Language-Agnostic Representation Learning for Product Search on E-Commerce Platforms

Aman Ahuja*
Virginia Tech
Arlington, VA
aahuja@vt.edu

Nikhil Rao
Amazon
Palo Alto, CA
nikhilsr@amazon.com

Sumeet Katariya
Amazon
Palo Alto, CA
katsumee@amazon.com

Karthik Subbian
Amazon
Palo Alto, CA
ksubbian@amazon.com

Chandan K. Reddy
Virginia Tech
Arlington, VA
reddy@cs.vt.edu

ABSTRACT

Product search forms an indispensable component of any e-commerce service, and helps customers find products of their interest from a large catalog on these websites. When products that are irrelevant to the search query are surfaced, it leads to a poor customer experience, thus reducing user trust and increasing the likelihood of churn. While identifying and removing such results from product search is crucial, doing so is a burdensome task that requires large amounts of human annotated data to train accurate models. This problem is exacerbated when products are cross-listed across countries that speak multiple languages, and customers specify queries in multiple languages and from different cultural contexts. In this work, we propose a novel multi-lingual multi-task learning framework, to jointly train product search models on multiple languages, with limited amount of training data from each language. By aligning the query and product representations from different languages into a language-independent vector space of queries and products, respectively, the proposed model improves the performance over baseline search models in any given language. We evaluate the performance of our model on real data collected from a leading e-commerce service. Our experimental evaluation demonstrates up to 23% relative improvement in the classification F1-score compared to the state-of-the-art baseline models.

CCS CONCEPTS

• **Information systems** → **Content ranking**; **Query representation**; *Online shopping*; *Learning to rank*.

KEYWORDS

Product search, deep learning, E-commerce, multi-task learning, cross-lingual models

ACM Reference Format:
Aman Ahuja, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K. Reddy. 2020. Language-Agnostic Representation Learning for Product Search on E-Commerce Platforms. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371852

1 INTRODUCTION

Online retail stores such as Amazon and Walmart have now become an integral part of consumers' lifestyle. With an ever increasing catalog size, product search is the primary means by which a customer finds the specific item she is interested in. Moreover, product search is also used as a proxy for discovery when the customer does not have a specific item in mind, leading to queries like `'gifts for 10 year old boy'`. A good search engine should be able to parse any query provided by the customer, and display results that are most relevant. A consequence of expanding online stores world-wide is the listing of items across multiple countries, and the option for a customer to search for items using multiple languages. For instance, `amazon.com` allows customers to browse and shop in both English and Spanish (Español). Each version of the store shows products in the specific language (based on country or user preference) and allows search in that language. To ensure high customer satisfaction, the store should be able to surface relevant results for queries typed in multiple languages, across multiple countries.

Figure 1: (Top row) A simple translation of the query from English to Italian results in incorrect items being shown for the latter. (Bottom row) Illustration of our proposed method. The representations of products in languages L1 and L2 are learned so that similar products across languages (p1, p2 and P1) are closer. The same for queries (Q1, Q2, q1, q2).

*Work done during an internship at Amazon.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371852

Information Retrieval (IR) models are typically trained using historical data of query and click logs from the website. However, this data is often noisy and suffers from counterfactual biases. Thus, models trained on such data give results that often mismatch the user query intent [8]. Training a robust classifier that can categorize a query-item pair as relevant or not requires a large amount of human annotated data, which is an expensive and time consuming endeavor. The non-triviality of this task is further compounded by the listing of products in multiple countries, and users issuing queries in multiple languages in these stores. One might need to obtain large amounts of human annotated data in multiple languages.

Second challenge with generating sufficient training data across multiple countries is the rarity of irrelevant items. Most query-product samples collected from the search engine will have samples labeled as relevant, and obtaining a large enough set of irrelevant items from multiple countries is non-trivial.

Finally, items that are popular in one country might not be popular in another country, and there are cultural differences across countries that need to be modeled as well. For example, for the query `'men's running shoe'`, the most popular item in USA might not be the most popular item in India. Similarly, `'gifts for 10 year old boy'` are different in USA and India, due to varying cultures. For this reason, a simple translation of the queries and products (e.g., using a sequence to sequence model [30]) will not work. We experimentally verify this in the sequel (Section 4.6).

The above issues make it challenging to train accurate and robust relevance models for cross-lingual product search. However, training such a model is of extreme importance: it allows for sharing of (sparse) annotated data between countries, and a shared model will make it easier to maintain. Furthermore, multi-task learning has shown to be advantageous when compared to single task models, improving the performance of each task in the process. To this end, we propose **L**anguage-**A**gnostic **P**roduct **S**earch (**LAPS**), a novel multi-task learning strategy to learn a language-agnostic query and product representation for query-product matching, with a focus on improving cross-lingual query-product relevance in e-commerce online stores. Our work aims to reduce the number of mismatched items shown in response to a user-specified query across multiple languages across multiple countries. To be more precise, given a tuple (q, p) of a query q and a product title p, we want to classify whether this tuple is a match or not, based on the relevance between q and p [1]. Learning language-agnostic representations for queries and products allows us to learn a representation for queries from different languages into a common latent query space, and for products from different languages into a common latent product space, respectively. This representation can then be used to train a common shared classifier, that can classify whether a query-product pair is relevant, irrespective of its language. It also enables the classifier to jointly learn from the search data from different languages, as the classifier now uses the representation from the shared vector spaces. Having a common representation allows the classifier to be more robust, by accounting for inter-language vagaries. Once a shared representation is learned, and we have a means to embed queries and items in their language-agnostic latent spaces, we can generalize the classifier to new languages.

The contributions of our work are as follows: We first propose an efficient product classification model that takes as input a user query, and a product title, and classifies whether they are relevant or not. We use transformer-based encoders to learn the representation for both query and product titles. In contrast to the traditional recurrent neural network (RNN) based models, transformer units can compute the text sequence representation in a fully parallelized manner, making our model an ideal fit for production environments, where latency is a primary concern. Second, to overcome the problem of limited availability of training data, we propose a novel mechanism to learn language-independent representations for query and product titles, so that the model can jointly utilize data from multiple languages for training. Our framework is a multi-task model, where each task corresponds to a language, with the shared parameters being from the classifier. We learn language-specific query and product title encoders. We perform extensive qualitative and quantitative experiments on data from an e-commerce website and show that the proposed LAPS model outperforms several mono-lingual and multi-lingual baselines. To the best of our knowledge, our work is the first to address the problem of cross-lingual information retrieval in a product search context, where similarities between queries are based on items purchased and not a simple translation.

The rest of this paper is organized as follows: Section 2 provides an overview of the existing techniques related to the work proposed in this paper. In Section 3, we introduce the proposed LAPS model, and provide details regarding the training and optimization process. Section 4 describes the details of the experimental evaluation including baseline techniques and the results obtained from our empirical evaluation. Finally, Section 5 concludes the paper, with possible directions for future work.

2 RELATED WORK

Search and Ranking Models: Most of the earlier works in the information retrieval domain are inspired by the keyword-based methods, which were first used for document retrieval [26, 27]. Recently, many neural network based ranking models have been proposed, primarily focused on text search. Such methods can broadly be grouped into two categories: Representation based methods learn a fixed vector representation for queries and documents, and then compute their relevance. Examples of such methods include Deep Semantic Search Model (DSSM) [14] and Convolutional DSSM (CDSSM) [29]. These methods learn query and document embeddings by using n-grams, which do not effectively learn semantic intent. This is overcome by using word embeddings by ARC-1 model [13]. Another class of neural ranking models are interaction-based methods, that first compute interaction matrices between query and document vectors, and then pass this matrix through a classification network. MatchPyramid [23] belongs to this class of models, where it uses fixed word embeddings to compute the interaction matrix. This does not encode the contextual dependencies in the representation, and can lead to poor performance. MV-LSTM [32] and HAR [35] address this problem by using Bi-RNN encoders to encode query and document word vectors, and then compute

[1] Note that the products that are finally shown to the user depends on multiple factors beyond relevance, but that is beyond the scope of this paper.

their relevance. However, the use of LSTM/GRU layers makes these models computationally slow. Moreover, interaction-based methods cannot be used for online inference, since they need to compute interaction vector of a query with every document in the corpus.

Product Search and Representation Learning: E-commerce search is a broad area of research, and has multiple facets and unique challenges, as discussed in [15]. Several different techniques, ranging from theory-based models [18], to keyword-based models [9], have been proposed in this domain. The authors of [20] proposed an e-commerce search model that ranks best-selling products higher in product search. The work in [7] studied the importance of visual attractiveness in e-commerce search. Another line of work in this domain focuses on the impact of product diversity in order to improve the search results [25]. In the e-commerce search domain, a majority of recent works focus on utilizing user activity data (such as browsing history and click-through rate) for personalizing search results [4, 22, 33]. These behavioural models utilize the implicit feedback signals from customers, based on their browsing history, for improving the search results. However, there exists a very fat tail of user queries for which there is no reliable behavioral data, and hence methods that rely on user behavior typically do not generalize well.

Machine Translation and Cross-Lingual Search: Machine translation using sequence-to-sequence models is a popular topic of research [2, 30]. In [19], the authors propose a translation-based methodology for product categorization in e-commerce platforms. However, translation-based methods are aimed at generating a text sequence in a target language, given an input in a different language. Using neural networks for semantic similarity between a pair of text sequences has been studied before in [24, 34], and for matching sentences from different languages in [5]. A key difference in our context is that we are interested in cross-lingual product search, where notions such as semantic similarity carry less meaning. Most previous works on cross-lingual IR (CLIR) has primarily focused on dictionary based methods to translate queries between languages [3]. In [28], the authors propose to learn shared representations for CLIR, but the representations depend on language pairs, and no effort is made to align queries across languages in a latent space. Moreover, works on standard IR does not carry over seamlessly to product search, due to various challenges mentioned above.

3 THE PROPOSED MODEL

In this section, we introduce our proposed Language-Agnostic Product Search (LAPS) model, which is a neural network based model which can classify whether a given query-product pair is relevant or not, and can jointly be trained using the data from various languages.

3.1 Problem Setup and Notations

Let us assume that there are a set of languages $L = \{l_1, .., l_k\}$, and human annotated query-product data $D^{cls} = \{d_{l1}, .., d_{l_k}\}$ from each of these languages. Let $d_{l_i} = \{(q_1^{l_i}, p_1^{l_i}, y_1^{l_i}), .., (q_{|l_i|}^{l_i}, p_{|l_i|}^{l_i}, , y_{|l_i|}^{l_i})\}$, where $|l_i|$ is the number of pairs from the language l_i. In this paper, we assume the labels $y_j^{l_i}$ to be of binary relevance, but the method that is being proposed in this work can be trivially extended to other related problems such as regression, ranking, etc. The main goal of this work is to build a model that can classify an unknown query-item pair in any language in L as being relevant or not.

To further guide the model training, we also have two additional datasets. $D^{QQ} = \{d_{l_i,l_j}^{qq}\}_{i,j=1,i\neq j}^{k}$ is a query-query dataset across pairs of languages, such that each sample corresponds to a pair of queries that should be "aligned" (to be made clear later). Similarly, we also assume we have a product-product dataset $D^{pp} = \{d_{l_i,l_j}^{pp}\}_{i,j=1,i\neq j}^{k}$. In the cross-lingual e-commerce search that is being investigated in this work, D^{PP} are pairs of products that are cross-listed across countries. D^{QQ} consists of query pairs in two countries that led to the purchase of the same (cross-listed) item.

3.2 Model Architecture

The basic building blocks of LAPS are described in detail below. Figure 2 shows a detailed schematic of the architecture.

3.2.1 Word Embeddings and Dynamic Vocabulary. To feed the input query and product title word features to our model, we use an embedding lookup layer in our model. This layer takes as input the raw word tokens $\{w_1^q, .., w_m^q\}$ for the query q and $\{w_1^p, .., w_n^p\}$ for the product title p, and returns the word embedding vectors $\{e_1^q, .., e_m^q\}$ and $\{e_1^p, .., e_n^p\}$ for q and p, respectively.

In contrast to the traditional text data, we require our models to deal with a dynamic vocabulary, since items are continually being listed and de-listed from the service, and user preferences change over time and display seasonality. We use Sentencepiece [17], a kind of subword tokenization to deal with this problem. Second, user queries and item titles are structured very differently from traditional "natural" language. Hence, we create separate corpora of anonymized user query logs and item titles for every language, and train our sentencepiece embeddings separately for queries and titles, for each language.

3.2.2 Encoders for Query and Product Titles. We use a transformer [31] based encoder to embed the contextual information in the query and title sequences, as they are computationally cheaper than recurrent architectures such as RNN, GRU, and LSTM [6, 12]. Furthermore, recurrent architectures compute word representations conditioned on adjoining words, and as we mentioned earlier, user queries and item titles do not follow standard natural language structure to make recurrent units meaningful.

The encoder consists of a layer of transformer units, that takes as input the word vectors, and returns the contextual representation for each of these words in the sequence. Given the embedding vectors $\{e_1^q, .., e_m^q\}$ and $\{e_1^p, .., e_n^p\}$ from the previous layer, the encoders return the contextual representations $U^q = \{u_1^q, .., u_m^q\}$ and $U^p = \{u_1^p, .., u_n^p\}$, respectively.

3.2.3 Self-Attention Layer. To compute fixed-dimensional vectors for the query and product titles, we utilize a self-attention pooling layer with a scaled dot-product attention [31]. This choice was made considering the fact that it is much faster compared to other attention mechanisms such as [2] or [21], while giving a performance similar to others. This layer takes the output of the query and product transformer encoders U^q and U^p, and returns a pooled representation v^q and v^p, respectively, where

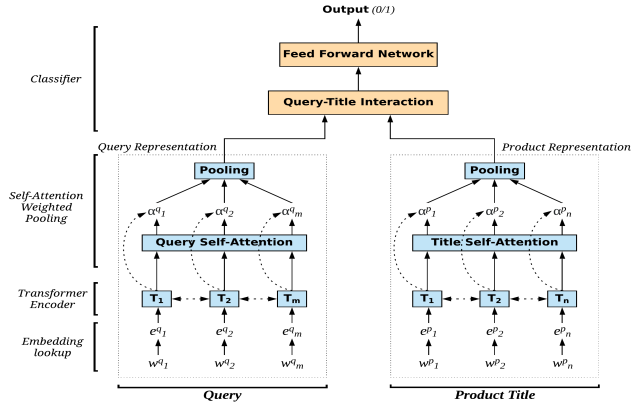


Figure 2: Transformer-based Mono-lingual product search model.

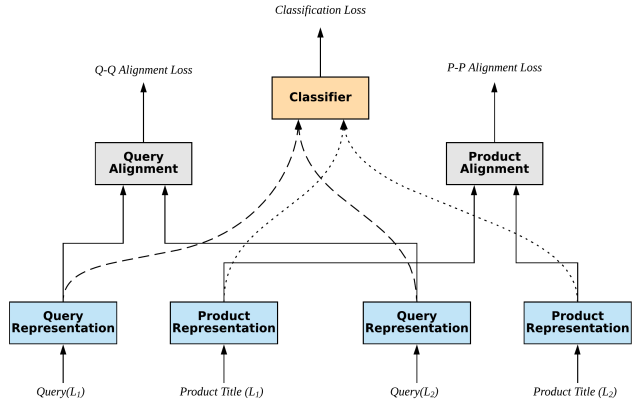


Figure 3: Bi-lingual product search model. The model from Figure 2 is used to learn the query and product representations, but an additional alignment module is added (in gray blocks), while the underlying classifier is shared. The alignment module helps align the query and product representations across two languages.

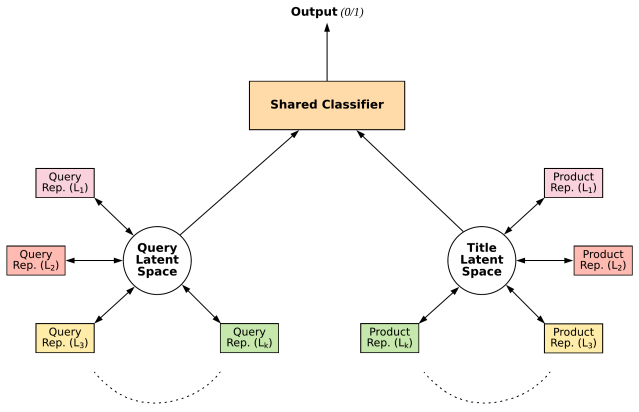


Figure 4: Multi-lingual product search model. This architecture is an extension of the bi-lingual model shown in Figure 3 to multiple languages. The alignment module aligns query and item representations across several languages while the underlying classifier is shared.

$$a_i^q = \frac{u_i^q (u_i^q)^T}{\sqrt{d_u}}, \qquad \alpha_i^q = \frac{\exp(a_i^q)}{\sum_{j=1}^{m} \exp(a_j^q)}, \qquad v_q = \sum_{t=1}^{m} \alpha_t^q u_t^q \quad (1)$$

Here, d_u is the dimension of the transformer outputs. The representation v_p of the product title can also be computed similarly.

3.2.4 Classifier. The query and product feature vectors are fed to a classifier, which computes the final relevance label. The classifier has two main components:

Interaction Layer: After learning the query and product representations, we compute the interactions between these two representations. The interaction vector x is computed as follows: $x = [v_q, v_p, v_q \odot v_p]$. $\odot$ denotes the Hadamard product.

Classification Layer: The interaction vector x is fed to a classification module that computes the final relevance label r. This module is a 3-layer feed-forward network with Rectified Linear Unit (ReLU) activation function.

Algorithm 1 Training Language-Agnostic Product Search Model

Require: Language set L, Classification dataset D^{cls}, alignment datasets D^{QQ}, D^{PP}, number of epochs N_{epochs}, network hyperparameters, step sizes η, γ.

1: Initialize model parameters $\{\theta_q^{l_i}\}_{i=1}^k, \{\theta_p^{l_i}\}_{i=1}^k, \theta_{cls}$.
2: Pre-train the classifier and the encoders using the procedure described in Section 4.5
3: **for** $epoch \leftarrow 1$ to N_{epochs} **do**
4: **for all** $l_i \in L$ **do**
5: **for all** $j \neq i$ **do**
6: Train $\theta_q^{l_i}, \theta_q^{l_j}$ on the query alignment task using d_{l_i,l_j}^{qq}.
$\theta_q^{l_i} \leftarrow \theta_q^{l_i} - \gamma \nabla_{\theta_q^{l_i}} \mathcal{L}_{l_i,l_j}^{QQ} \quad \theta_q^{l_j} \leftarrow \theta_q^{l_j} - \gamma \nabla_{\theta_q^{l_j}} \mathcal{L}_{l_i,l_j}^{QQ}$
7: Train $\theta_p^{l_i}, \theta_p^{l_j}$ on the product alignment task using d_{l_i,l_j}^{pp}.
$\theta_p^{l_i} \leftarrow \theta_p^{l_i} - \gamma \nabla_{\theta_p^{l_i}} \mathcal{L}_{l_i,l_j}^{PP} \quad \theta_p^{l_j} \leftarrow \theta_p^{l_j} - \gamma \nabla_{\theta_p^{l_j}} \mathcal{L}_{l_i,l_j}^{PP}$
8: Train $\theta_q^{l_j}, \theta_p^{l_j}, \theta_{cls}$ on classification task using $d_{l_j}^{cls}$.
$\theta_q^{l_i} \leftarrow \theta_q^{l_i} - \eta \nabla_{\theta_q^{l_i}} \mathcal{L}_{l_i}^{cls} \quad \theta_p^{l_i} \leftarrow \theta_p^{l_i} - \eta \nabla_{\theta_p^{l_i}} \mathcal{L}_{l_i}^{cls}$
$\theta_{cls} \leftarrow \theta_{cls} - \eta \nabla_{\theta_{cls}} \mathcal{L}_{l_i}^{cls}$

3.3 Language Alignment for Queries and Product Titles

The next step is to align queries and product titles across multiple languages, in order to utilize a shared classifier. Specifically, we need to embed queries (or products) in a universal query (or product) vector space that is language-agnostic. To have a meaningful language-agnostic representation, we need to ensure that queries from multiple languages that have the same product intent are aligned in the universal query latent space. For example, consider the queries '`audio jack cable`' and '`aux cables`' in English, and '`aux kabel`' and '`kopfhörerkabel`' in German. All these queries have a similar product intent, which is aux cables. Hence, in the language-agnostic query latent vector space, they

should be close to each other. Similarly, titles of same products in different languages, such as 'Firplast Round Cardboard' and 'Firplast Rund aus Pappe', should be aligned close to each other in the language-agnostic product vector space.

To this end, we use two alignment modules, one for the queries, and other for the product titles. Each of these alignment modules ensures the language-independent alignment of queries and product titles in their individual common vector spaces. Figure 3 illustrates this concept for two languages. Given two queries, q^{l_i} and q^{l_j}, from languages l_i and l_j, respectively, we first compute their pooled representation using their respective language-specific encoders followed by pooling, to get the representations $v_q^{l_i}$ and $v_q^{l_j}$, respectively (Equation (1)). We then minimize the squared distance $d(v_q^{l_i}, v_q^{l_j})$ between these two vectors, which we term as the query-query alignment loss. A similar process is carried out for the product titles to align the pooled representations from different languages with each other. More importantly, while training the alignment module, only the weights of the respective encoders are allowed to be tuned, and all other encoder weights are frozen.

3.4 Shared Classifier for Query-Product Pairs from Multiple Languages

Figure 4 shows our complete LAPS architecture. We repeat the architecture for two languages described above to the set of all languages L. Finally, we obtain a language-agnostic query (and product) space, where the respective encoders can embed queries and product titles. These query-title pairs are fed to a common classification module (Figure 2) to obtain the relevance score for the query-product pair. While training the classifier, we only train the weights of the classifier, and the query and product title encoders of the language to which the query-product pair belongs to. The weights of all other encoders are frozen.

3.5 Optimization

We will now provide additional details about model training. Note that the dataset D^{cls} is made up of human annotated data from multiple languages, and is relatively small. D^{PP} can be obtained from product catalogs, while D^{QQ} can be obtained from anonymous user logs. In our multi-task learning based framework that optimizes both classification and the query-product alignment tasks, each dataset is used to optimize the loss corresponding to the using parts of our LAPS model shown in Figure 4.

Classification Loss: We use a weighted binary cross entropy for D^{cls}, to account for the heavy class imbalance. Specifically, since the dataset is created from sampling search engine results, there exists far more relevant items than irrelevant ones. For a given language l_i, the classification loss is given by:

$$\mathcal{L}_{l_i}^{cls} = \sum_{(q_j^{l_i}, p_j^{l_i}), y_j^{l_i} \in d_{l_i}^{cls}} y_j^{l_i} \log f^{cls}(q_j^{l_i}, p_j^{l_i}) + \alpha * (1 - y_j^{l_i}) (1 - \log f^{cls}(q_j^{l_i}, p_j^{l_i})) \quad (2)$$

Here, α is the scaling factor for the minority class (in our case, is the irrelevant pairs with label = 0). $f^{cls}(\cdot)$ is the classifier for LAPS.

Alignment Loss: To align the query-query and product-product representations together, we need to ensure that the queries in different languages with similar product intent are close to each-other in the feature space. We use the squared Euclidean distance as the objective function for the alignment task. Given a pair of languages l_i and l_j, the objective function is:

$$\mathcal{L}_{l_i,l_j}^{QQ} = \sum_{(q^{l_i}, q^{l_j}) \in d_{l_i,l_j}^{qq}} ||v_q^{l_i} - v_q^{l_j}||_2^2, \quad (3)$$

where the minimization is over all pairs in D^{QQ}. A similar loss is applied for all pairs in D^{PP}:

$$\mathcal{L}_{l_i,l_j}^{PP} = \sum_{(p^{l_i}, p^{l_j}) \in d_{l_i,l_j}^{pp}} ||v_p^{l_i} - v_p^{l_j}||_2^2 \quad (4)$$

The detailed training procedure is described in Algorithm 1.

4 EXPERIMENTAL RESULTS

In this section, we perform an extensive set of experiments on anonymized search data logs from Amazon.com, with products listed in multiple countries and languages. First, we present quantitative results, comparing LAPS with several baselines, and evaluating it's performance on the relevance task, as well as how well LAPS aligns queries and titles in the language-agnostic space. Next, we also include qualitative results, studying the quality of the query and product alignment in the aforementioned vector space.

4.1 Dataset

The data we use in our experiments is collected from five different languages: French *(FR)*, Spanish (*ES)*, Italian (*IT)*, English *(EN)*, and German *(DE)*. As explained in section 3.1, we need 3 datasets. We obtain D^{cls} via human annotated query-product pairs sampled from the search results in each of the above country-specific services. The annotators return a binary label that indicates the item's relevance to the query. Since this dataset consists of human labels, the relative number of pairs is small. We provide some basic statistics of our datasets in Table 1.

Table 1: Samples per country in D^{cls}. The positive and negative labels refer to relevant and irrelevant samples.

Dataset	# Train pairs (+ | -)	# Test pairs (+ | -)
FR	42,299 (37,267 | 5032)	20,797 (17,930 | 2,867)
ES	59,223 (50,813 | 8,410)	23,395 (19,973 | 3,422)
IT	118,404 (91,147 | 27,257)	46,309 (38,908 | 7,401)
EN	313,390 (289,864 | 23,526)	48,504 (45,638 | 2,866)
DE	287,346 (267,990 | 19,356)	64,392 (58,952 | 5,440)

We next describe the query and product alignment datasets, D^{QQ} and D^{PP}. For the former, we use anonymous user logs across the above countries. To create a query-query pair, we consider an item that is cross-listed across different countries. Then, we let $q_i, q_j \in D^{QQ}$ if q_i was used to purchase the item in country i, and q_j was used to purchase the same item in country j. Thus, the pairs of queries that we want to align are not perfect translations of each other, but instead queries that have the same inherent product in

mind. This is important, since the same item might be purchased in two countries via queries that are not translations. The dataset D^{PP} is easier to create, since we already have the cross-listed items from the catalog across the countries. Hence, for both query-query and product-product alignment tasks, we have ${}^5C_2 = 10$ pairwise combinations. Detailed statistics are shown in Table 2.

Table 2: Number of samples in the Alignment datasets.

Dataset	#QQ pairs (Train | Test)	#PP (Train | Test)
FR-ES	444,793 | 23,430	679,420 | 32,622
FR-IT	581,536 | 30,639	686,894 | 36,037
FR-EN	762,100 | 37,900	682,948 | 35,944
FR-DE	765,520 | 34,480	751,774 | 39,324
ES-IT	415,814 | 21,891	680,797 | 35,388
ES-EN	592,018 | 31,234	676,631 | 5,812
ES-DE	640,057 | 33,711	680,164 | 35,922
IT-EN	766,088 | 33,912	739,443 | 39,188
IT-DE	766,801 | 33,199	748,901 | 39,637
EN-DE	777,001 | 22,999	745,929 | 39,526

4.2 Evaluation Metrics

For the quantitative evaluation of our model against baseline techniques, we use Precision, Recall, and F1-Score, since we deal with classification tasks. For measuring the performance of the alignment tasks, we use Recall@k for $k \in \{1, 3, 5\}$. We use the notations P, R, F to denote Precision, Recall, and F1-score, respectively.

4.3 Baseline Methods

We compare our LAPS model against the following baselines:

- **Mono-lingual Model:** The simple transformer-based mono-lingual relevance model (as shown in Figure 2). The model takes a query and a product title as the input, encodes them through a transformer and self attention, and computes their relevance using the classifier. This is the simplest baseline model that we will use for comparison.
- **Translation to English:** We first train the mono-lingual model mentioned above on English data, since that is the largest amongst the classification datasets we have. For all other languages, we translate the test queries and product titles to English using AWS translate, a publicly available translation service, and then evaluate the performance using the pre-trained mono-lingual model.
- **Translation to Italian:** Since *IT* was the best performing mono-lingual model, we hypothesized that translation to Italian could potentially give a good performance. Hence, similar to the above baseline, we translate the query and product titles from other languages to Italian, and then evaluate the performance on the mono-lingual *IT* model.
- **MatchPyramid [23]:** This model computes pair-wise interactions between the word vectors of two sequences, which is then passed through a CNN to compute the relevance between the sequences. The interaction matrix is independent of the input features. Hence, we compute the interaction matrix of (q, p) pair using their language-specific word vectors, and then use a common CNN shared across all languages for relevance computation.
- **MV-LSTM [32]:** This model computes the contextual representation of the two text sequences using Bi-LSTM encoders, and then the interaction matrix using these contextual word representations. It then uses a feed-forward network for computing relevance.
- **Model-Agnostic Meta Learning [11]:** To study the impact of using cross-lingual training over more sophisticated approaches such as meta learning, we use this baseline for our experiments. In our implementation, we regard each language as one task, and hence, given a query and product title from this task, the objective of the model is to predict their relevance. The model uses a shared classifier, but different encoders for queries and titles for each language.

4.4 Implementation Details

We implemented our model in TensorFlow [1]. The model was trained using Adam optimizer [16] with a learning rate of 10^{-4}. The encoders for query and titles for all the languages used two attention heads, with an output embedding dimension of 256. The model was trained using a dropout of 0.3. For each country, the initial token embedding dimension was fixed to be 256. The value of the cross entropy weight for the minority defect class was set to 10. All these hyperparameter values were obtained based on the empirical results.

4.5 Model Initialization and Pre-training

Since pre-training [10] generally helps in improving the performance of neural models by providing a good baseline initialization for complex models, we randomly choose one language $l_i \in L$, and train the corresponding encoders $\theta_q^{l_i}$ and $\theta_p^{l_i}$, and the shared classifier θ_{cls}, using the classification data from l_i. Then, we fix the classification weights and the encoder weights for l_i, and $\forall l_j \in L \backslash l_i$. We pre-train the corresponding transformer encoders by performing the pair-wise training using $d^{qq}_{l_i,l_j}$ and $d^{pp}_{l_i,l_j}$.

4.6 Quantitative Results

We will now present the results obtained in the empirical evaluation of our model. We refer to our LAPS model trained on T languages as LAPS-T. In Table 3, LAPS-2 refers to LAPS with 2 languages. Also, the notation LAPS-2(+FR) refers to the LAPS model trained on FR and the language corresponding to the particular column.

Model architecture analysis: From Table 3, we can observe that MatchPyramid and MV-LSTM have significantly lower performance compared to the mono-lingual version of our transformer model. Since MatchPyramid uses word vectors to compute the similarity matrix, it's performance is largely dependent on the quality of word embeddings. However, as mentioned earlier, in the product search scenario, the word vectors do not have rich semantic information. MV-LSTM includes contextual information via a Bi-LSTM, but we can observe that the transformer based models work better, due to lack of proper ordering of words in user queries.

Multi-lingual learning analysis: We observe that using multi-lingual training significantly improves the performance of the search defect model. Even the bi-lingual LAPS significantly outperforms the other baselines by a wide margin. This can be attributed

Table 3: Quantitative comparison results of the proposed LAPS model with several state-of-the-art models on five different query-product datasets from various European languages. We observe that the penta-lingual (LAPS-5) model achieves the best results compared to using fewer languages, and also the other baseline methods. NA indicates 'Not Applicable'.

Model	Dataset														
	FR			ES			IT			EN			DE		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Mono-lingual	40.93	44.15	42.48	41.54	44.72	43.07	49.93	49.86	49.89	38.98	38.51	38.74	34.30	37.61	35.88
Translate (EN)	35.89	49.93	41.76	31.08	56.01	39.97	34.44	49.75	40.70	38.98	38.51	38.74	22.95	34.85	27.68
Translate (IT)	24.68	**60.89**	35.12	23.86	54.05	33.10	49.93	49.86	49.89	12.34	**49.09**	19.72	12.56	48.09	19.92
MV-LSTM [32]	30.59	53.09	38.82	30.19	43.12	35.51	30.08	52.87	38.34	30.97	31.93	31.44	29.03	**51.07**	37.02
MatchPyramid [23]	19.02	53.71	28.09	17.63	**63.61**	27.61	22.09	**55.07**	31.53	9.29	40.93	15.15	13.01	46.40	20.33
MAML [11]	42.97	42.26	42.61	43.88	51.76	47.50	49.04	52.10	50.52	32.06	41.60	36.21	36.85	37.85	37.34
LAPS-2(+ FR)	NA	NA	NA	47.19	45.13	46.14	50.42	51.08	50.75	38.38	36.13	37.22	37.47	41.12	39.21
LAPS-2(+ ES)	41.25	47.72	44.25	NA	NA	NA	52.34	48.41	50.30	37.07	39.01	38.01	39.71	40.24	39.97
LAPS-2(+ IT)	49.53	42.89	45.97	40.94	54.46	46.74	NA	NA	NA	36.94	41.95	39.29	43.18	40.68	41.89
LAPS-2(+ EN)	50.75	46.15	48.34	52.17	52.89	52.33	49.87	54.27	51.98	NA	NA	NA	43.51	43.18	43.35
LAPS-2(+ DE)	45.90	48.81	47.31	51.51	54.08	52.76	50.71	51.05	50.88	37.74	40.09	38.88	NA	NA	NA
LAPS-3(ES+DE+UK)	NA	NA	NA	51.74	53.05	52.39	NA	NA	NA	39.52	40.58	40.04	44.50	42.09	43.26
LAPS-3(IT+DE+UK)	NA	NA	NA	NA	NA	NA	55.50	49.68	52.43	38.81	40.49	39.63	**45.61**	41.12	43.25
LAPS-5	**53.47**	52.28	**52.87**	**56.83**	54.63	**55.71**	**57.93**	52.23	**54.93**	**39.54**	41.63	**40.56**	44.80	43.77	**44.28**

Table 4: Similar Query and similar Product results using the language agnostic vectors. The upper (red) and lower (green) triangular portions correspond to product and query alignments respectively. For each language pair, the first row represents the recall values obtained by training a bi-lingual model, and the second row is the recall of LAPS-5.

Dataset		Dataset														
		FR			ES			IT			EN			DE		
		R@1	R@3	R@5	R@1	R@3	R@5	R@1	R@3	R@5	R@1	R@3	R@5	R@1	R@3	R@5
FR		NA	NA	NA	26	39	45	14	25	31	19	31	37	20	32	39
		NA	NA	NA	41	56	63	38	53	60	35	50	58	38	54	61
ES		30	39	42	NA	NA	NA	11	20	25	33	50	59	19	33	40
		65	69	71	NA	NA	NA	37	52	58	52	68	75	31	45	53
IT		28	39	44	36	44	46	NA	NA	NA	26	40	46	23	37	44
		46	48	51	54	58	62	NA	NA	NA	35	50	57	31	46	53
EN		57	63	65	56	60	62	64	71	73	NA	NA	NA	37	51	58
		66	74	76	73	75	77	66	74	76	NA	NA	NA	32	46	53
DE		28	37	40	54	64	66	43	48	51	67	71	74	NA	NA	NA
		53	62	69	31	56	59	47	55	58	71	74	77	NA	NA	NA

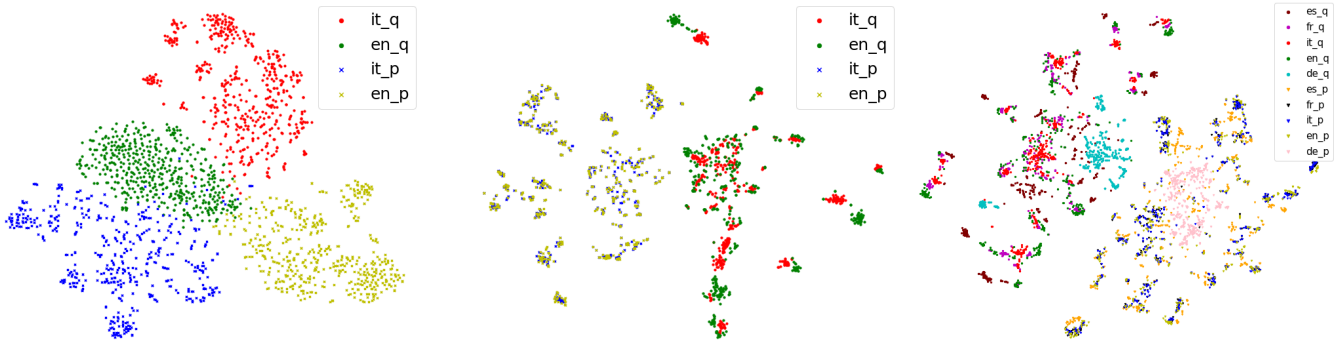
to the fact that multi-lingual training helps in overcoming the data sparsity issue, and exposes the model to more training data (covering a wide range of cases), which otherwise might not be present in the single language data. Translation baselines have not been trained to learn queries that are similar based on purchases, but rather trained to produce accurate natural language translations. This makes it hard for the model for discover semantic and product intent of the query. Using data from multiple languages also helps the model learn a richer semantic understanding of the queries and products, since we use similar tuples from multiple language pairs in the training process.

Impact of number of languages used in training: In general, we observe that adding more languages in the training dataset helps in improving the model performance. The tri-lingual model outperforms the bi-lingual models in almost all the cases (except for DE where the performance is at par with the bi-lingual models). The penta-lingual model significantly outperforms all the other versions of LAPS, which indicates that adding more languages is generally better, as it exposes the model to more training data.

Table 4 quantifies the performance of our alignment module into a language-agnostic query and product space. The lower triangular portion represents Recall@k values for query alignment, and the upper triangular portion for the product alignment. The recall is computed by using the ground truth query and item pairs in the D^{QQ} and D^{PP} test datasets, and considering the K nearest neighbors by Euclidean distance over the embeddings of that particular language. With the exception of DE-ES (query-query) and EN-DE (product-product), we see that the LAPS-5 model significantly improves upon the recall values compared to a bilingual (LAPS-2) alignment, indicating that the presence of other languages strongly forces the "correct" alignment.

Table 5: Nearest Neighbors of Queries in one language, for all other languages.

Source Query (EN): *desktop speakers*			
FR	**ES**	**IT**	**DE**
sony srs-xb10 noir	altavoces ordenador sobremesa	cassa bluetooth sony extra bass	lautsprecher pc
haut parleur bluetooth	altavoz ordenador	cassa bose bluetooth portatile	bluetooth für den pc
mini enceinte bluetooth	altavoces pc	cassa jbl bluetooth portatile powerful	pc boxen
haut parleur pc	microfono pc	jbl flip 4 speaker	lautsprecher laptop
Source Query (DE): *microsd-karte*			
FR	**ES**	**IT**	**EN**
carte sd classe 10	micro sd 64 gb clase 10	scheda sd 64 giga huawei	sandisk extreme micro sd 64gb
carte memoire sdxc	tarjeta memoria	scheda sd 128gb	micro sd card sandisk
carte micro sd	tarjeta sd 64 gb	memoria sd 16 gb	sd cards 64gb
carte micro sd pour switch	tarjeta memoria movil	scheda micro sd	micro sd card 64



(a) Mono-lingual models trained on IT and EN. The queries and items for two languages are spread out, since no effort is made to align them.

(b) LAPS-5 (IT and EN selected) model results. The same as in Figure 5c, with IT/EN highlighted.

(c) LAPS-5 model results. Query and item embeddings separate out (approximately along the 45° line). Across languages, queries and items are intermingled, showing that the L2 loss that we use helps in alignment.

Figure 5: TSNE embeddings for the query and item representations learned from Mono-lingual and LAPS-5 models.

Table 6: Same language nearest neighbors of queries.

Source Query (EN): *desktop speakers*
bluetooth speakers
usb speakers
wireless speakers bluetooth powerful
portable speaker
Source Query (DE): *microsd-karte*
sd karte micro 32 gb
micro sd karte 64
mini sd karte
micro sd - karte

4.7 Qualitative Results

For a qualitative evaluation, we analyze the language-agnostic embedding space obtained by our model. We obtain the nearest neighbor queries for a source language in all other languages in Table 5, and for the same language in Table 6. We also plot the query and item embeddings using t-distributed stochastic neighbor embedding (TSNE) method in Figure 5. From Figure 5a, we can observe that a model trained separately on single language obtains good separation of queries and items, but no alignment. However, in Figure 5c, LAPS-5 aligns queries and items across multiple languages while maintaining query-item separability.

5 CONCLUSION

In this paper, we presented Language-Agnostic Product Search (LAPS), a novel neural model to predict the relevance of a query-product pair for improving e-commerce search across multiple languages. The model utilizes training data from multiple languages and uses cross-lingual training to learn a language-independent representation for queries and product titles. We use an efficient transformer-based network with self attention pooling, to learn the representations for queries and product titles. By using data across multiple languages, we showed that we improve the performance of our model over baselines trained on single language. Experimental evaluation with state-of-the-art baselines suggest that LAPS gives significant reduction in the number of irrelevant search results.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Lerning Representations (ICLR)* (2015).

[3] Lisa Ballesteros and Bruce Croft. 1996. Dictionary methods for cross-lingual information retrieval. In *International Conference on Database and Expert Systems Applications*. Springer, 791–801.

[4] Keping Bi, Choon Hui Teo, Yesh Dattatreya, Vijai Mohan, and W Bruce Croft. 2019. Leverage Implicit Feedback for Context-aware Product Search. (2019).

[5] Johannes Bjerva and Robert Östling. 2017. Cross-lingual learning of semantic textual similarity with multilingual word representations. In *21st Nordic Conference on Computational Linguistics, NoDaLiDa, Gothenburg, Sweden, 22-24 May 2017*. Linköping University Electronic Press, 211–215.

[6] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[7] Wei Di, Anurag Bhardwaj, Vignesh Jagadeesh, Robinson Piramuthu, and Elizabeth Churchill. 2014. When relevance is not enough: Promoting visual attractiveness for fashion e-commerce. *arXiv preprint arXiv:1406.3561* (2014).

[8] Zhicheng Dou, Ruihua Song, Xiaojie Yuan, and Ji-Rong Wen. 2008. Are click-through data adequate for learning web search rankings?. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 73–82.

[9] Huizhong Duan, ChengXiang Zhai, Jinxing Cheng, and Abhishek Gattani. 2013. Supporting keyword search in product database: a probabilistic approach. *Proceedings of the VLDB Endowment* 6, 14 (2013), 1786–1797.

[10] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11, Feb (2010), 625–660.

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1126–1135.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*. 2042–2050.

[14] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2333–2338.

[15] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. 2017. On application of learning to rank for e-commerce search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 475–484.

[16] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Lerning Representations (ICLR)* (2015).

[17] Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226* (2018).

[18] Beibei Li, Anindya Ghose, and Panagiotis G Ipeirotis. 2011. Towards a theory model for product search. In *Proceedings of the 20th international conference on World wide web*. ACM, 327–336.

[19] Maggie Yundi Li, Stanley Kok, and Liling Tan. 2018. Don't Classify, Translate: Multi-Level E-Commerce Product Categorization Via Machine Translation. *arXiv preprint arXiv:1812.05774* (2018).

[20] Bo Long, Jiang Bian, Anlei Dong, and Yi Chang. 2012. Enhancing product search by best-selling prediction in e-commerce. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2479–2482.

[21] Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[22] Alessandro Magnani, Feng Liu, Min Xie, and Somnath Banerjee. 2019. Neural Product Retrieval at Walmart. com. In *Companion Proceedings of The 2019 World Wide Web Conference*. ACM, 367–372.

[23] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[24] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2249–2255.

[25] Nish Parikh and Neel Sundaresan. 2011. Beyond relevance in marketplace search. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2109–2112.

[26] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[27] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[28] Shota Sasaki, Shuo Sun, Shigehiko Schamoni, Kevin Duh, and Kentaro Inui. 2018. Cross-lingual learning-to-rank with shared representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 458–463.

[29] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 373–374.

[30] 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.

[31] 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.

[32] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[33] Liang Wu, Diane Hu, Liangjie Hong, and Huan Liu. 2018. Turning clicks into purchases: Revenue optimization for product search in e-commerce. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 365–374.

[34] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics* 4 (2016), 259–272.

[35] Ming Zhu, Aman Ahuja, Wei Wei, and Chandan K Reddy. 2019. A Hierarchical Attention Retrieval Model for Healthcare Question Answering. In *The World Wide Web Conference*. ACM, 2472–2482.