present a method we call "Listing2Query", or "L2Q", which uses a Sequence Labeling approach to learn token importance from our users' search queries and on-site behaviour. We used pairs of listing titles and their matching search queries, and leveraged a contextualized character language model, to train L2Q as a bidirectional recurrent neural network to produce token importance weights. We demonstrate that plugging these weights into relatively straightforward listing similarity methods is a simple way to significantly improve the similarity results, even to the extent that it consistently outperforms those created by popular representations such as BERT. Notably, this approach is not reserved to only large online marketplaces but can be generalized to other cases that include a search-driven experience and a recall set of short documents.

CCS CONCEPTS

• **Applied computing → Electronic commerce**.

KEYWORDS

Machine Learning, Sentence Similarity, E-commerce

ACM Reference Format:
Gilad Fuchs, Yoni Acriche, Idan Hasson, Pavel Petrov. 2020. Intent-Driven Similarity in E-Commerce Listings. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412715

1 INTRODUCTION

eBay is an online marketplace that connects between buyers and sellers, and provides different tools to help its users to compare and discover similarities between listings. This is used, for example, to recommend prices to sellers by comparing prices of similar listings, to identify miscategorized listings or to recommend similar listings to buyers. In its strictest definition, the similarity between listings can be viewed through the lens of internationally recognized product identifiers, such as Global Trade Item Number (GTIN) or Universal Product Code (UPC). However, relying solely on using product codes for identifying similar listing suffers from two main drawbacks. First, it assumes that all sellers provide these codes, which, empirically, is not the case. Second, not every product has a product identifier. This is especially true for "long-tail" products, like collectables, or hand-made products. On the flip side, when identifying similar listings not by product identifiers one needs to accept that the similarity will be stochastic and undefined, and that its quality will ultimately be subjective to the opinion of its users. This also poses a challenge for success-measurement, a topic that we address in the following sections. In our case, identifying non-identical, yet close, listings as similar is acceptable, if not required, for most of our downstream tasks. For example, a similar listing recommendation engine would be considered bland if all of its recommendations were to be identical to the one that the buyer is currently viewing.

As for eBay's listing entity, each contains information about a product that the seller is putting up to sale, together with its commercial terms. Most notably this would include the listings title, description, condition, price, shipping cost, product pictures and attribute values (brand, size, GTIN, etc.). Since the listing title often contains concise and relevant information about the listing our current discussion is focused solely on the task of measuring similarity by comparing between different listing titles.

Generally, the task of measuring similarity between listings is not new to eBay. Historically, listing similarity was done via classic Information Retrieval techniques, such as BM25 [30] or Jaccard [15]. A major step forward was to use token representations to create title embeddings, which allow to measure the distance between title embedding vectors. In this respect, the first method was to represent the title vector as the average, or weighted average, of the token embedding vectors that each title consists of [17, 24]. Later, more advanced contextual representations such as BERT [10] allowed to also preserve the context between the tokens of the title, and indeed showed an improvement in most of our similarity metrics. However, to this point, these title embedding methods highlighted a consistent error pattern. Usually, we would see a pair of listings that were considered similar, and that indeed share almost the same set of tokens, except for just one or two. Alas, these were usually the most important tokens to define the product, like, for example, the precise model name of the product (e.g replace iPhone 8 with iPhone 11). Moreover, we saw that the addition of negligible phrases to the title, like the promotional "new listing!" or "choose your size", which sellers often use, has an disproportional defocusing effect on the embedding and the final listing similarity

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412715

outcome. To solve this we needed to find a method for identifying the key tokens in every title and to ingest this as input into the listing similarity measurement.

The goal of this paper is to describe a method for learning the importance of title-tokens from on-site user behavioural data, in order to ultimately improve our listing similarity. Our key hypothesis is that in most cases buyers already search for what they think is most important in the title of the product that interests them. Mapping between what users search and the listing-click or purchase actions that follow allows to learn what are the most important tokens in listing titles. Once these patterns are learned using a supervised model (which we call "Listing2Query", or, in short, "L2Q"), we plug its predictions as weights in a weighted average of eBay-specific word embeddings to create improved title embeddings. Apart from the immediate boost in similarity results in every empirical evaluation that we conducted, we think that the main strength of the method lies in the simple manner in which the L2Q model results can be plugged directly into existing large scale systems, without requiring significant engineering overhead.

An overview of the paper: In the next section we describe related work in the field of textual similarity and sentence embedding. In the section following it we present a methodology for mapping between listings to queries to create the L2Q training data and describe the model we trained. This is followed by the results of a qualitative and quantitative evaluation, in which we show how the L2Q model can improve the performance of two listing similarity tasks in eBay's backend. Next, we describe the results from an A/B test that we conducted to a production deployment of L2Q. We conclude by discussing the tradeoffs between different L2Q model flavours and potential improvements.

2 RELATED WORK

As language modeling is a fast growing field of research in NLP, most recent work in textual similarity focuses on techniques for word-sequence vector representations, and the use of those in a variety of downstream similarity tasks. For single-word unsupervised embedding, the most notable works include Word2Vec [24], GloVe [27] and FastText [6]. As for multi-word embedding (e.g. sentence, paragraph and document embedding), early work suggested different techniques to compute those as a function of the vectors of the words that comprise the sentence or paragraph [4]. More recent work has focused on generating such embedding directly by an unsupervised manner, like with Skip-Thoughts [19], Quick-Thoughts [20] and sent2vec, [26] or by a supervised manner, like with InferSent [9]. More recent work in that field focuses on generating a contextualized embedding, like with ELMo [28], Flair [2] and, perhaps most notably, BERT [10].

Regarding related work in the e-commerce space, the structure of listing titles is mostly examined through the lens of product matching (a sub-task of Entity Matching), by extracting specific attribute-value pairs and using them as input for a matching function [11, 22, 23, 29]. In contrast, Shah et al. [31] uses a classifier for the product matching task. Others have used various neural-networks based representations of products to improve query to product matching or to create personalized product recommendations [1, 33, 35].

3 METHODS

In the following section we describe how we created the L2Q model. This is followed by a description of the fine-tuning steps that we took for the BERT model that was used to benchmark L2Q.

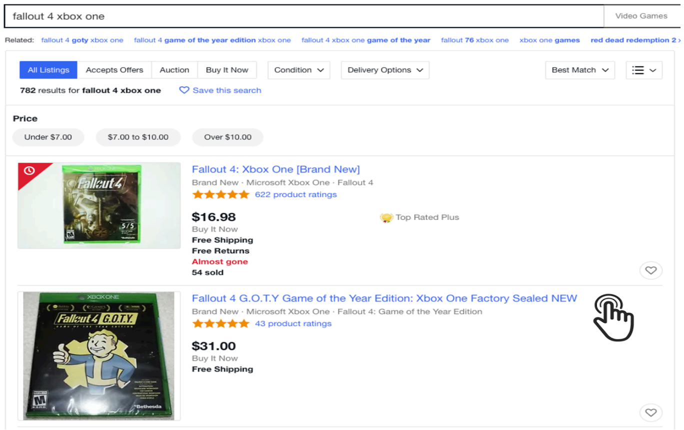
3.1 L2Q model training

The training dataset for the L2Q model consisted of 8 million random pairs of titles, and their matching search queries from the eBay US website. As demonstrated in Figure 1, each of these entities represents a purchase-intent pattern that starts from a user search in eBay's search engine, and ends with a click on a certain listing that got returned in the search results. A more conservative settings for this coupling is to focus only on sessions that end with the user purchasing the listing. While such setting can improve the accuracy of the coupling (by removing sessions where users accidentally clicked on a listing, for example), it may also hurt the coverage of the model, as some listings will often get clicked on, but hardly ever purchased.

The dataset was pre-processed for the L2Q model training by transforming the tokens to lowercase and removing known stopwords and non-alphanumeric characters. We then split it by eBay's vertical classification (e.g. “Fashion”, “Electronics”, “Home & Garden”). In order to train a sequence labeling model, we labeled each of the tokens in the title as a binary indicator to mark whether it was used in the search query that is mapped to that instance. For example, in the case where the search query “iphone x unlocked” led the user to click on the title “Apple iPhone X unlocked 64GB black”, then the title will be labeled as follows - {“Apple”: 0, “iPhone”: 1, “X”:1, “unlocked”: 1, “64GB”: 0, “black”: 0}. The median number of keywords per query is 3, and the median number of tokens in the titles is 12. On average 25% of the tokens in each title were tagged as positive. Note that since different user-queries can lead to the same listing, the training data often consists of multiple different label sets for the same listing title. This is advantageous since it allows the model to learn the distribution of queries per listing, and therefore to set less restrictive classification boundaries and identify the inner hierarchy within the tokens in the title.

For modeling we use the contextual string embeddings for sequence labeling approach via the Flair framework [2], which achieved state-of-the-art results in multiple sequence labeling tasks, such as Named Entity Recognition (NER) and part-of-speech (PoS) tagging. This framework allows to stack multiple word embeddings from various pre-trained models, which are then passed into a vanilla bidirectional Long short-term memory (BiLSTM) recurrent neural network and a subsequent conditional random field (CRF) decoding layer [2, 12, 21].

In addition to the common pre-trained embeddings, Flair provides its own pre-trained embedding which has the advantage of generating context-dependent embedding per token (e.g. the token “new” will have different embedding in the context of “new nike shoes” vs. “new balance shoes”). Flair embeddings were generated by training a bidirectional language model (LM), consisting of a character-based BiLSTM with 2048 hidden states, aiming to predict the continuations of each input sentence [3]. Training was done using SGD with a batch size of 100, clipping gradients at 0.25 and dropout probabilities of 0.25 on the 1-billion word corpus [7]. For



782 results for fallout 4 xbox one

Fallout 4: Xbox One [Brand New]
Brand New · Microsoft Xbox One · Fallout 4
★★★★★ 622 product ratings
$16.98
Buy It Now
Free Shipping
Free Returns
Almost gone
54 sold

Fallout 4 G.O.T.Y Game of the Year Edition: Xbox One Factory Sealed NEW
Brand New · Microsoft Xbox One · Fallout 4: Game of the Year Edition
★★★★★ 43 product ratings
$31.00
Buy It Now
Free Shipping

Figure 1: Example of the coupling process between queries and listing titles, which is later used in the L2Q model training. In this example, a the user searched for "fallout 4 xbox one", got the search results and then clicked on the listing with the title "Fallout 4 G.O.T.Y Game Of the Year Edition: Xbox One Factory Sealed NEW". The title-token set that overlaps with the query-tokens is tagged with positive labels. The rest of the title-tokens are tagged with negative labels.

contextual word embeddings the hidden state after the last character of a specific word was extracted from the forward LM and concatenated to the word's first character from the backward LM. This allows to generate embeddings which holds the information of both the word itself and the context from both sides [3].

To train the L2Q model, we use a combination of the character-based Flair embedding (forward and backward) stacked with GloVe embedding [27] following BiLSTM-CRF with 256 hidden states as suggested and pre-implemented by the Flair framework [2]. Note that the character-level embedding can be advantageous in our case given the sellers' unique and highly diverse jargon. Although the use of a CRF layer increases the training time of the model, our model validation results have shown that it improves the model's performance. Training was done for a maximum of 10 epochs with a batch size of 32, and a learning rate of 0.1 and with SGD optimizer on a single GPU. The best model was selected based on the validation set F1 score.

For the token-level embeddings we trained a Word2Vec model using approximately 5 million eBay US titles, which were pre-processed using the same steps as in the L2Q model, and with a minimal token count of 5. The Word2Vec model was trained to produce vectors with 100 dimensions. The final L2Q title-level embeddings ('W2V+L2Q') were created by computing a weighted average of the Word2Vec token-level vectors that each title consists of, using the predictions produced by the L2Q model as weights. For example, in the case of the title "Apple iPhone X Black Case Genuine Original", the final title embedding will be most affected by the vectors of "iPhone", "X" and "Case" as their model prediction scores are the highest in the title

In all cases, the nearest neighbors were found using L2 distance (using the faiss package [16]). Of note, using the cosine similarity instead of L2 provided similar results.

3.2 BERT embeddings and fine-tuning

In this paper we used BERT, alongside with other methods, to benchmark the L2Q-driven embedding listing similarity results. For this we used the pre-trained bert-base-uncased model which includes 12 encoder layers, 12 attention heads per layer and 768 hidden units. For fine-tuning we used a total of 50,000 title pairs to create a classification task in which non-identical listing titles of the same product were marked as positive (product identity was derived using exact product identifier codes). On the flip side, we marked listing examples that are in the same category, yet belong to a different product, as negative. Fine-tuning was done for a maximum of 5 epochs with a batch size of 32, learning rate of 2e-5, maximum sequence length of 128, with Adam optimizer [18] on a single GPU. We used the bert-as-service package [34] to generate titles embeddings for both BERT default and the fine-tuned version of it. Titles embeddings were generated by average pooling of all the tokens representation in the second-to-last hidden layer.

4 RESULTS AND APPLICATIONS

In the following section we present the L2Q model results as measured directly on the sequence labeling task. We then show how the model can be applied to improve the performance of two listing similarity tasks that are driven by Word2Vec and the Jaccard distance, and present the results from our production deployment.

4.1 L2Q Results for Sequence Labeling

As L2Q is trained as a sequence labeling model, the output prediction set consists of the estimated likelihood that each of the title's tokens will appear for the matching search query. Table 1 shows the model evaluation results, as measured using precision, recall and F1 score on a test data, while using the common threshold of 0.5 for score discretization. In order to avoid the noisy effect of outliers, we focus our analysis on titles that only have 3 or more matching queries. To demonstrate the necessity of the CRF layer, we divide our model evaluation results to those with and without this layer. As seen, the use of a CRF layer almost always increases the model's F1 score, and overall by an average of 2 points (similar results were observed in the validation data). In table 2, we show three prediction examples of the model, and compare each token's expected likelihood to appear in the query to the true proportion that was observed on the site. For reference, we also show the top 5 queries that have historically led users to each of the test listings, alongside with their respective proportions.

4.2 L2Q for Weighted Title Embeddings

The main task that we tested L2Q for is the listing similarity task. To measure the impact of the model we first perform a qualitative analysis to compare the results between the different benchmark models. For this, we compared the nearest neighbors results from our W2V+L2Q embeddings to (1) those created by using a simple Word2Vec token average ('W2V'), (2) a Word2Vec token average that is weighted by TF-IDF ('W2V+TF-IDF'), (3) a default BERT model and (4) a BERT model that was fine-tuned using eBay's title corpus. As seen in Table 3, while the nearest neighbors retrieved in example 3a by W2V, W2V+TF-IDF and BERT were indeed of the right gaming console ("Xbox One") they show a different video game than that of the test listing ("Fallout 4"). Similarly, the fine-tuned BERT model retrieved the right video game but the wrong gaming console ("PlayStation 4" instead of "Xbox One"). In contrast, W2V+L2Q retrieved a title which includes both the right video game and gaming console. Note that all the methods except for W2V+L2Q retrieved titles which include the phrase "Brand New Factory Sealed". We estimate that this phrase contributed to the lower quality of the retrieved similar titles. Examples 3b and 3c further demonstrate L2Q+W2V's superiority in similar cases.

Conducting a quantitative comparison between the listing similarity results of different methods requires to decide on an evaluation methodology. This is not straightforward in cases like this where there is no ground truth as to which listings are indeed considered similar. We decided to measure three main metrics- Same Brand Percentage, Same GTIN Percentage and Same ISBN Percentage. The first metric measures whether the pair of items that were found as similar share the same brand attribute (brand names are usually provided by the sellers). The idea behind this is that items that share the same brand name are more likely to be similar to one another. This is clearly not a perfect way to measure similarity since two relatively similar products can have a different brand name, and two very disjointed products can share the same brand name. The two latter metrics, driven by GTIN and ISBN (International Standard Book Number), measure the percentage of listing pairs that have an identical product identifier. Although much stricter than Same Brand Percentage, this metric is also not ideal for our use cases since two very similar products can have a different product identifier. Although these disadvantages, when combined together, these metrics do give a sense of how the similarity methods benchmark against each other. As can be seen in table 4, we found that W2V+L2Q outperforms all other methods. This is aligned with the results observed in the qualitative evaluation.

Note that during our experimentation phase we often attempted to come up with a more basic token ranking approach, which, potentially, can be easier to compute. For example, one attempt included weighting the title tokens using their respective frequency in user queries. We saw that this, together with other similar variants, yielded significantly worst results than L2Q (not shown here for the sake of abbreviation). The error analysis showed that simply using search-token frequency creates a bias towards tokens that appear in many different types of titles (e.g. "black" or "game"), and therefore, it failed to give the proper weight to unique key tokens in every product (e.g. "Fallout"). The seemingly obvious fix to this would be weight every title token by the exact search queries that have led to that specific listing. This is not a good solution since it limits the scope of the similarity search to only items that have large historical user traffic.

Last, for an evaluation that is more in-line with how these similarities would be used in different downstream tasks, we conducted a blind test that involved human agents. To measure the exact delta from using L2Q weights we compared the results from W2V+L2Q to those from the regular W2V. The agents were presented with a total of 1000 random results from W2V and W2V+L2Q, and were asked to rate their relevance compared to the title of some seed listing. The rating options were: "Extremely Similar", "Very Similar", "Somewhat Similar" and "Irrelevant". The results showed that W2V+L2Q produced 20.5% more perfect fits (i.e. "Extremely Similar") than W2V (P-Value = 0.0292). On the flip side, W2V+L2Q also reduced the number of mediocre to bad fits by 27.5% (P-Value < 0.001).

4.3 L2Q For Weighted Jaccard Distance

Another example for leveraging L2Q in eBay's backend environment comes from the Entity Matching task [32]. Here, we tested improving the Jaccard distance measurement between different product titles (like with listings, each product entity in eBay's catalog has only one title). The Jaccard similarity index is defined as the number of identical tokens between two titles, relative to the total number of unique tokens in these titles. The Jaccard distance is computed by subtracting the Jaccard Index from 1. This metric can be highly advantageous from an engineering standpoint due to its simplicity, especially when implemented in large scale systems. Furthermore, the Jaccard distance is still prevalent in legacy production applications, mainly since replacing it is often costly. Therefore, our main motivation was to test if L2Q can significantly improve the results of a Jaccard-distance-driven system, while keeping a relatively low engineering footprint. We also presumed that L2Q can compliment the Jaccard distance well, due to the latter's tendency of giving an un-proportional weight to low-relevance tokens. The specific backend system that we tested on is intended to do Entity Matching to find potential duplicate product entities in eBay's catalog.

In Algorithm 1, we introduce a new approach for computing a weighted Jaccard similarity index/distance. Note that this algorithm is different than the one used in previous work [8, 14]. In our case, we plug the predictions from the L2Q model as weights.

To test the L2Q-weighted Jaccard distance in eBay's duplicated products detection system, we used 58,273 title pairs that were annotated by human agents specifically for this task. The set contains 9,084 true duplicate product pairs and 49,189 false duplicates. Note that the evaluation set contained title pairs with varied Jaccard

Table 1: The Recall, Precision and F1 score of the L2Q model in the sequence labeling task, with and without a CRF layer. Best F1 Score between the two is marked in bold. The number of test tokens in each vertical is denoted by n.

Vertical	n	Without CRF			With CRF		
		Precision	Recall	F1	Precision	Recall	F1
Electronics	403,629	0.80	0.66	0.72	0.78	0.69	**0.73**
Fashion	144,544	0.77	0.52	**0.62**	0.7	0.56	**0.62**
Media	20,222	0.67	0.63	0.65	0.66	0.68	**0.67**
Business & Industrial	55,673	0.72	0.49	0.58	0.73	0.59	**0.65**
Lifestyle	178,928	0.74	0.53	0.62	0.71	0.57	**0.63**
Home & Garden	157,295	0.68	0.44	**0.53**	0.66	0.42	0.51
Collectibles	119,651	0.79	0.53	0.63	0.74	0.57	**0.65**
Average		0.74	0.54	0.62	0.71	0.58	**0.64**

Table 2: Prediction examples of the L2Q model, which include each of the test token's expected likelihood to appear in the query, combined with its true proportion as observed on the site. In the bottom part we show the top 5 queries that have historically led users to each of the test listings alongside with their respective reoccurrence percentage.

Title 1	Apple	Watch	Series	4	GPS	40mm	Gold	Case	with	Pink	Sand	Sport	Band
L2Q Scores	0.95	0.95	0.63	0.82	0.12	0.18	0.15	0.01	0.00	0.09	0.01	0.01	0.02
True observed proportion	0.87	0.86	0.61	0.77	0.06	0.20	0.11	0.02	0.00	0.08	0.02	0.02	0.03

Top-5 queries	Query reoccurrence pct.
apple watch series 4	13
apple watch 4	6
apple watch series 4 40mm	6
series 4 apple watch	3
apple watch 4 40mm	2

Title 2	Cute	Cartoon	Silicone	Earphone	Protective	Cover	for	Apple	Airpods	Charging	Case
L2Q Scores	0.18	0.05	0.14	0.08	0.05	0.15	0.00	0.13	0.66	0.09	0.63
True observed proportion	0.11	0.05	0.06	0.02	0.01	0.24	0.00	0.11	0.68	0.01	0.59

Top-5 queries	Query reoccurrence pct.
airpods case	12
airpod case	5
airpods cover	3
airpods case cover	3
apple airpods case	3

Title 3	New	in	Sealed	Box	Samsung	Galaxy	Note	8	n950	USA	Unlocked
L2Q Scores	0.26	0.00	0.04	0.01	0.53	0.53	0.86	0.87	0.13	0.01	0.31
True observed proportion	0.22	0.00	0.03	0.02	0.54	0.50	0.89	0.85	0.00	0.01	0.31

Top-5 queries	Query reoccurrence pct.
note 8	15
samsung galaxy note 8	9
samsung note 8	9
galaxy note 8	8
samsung galaxy note 8 unlocked	4

Table 3: A qualitative comparison of listing-title nearest neighbors search results using different embedding methods.

(a) Test Title	**Fallout 4**: Game of the Year Edition (**Xbox One** XB1) Brand New Factory Sealed
W2V	Hitman Definitive Edition (Xbox One XB1 2018) Brand New Factory Sealed
W2V + TF-IDF	Fallout 3 Game of the Year Edition Xbox One Xbox 360 Brand New Factory Sealed
BERT	Star Wars: Battlefront II 2 (Xbox One XB1) Brand New Factory Sealed
BERT-FT	Fallout 4: Game of the Year Edition (PlayStation 4 PS4) Brand New Factory Sealed
W2V + L2Q	**Fallout 4**: Game of the Year Edition (Microsoft **Xbox One**, 2017) DLC INCLUDED
(b) Test Title	**Minnesota Wild** NHL Reebok Snapback Hat Green Black
W2V	New Dallas Stars Mens OSFA Flatbrim Snapback Black Green Reebok Hockey Hat
W2V + TF-IDF	Anaheim Ducks men’s Reebok baseball Snapback baseball hat new nhl
BERT	San Jose Sharks NHL Reebok Snapback Hat Cap
BERT-FT	New York Jets NFL Reebok Snapback Hat Cap White Green
W2V + L2Q	Fanatics Branded **Minnesota Wild** Black Team Haze Adjustable Snapback Hat
(c) Test Title	**OnePlus 5T** A5010 64GB (FACTORY UNLOCKED) Midnight Black
W2V	OnePlus 6t - 128GB -(A6010) Midnight Black (Unlocked)
W2V + TF-IDF	OnePlus 6T A6013 128GB + 8GB (FACTORY UNLOCKED) 6.41" Black Brand New
BERT	OnePlus 6 128GB A6000 (FACTORY UNLOCKED) 8GB RAM Black
BERT-FT	OnePlus 6T A6013 128GB + 8GB (FACTORY UNLOCKED) 6.41" Black Brand New
W2V + L2Q	**OnePlus 5T** A5010, Unlocked, Dual SIM, 6.01’, 15.9MP, Color Options (Unsealed)

distance values over the possible range, and it was not restricted to a specific range of values. Using the L2Q-weighted Jaccard distance metric increased the recall from 33% to 51%, in comparison to regular Jaccard distance, within a fixed False Positive Rate (FPR) range that is relevant for our business use case. Accordingly, using the L2Q-weighted Jaccard distance improved the area under the ROC curve from 0.88 to 0.9.

4.4 Online Experimentation

The first production downstream task that we deployed L2Q in was eBay’s Price Guidance. Price Guidance is a feature that lives within the new-listing creation form, and is intended to help sellers to decide what should be the listing price for their fixed-price listing. From a business perspective, the goal of the feature is to save the time it takes sellers to conduct market research, increase the likelihood of conversion (i.e. the listing getting sold), reduce the time it takes to convert, and ultimately, increase the liquidity of the marketplace. From a modeling standpoint, eBay has different models to create this guidance in different scenarios, mostly depending on the amount of data that is available about the listing. The current production Price Guidance model that was used as a benchmark is driven by a regression model that takes the prices of

Table 4: Model performance comparison of W2V, W2V weighted by TF-IDF (W2V+IDF), BERT, fine-tuned BERT (BERT+FT) and W2V weighted by L2Q (W2V+L2Q). Best results are marked in bold.

Method	same brand (%)	same GTIN (%)	same ISBN (%)
W2V	42.05	53.75	62.71
W2V+TFIDF	40.87	48.63	55.91
BERT	38.87	52.22	65.27
BERT+FT	41.07	54.91	67.41
W2V+L2Q	**48.13**	**55.82**	**69.47**
n	95,313	12,975	2,010

Algorithm 1 Weighted Jaccard Distance using L2Q

```
procedure WeightedJaccardDistance($textA, textB$)
    $aScores \leftarrow L2Q\_scores(textA)$
    $bScores \leftarrow L2Q\_scores(textB)$
    $sumAScores \leftarrow sum(aScores)$
    $sumBScores \leftarrow sum(bScores)$
    $intersectSc \leftarrow 0$
    for all $token$ in $textA$ do
        if $token$ in $textB$ then
            $intersectSc+ = aScores[token] + bScores[token]$
    $wJaccardSim \leftarrow intersectSc/(sumAScores + sumBScores)$
    $wJaccardDist \leftarrow 1 - wJaccardSimilarity$
    return $wJaccardDist$
```

similar listings as input, and outputs a guidance price that is based on a basic aggregation of those historical prices. The listing similarity method behind this model relies on eBay's main search engine, which is considered to be highly optimized for classic Information Retrieval tasks. To test the performance of the L2Q-driven similarity we switched the listing similarity mechanism to focus on minimizing the distance between the L2Q-weighted title Word2Vec vectors that were trained on eBay's corpus. Each of these title embedding vectors was computed as described in section 4.2.

Testing L2Q-driven Price Guidance in production environment required creating a scalable infrastructure to execute the model and to find the most similar listings. As for title vector computation, W2V vectors can be fetched from a pre-calculated dictionary, and are therefore easy to be used in production environment. In contrast, since L2Q predictions need to be computed in real-time, a dedicated service is required. Next, we used the locality-sensitive hashing (LSH) algorithm [13] to reduce the number of similar-listings candidates to a short-list, and a cosine similarity to find the best overall candidates. The service is implemented in Scala and Java programming languages, with the BLAS library [5] used for linear algebra calculations. The service is deployed in Tomcat containers on a cluster of 9 VM servers running Ubuntu Linux (each VM has 178 GB RAM, 24 CPU cores). L2Q model inference was done on NVIDIA Tesla V100 and P100 GPUs with dynamic resource allocation from a cluster of 432 GPUs. The entire process has a median latency of 88ms (0.95-quantile: 264 ms, 0.99-quantile: 374 ms).

Since there is no single price that is considered "right" for a product, there is no ground truth that we can compare our model's results to. This makes measuring success challenging. As a first step, we decided to measure the average difference between our suggested price and the price that was set by the seller, as done often to measure the accuracy of a regression task. However, unlike with classic regression, here we do not assume that the seller's price was necessarily set right. Instead, the idea behind this measure is to test the level of acceptance between the sellers and our price guidance. This is important since if sellers often find that the guidance "makes sense" then they would be more likely to adopt it, which can increase their trust with the guidance, and ultimately, help with the feature's long-term goals. Specifically, the model's impact was evaluated using three metrics - mean absolute error (MAE), root mean square error (RMSE) and the percentage of sellers which adopted our recommended price within a 5% bi-directional margin (P5). These serve as a proxy to the seller's acceptance with the guidance, and are correlated with long-term business success on the platform. The A/B test was conducted in June 2020 and lasted 30 days. During the test 10% of the sellers got the L2Q-driven guidance, while the rest got the default production model. The sub-group of sellers who received the L2Q-driven price was picked at random once before the test started. To assure that higher adoption rates are not driven by consistently higher price recommendations by one of the competing methods we examined the average paired difference between the two and found that it is close to zero. Approximately 1.6M listings were served with the old guidance and 187K were served with the new L2Q-driven guidance. The results showed that the L2Q-driven price guidance successfully lowered the MAE from 36.53 to 30.52 and the RMSE from 618.17 to 212.59 while increasing P5 adoption rate by 5.2% (all differences are statistically significant positive with p-value < 0.001). This shows that the guidance was more relevant and in-line with what the sellers expected, which implies that the underlying listings that the L2Q-driven similarity method compared with were more similar to the seed listing. Given the L2Q-driven guidance successful impact in driving seller adoption it is expected to be deployed in full scale for US-based eBay sellers on December 2020. The next step is to generalize the L2Q modeling approach to non-English eBay sites, including those in French, German and Italian.

5 DISCUSSION AND CONCLUSIONS

In this paper we demonstrate the advantages of leveraging users behavior data to improve a variety of title similarity tasks in eBay. As for the L2Q modeling approach, we find sequence labeling, which provides token importance weights, to be highly generalized for many downstream tasks. For example, apart from the success in the two use cases described above, we also experimented plugging L2Q to improve search query relaxation tasks. E.g., if a user is using an external source to copy-paste a full title into eBay's search, like "Fallout 4: Game of the Year Edition (Xbox One XB1) Brand New Factory Sealed", then L2Q can help relax the query by giving a low importance weight to the tokens with the lowest score – in this case, "Brand", "New", "Factory", "Sealed", "the" and "of".

Another potential use of the L2Q weights is with helping sellers to perform Search Engine Optimization (SEO) on their title, by suggesting the tokens that are most likely to be searched by potential buyers. Moreover, in a keyword-driven advertising platform, such an approach can help sellers decide which search tokens they should place a bid on, and the nominal distribution of such bids. Apart from being highly generalized, another advantage of the sequence labeling approach is the ability to potentially increase model interpretability. For example, when applied on a title similarity task, the system can visually show the seller what are the main criteria that it took under consideration when comparing their listing to other listings in the marketplace. From an engineering standpoint, this approach enables using L2Q as a separate layer to enrich different existing machine-learning models, in an almost like a plug-and-play-style integration, which leaves a low engineering footprint.

Apart from the sequence labeling modeling approach, we also experimented with training L2Q as a Convolution Neural Network (CNN) Seq2Seq model via the FAIRSEQ framework [25]. Overall, our results suggest that although this approach provides good analytical results (and perhaps deserves its own work), it is less generalized for different downstream tasks than sequence labeling. As far as advantages over sequence labeling, the output of the Seq2Seq flavor of L2Q was not limited to the input vocabulary of the title, and therefore, might be better suited for tasks like title SEO and search-query bid suggestions.

The approach presented in this work can be extended to other prediction tasks that involve behavioral elements from the users' journey. For example, predicting which filters users use on the search results page can be used as a proxy to learn the kind of aspects in the product that buyers care about the most. With this in mind, we think that L2Q is not solely restricted to eBay's case, and can be applied in various online platforms. The shared characteristics of those is the search-driven user experience, combined with a recall set that is comprised of documents that are relatively short.

ACKNOWLEDGMENTS

We wish to thank Jegan Gopalakrishnan Karunakaran for his contribution in developing the CNN modeling approach, Yotam Eshel for his assistance in the product deduplication work, and Ido Guy, Slava Novgorodov and Alex Nus for their helpful feedback.

REFERENCES

[1] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W. Bruce Croft. 2017. Learning a Hierarchical Embedding Model for Personalized Product Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 645–654. https://doi.org/10.1145/3077136.3080813

[2] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Association for Computational Linguistics, Minneapolis, Minnesota, 54–59. https://doi.org/10.18653/v1/N19-4010

[3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 1638–1649. https://www.aclweb.org/anthology/C18-1139

[4] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017 ; Conference date: 24-04-2017 Through 26-04-2017*.

[5] L Susan Blackford, Antoine Petitet, Roldan Pozo, Karin Remington, R Clint Whaley, James Demmel, Jack Dongarra, Iain Duff, Sven Hammarling, Greg Henry, et al. 2002. An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Software* 28, 2 (2002), 135–151.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. https://doi.org/10.1162/tacl_a_00051

[7] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. arXiv:cs.CL/1312.3005

[8] Ondrej Chum, James Philbin, and Andrew Zisserman. 2008. Near Duplicate Image Detection: min-Hash and tf-idf Weighting. *BMVC 2008 - Proceedings of the British Machine Vision Conference 2008*. https://doi.org/10.5244/C.22.50

[9] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. *CoRR* abs/1705.02364 (2017). arXiv:1705.02364 http://arxiv.org/abs/1705.02364

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[11] Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text Mining for Product Attribute Extraction. *SIGKDD Explor. Newsl.* 8, 1 (June 2006), 41–48. https://doi.org/10.1145/1147234.1147241

[12] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. http://arxiv.org/abs/1508.01991 cite arxiv:1508.01991.

[13] Piotr Indyk and Rajeev Motwani. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing (STOC '98)*. Association for Computing Machinery, New York, NY, USA, 604–613. https://doi.org/10.1145/276698.276876

[14] Sergey Ioffe. 2010. Improved Consistent Sampling, Weighted Minhash and L1 Sketching. In *2010 IEEE International Conference on Data Mining* (2010-12). IEEE, 246–255. https://doi.org/10.1109/ICDM.2010.80

[15] Paul Jaccard. 1901. Distribution de la Flore Alpine dans le Bassin des Dranses et dans quelques régions voisines. *Bulletin de la Societe Vaudoise des Sciences Naturelles* 37 (01 1901), 241–72. https://doi.org/10.5169/seals-266440

[16] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *CoRR* abs/1702.08734 (2017). arXiv:1702.08734 http://arxiv.org/abs/1702.08734

[17] Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 941–951. https://doi.org/10.18653/v1/P16-1089

[18] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:cs.LG/1412.6980

[19] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 3294–3302. http://papers.nips.cc/paper/5950-skip-thought-vectors.pdf

[20] Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *CoRR* abs/1803.02893 (2018). arXiv:1803.02893 http://arxiv.org/abs/1803.02893

[21] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1064–1074. https://doi.org/10.18653/v1/P16-1101

[22] Karin Mauge, Khash Rohanimanesh, and Jean-David Ruvini. 2012. Structuring E-Commerce Inventory. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1 (ACL '12)*. Association for Computational Linguistics, USA, 805–814.

[23] Gabor Melli. 2014. Shallow Semantic Parsing of Product Offering Titles (for Better Automatic Hyperlink Insertion). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. Association for Computing Machinery, New York, NY, USA, 1670–1678. https://doi.org/10.1145/2623330.2623343

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[25] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Association for Computational Linguistics, Minneapolis, Minnesota, 48–53. https://doi.org/10.18653/v1/N19-4009

[26] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. *CoRR* abs/1703.02507 (2017). arXiv:1703.02507 http://arxiv.org/abs/1703.02507

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[28] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[29] Petar Ristoski and Peter Mika. 2016. Enriching Product Ads with Metadata from HTML Annotations.. In *ESWC (Lecture Notes in Computer Science)*, Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange (Eds.), Vol. 9678. Springer, 151–167. http://dblp.uni-trier.de/db/conf/esws/eswc2016.html#RistoskiM16

[30] S. E. Robertson and S. Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR '94*, Bruce W. Croft and C. J. van Rijsbergen (Eds.). Springer London, London, 232–241.

[31] Kashif Shah, Selcuk Kopru, and Jean-David Ruvini. 2018. Neural Network based Extreme Classification and Similarity Models for Product Matching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. Association for Computational Linguistics, New Orleans - Louisiana, 8–15. https://doi.org/10.18653/v1/N18-3002

[32] Andreas Thor. 2010. Toward an adaptive string similarity measure for matching product offers. In *INFORMATIK 2010. Service Science – Neue Perspektiven für die Informatik. Band 1*, Klaus-Peter Fähnrich and Bogdan Franczyk (Eds.). Gesellschaft für Informatik e.V., Bonn, 702–710.

[33] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 225–232. https://doi.org/10.1145/2959100.2959160

[34] Han Xiao. 2018. bert-as-service. https://github.com/hanxiao/bert-as-service.

[35] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product Knowledge Graph Embedding for E-Commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 672–680. https://doi.org/10.1145/3336191.3371778