Attention Gets You the Right Size and Fit in Fashion

KARL HAJJAR, Zalando SE

JULIA LASSERRE, Zalando SE

ALEX ZHAO[*], UC Berkeley

REZA SHIRVANY, Zalando SE

Avoiding returns in e-commerce platforms has become a critical issue in terms of both increasing customer satisfaction and decreasing carbon footprint. In the online fashion industry a very large part of the returns is due to size and fit issues that arise from the underlying complexities of shoe and garment manufacturing combined with subjective preferences of customers towards what fits them best. In this context, size recommendation systems capable of estimating a customer's size in thousands of available brands and categories ahead of purchase time are deemed invaluable in dramatically reducing the number of returns related to size and fit. We present a flexible and scalable size recommendation approach that overcomes some limitations of current state-of-the-art work by building upon recent advances in natural language processing and casting the size recommendation problem as a kind of "translation" problem (from articles to sizes) using an attention-based deep learning model for size and fit prediction. Through extensive experimental results, over millions of customers and articles, we demonstrate how this approach is capable of dealing with multiple customers buying from a single account, leveraging cross-category and temporal information to make better predictions, and providing explanations on the final size predictions it produces, thereby helping reduce the potential emotional costs of such predictions for customers.

CCS Concepts: • **Information systems → Data mining**; *Recommender systems*; • **Applied computing →** *Online shopping*.

Additional Key Words and Phrases: size and fit recommendation, fashion, e-commerce, attention mechanisms

ACM Reference Format:
Karl Hajjar, Julia Lasserre, Alex Zhao, and Reza Shirvany. 2020. Attention Gets You the Right Size and Fit in Fashion. In *Proceedings of RecSys '20 Workshops: ACM Conference on Recommender Systems (RecSys '20 -easy- fashionXrecsys '20).* ACM, New York, NY, USA, 15 pages.

1 INTRODUCTION

When shopping for fashion online, customers need to purchase garments and shoes without trying them on to see and feel how they fit. This leads to a great deal of uncertainty in the buying process and to the hurdle of returning articles that customers love but do not fit. Thus, many customers either have to return several purchased articles or remain reluctant to engage in the purchase process altogether. Being able to accurately predict sizes can therefore significantly contribute to increasing customer satisfaction and business profitability through reducing returns, which also reduces the carbon footprint of fashion e-commerce platforms. As an increasing number of people use online fashion stores to shop for articles, these platforms try to support their customers better by providing size information and advice

[*] Work done while interning at Zalando SE

in a passive or active form such as: (1) Size tables and aggregated article measurements [1] provided per brand and article category - this approach requires customers to measure different parts of their body and determine the right size themselves for desired articles by cross referencing their measurements with the size tables; (2) Customer-engaging questionnaires, dialogue-like mechanisms or processing textual feedback [2–4] - customers are asked to provide various explicit personal data such as age, height, weight, tummy shape, hips form, body type, favorite brand, usual sizes, fit preference, etc. to receive a size recommendation; (3) Computer vision and 3D approaches [3, 5–7] providing virtual fit-like solutions based on recent progress in 3D human body estimation [8, 9] - customers are required to provide personal information (as in approach 2.) and/or to take one or multiple pictures of their bodies in tight fitting clothes so a simulated avatar of their body can be built and their measurements predicted to recommend a size; (4) Approaches that leverage existing customers' purchase histories for size recommendation [10–17] - customers with a purchase history automatically receive a size recommendation without any need for providing explicit personal data or images.

All the aforementioned approaches have their own advantages and limitations and the comparison of these radically different methods in tackling the size and fit problem remains out of the scope of this work. In this work we focus on the fourth category of approaches where customers are not required to actively engage in the solution and the size recommender system leverages customers' existing purchase history to provide size recommendations at scale for millions of customers and thousands of brands and articles. Although recommending personalized articles to customers has a long history within machine learning and recommender systems, using methods that can automatically learn from data for size and fit recommendation has only recently received attention [10–17]. What is more, the problem of predicting the right size based on previous purchases is very challenging as: (a) Purchase data is very sparse- a customer only buys a tiny fraction of all the possible articles and sizes that exist; (b) It is also very noisy- a customer can buy various articles for multiple friends and family members in close and neighbouring sizes to their own; (c) The right size for a customer is very subjective- two customers with the exact same purchase history might still buy two different sizes for the same new article based on their perception of the right size; (d) Customers may have a high degree of emotional engagement with the sizing topic - even an accurate size recommendation can come with a high emotional cost for the customer when the recommended size differs from their own expectation.

In this work, we draw inspiration from recent successes of attention-based models in Natural Language Processing (NLP) [18–20] to bring forward a flexible and scalable approach to size and fit recommendation that overcomes the limitations of current state-of-the-art solutions. We propose to model the size prediction problem as an unconventional many-to-one "translation" problem, "translating" from an article to a size given a source sequence of articles (the context). Within this formulation, we take the input source sequence to be a customer's previous purchase history, consisting of all the articles purchased so far, along with the corresponding timestamps and sizes. Then, at a given time, a "query" article is provided to the model and it has to predict (or decode) the correct translation, that is, the right size of this query article for that particular input sequence of articles (which defines the customer in question).
Contributions: The aim of this work is not only to show that the proposed architecture surpasses state-of-the-art performance, but also to highlight the potential and flexibility of a well designed attention-based model in the size and fit problem space. The contributions of our work are as follows:

(1) We demonstrate, for the first time to the best of our knowledge, the value of attention-based approaches in tackling existing challenges of personalized size and fit recommendation in fashion e-commerce. We show such approaches are capable of efficiently leveraging scarce, subjective and noisy purchase data to provide accurate size recommendations.
(2) Our proposed approach overcomes several major limitations of current state-of-the-art approaches. It is trained once for all categories altogether and can digest new data online without having to be fine-tuned for new customers

or articles. It takes advantage of the contextual aspect of the problem to leverage cross-category correlations, which is necessary when recommending a size for categories that a customer has never bought before.

(3) We show how explicitly paying different amounts of attention to each previous purchase not only enables predicting sizes more accurately than state-of-the-art size recommendation methods but also enables customers to gain valuable insights as to why a particular prediction has been made for them (in single as well as multi-user behind an account scenarios), thereby moving towards reducing the emotional cost of unexpected size recommendations.

(4) We demonstrate how the adaptability of our approach leads to accuracy improvements on the difficult cold-start problem (new fashion category / user in an account / customer) and low number of previous purchases regime.

The outline of the paper is as follows. We present related work in section 2 and our approach in section 3. In section 4, we provide extensive details on the data and the experimental setup used to build a comprehensive set of experiments. In section 5 we present our results alongside both naive and state-of-the-art baselines, and discuss results on public datasets in section 6. We finally draw conclusions and lay out potential future work in section 7.

2 RELATED WORK

There has been growing literature about size and fit recommendation in recent years. In previous work, a size is always predicted by combining (through a dot product or a concatenation for example) a customer representation with an article representation. Those works can be split into two categories arising from a major conceptual difference: 1) Those which reduce the customer representation to a single vector (either by design, by averaging over past purchases, or by using Gaussian modeling, etc.) [10–16], and 2) Those which build flexible customer representations by considering a list of vector representations of a customer's past purchases [17]. Our work belongs to the latter category.

In the first category, a major series of work focuses on predicting if an article in a given size will be either fit, small or large for a given customer using their past purchase history. [10–13] all suggest estimating the "fitness" (Small, Fit, Large) between a customer representation and the representation of an article in a given size. While [10, 13] use latent factor models coupled with ordinal regression or metric learning, [11, 12] use Bayesian models with the difference that [12] uses a hierarchical structure and allows directly predicting the probability of any size given a customer-article pair, conditionally on the article being kept (good fit) in contrast with [10, 11, 13] which have to predict the fit (good fit, too big, too small) for all possible sizes one by one. All these works require a numerical mapping of sizes to be applied to other fashion categories than shoes. Finding such a mapping can be difficult and is in itself a research topic [21].

In another line of work, [14] and the Product Size Embedding (PSE) model [15] learn article embeddings for each article-size combination and customer embeddings which are then combined to predict which size of an article would be most likely kept by a customer. [14] pre-trains a skip-gram based Word2Vec [22] per fashion category to learn article embeddings whereas [15] learns them end-to-end, and both get a customer embedding by averaging their purchased article embeddings. In [14], the customer embedding and article embedding are concatenated along with additional article and customer features, and a Gradient Boosted Classifier [23] is trained to classify whether a customer will keep a specific size of an article, whereas in [15] inner products between the customer embedding and the embedding of an article in all possible sizes are computed to obtain scores that are then normalized into probabilities using a softmax.

More recently, a Siamese-like neural network architecture SFNet was introduced in [16] that is able to leverage cross-category correlations. The neural network used first encodes separately the customer and the article, then concatenates the two embeddings and feeds them to fully connected layers before predicting the size. The meta-learning approach MetalSF [17] also learns article embeddings along with size embeddings using fully connected layers, combined with a linear projection to map an article from the latent article space to the latent size space. This mapping is learned using linear regression on the customer's embedded past purchases, and a size is predicted by feeding the projected latent size representation of a new article to fully connected layers which output a distribution over sizes after a softmax.

One category, one model: All aforementioned approaches [10–12, 14, 15], except SFNet [16] and MetalSF [17], suffer from separate model training requirements (one model per fashion category) and expert size mapping development for each category. Aside from the computation costs, having to train multiple models separately dramatically reduces the amount of information available as input to each model (since to predict the size for an article in a given fashion category, only the articles from that same category can be considered). By restricting the number of articles considered in the purchase history of a customer to single categories, such approaches forbid leveraging cross-category information that is potentially useful for predicting the right size for a new article.

Multi-user accounts: The aforementioned methods have different strategies for explicitly dealing with multi-user accounts. Some use manually set thresholds on the range of the sizes purchased by a customer [14] while others use more advanced methods such as Gaussian mixture models [15], Dirichlet processes [12] or hierarchical clustering [10]. In contrast, SFNet [16] relies entirely on the customer embedding to somehow incorporate the information from different accounts during training and later use it at test time. MetalSF [17] does it implicitly through the encoding of gender and category.

Unlike [10–12, 14, 15], the model we present in this work does not have to be trained for different gender-category pairs separately. When given a query article for which we wish to predict the size, we let the model decide - using an attention mechanism - which articles in the previous purchase history are relevant to make a size prediction for that specific query article. This allows us to handle implicitly multi-user accounts and cross-category purchase histories within the predictive model without any additional work to identify multiple users, regroup articles per fashion category, or map sizes to numerical values.

In the remainder of this section we take a closer look at [16] and [17], as our model is similarly trained once for all categories and genders, can effectively utilize cross-category information, and uses deep learning to learn article and customer representations, but yet still bears significant conceptual and architectural differences with those works. The proposed approach differentiates notably from [16] and [17] in that it relies heavily on attention mechanisms [18] (self-attention and source attention) which enables very different types of abstractions compared to that of the fully connected layers [16, 17] and the linear regression used in [17]. A major difference between our model and MetalSF [17] on the one hand, and SFNet [16] on the other hand, is that even though in [16] the sequence of past purchases of a customer is taken into account in some part of the model (when learning customer embeddings), SFNet does not have direct access to this sequence when predicting the size for a new article (no direct comparison to other articles of the purchase history is possible), thereby relying solely on the aggregated information and losing the specific information about each individual article purchased by a customer. This distinction is analogous to the one which originally led to the introduction of attention mechanisms [24] in the decoding part of neural machine translation systems using recurrent neural networks, inducing major improvements in machine translation. In the proposed attention-based model, all past purchases of a customer are given as input context when predicting the size for a new article, irrespective of the articles' genders or categories. The model uses this information flexibly to provide predictions that always depend on the context of previous purchases and not on a single embedding that summarizes the whole purchase history of a customer as in [14–16]. This also implies that, in contrast to SFNet [16], our model can ingest any additional purchase at prediction time without any fine-tuning.

We further outline major leaps in the proposed approach compared to the recent MetalSF [17]. In [17], a customer is represented by their history of past purchases, which are embedded *independently* of the context of the other purchases.

In contrast, the self-attention mechanism we employ embeds past purchases mutually based on the context of the other purchases in the history of a customer. Additionally, in [17], the order of purchased sizes in time is not leveraged by the model as the past purchases of a customer are defined as a set. In contrast, the attention-based architecture we put forward is inherently designed for sequential problems (e.g. NLP tasks), allowing us to consider the purchases as an ordered sequence and thus to leverage important information on the evolution of customers' size and fit preference over time. Finally, in [17], the size representation of the query article is obtained through a linear combination of the (context-independent) size representations of the previous purchases with weights that can take any value, whereas our model uses a convex combination (linear combination with positive weights which sum to one). In [17], the weights of the linear combination are obtained by solving a linear regression problem from the representations of articles and sizes, whereas the weights of the convex combination in our model are obtained through an attention mechanism computed on article representations (involving multiple projections and non-linearities), which can yield more powerful representations. In the remainder of this work we demonstrate how the aforementioned critical characteristics of our approach play a strong role in advancing the state-of-the-art in a diverse set of real-world scenarios.

3 PROPOSED APPROACH

We build on recent advances in attention models and adapt [18] to the problem of size recommendation. A sentence is the sequence of past purchases of a customer C, referred to as **support purchases**, and is annotated with the associated sizes. When C is faced with a new article, referred to as **query article**, their support purchases are used to infer the size they should purchase this new article in. More formally, for a given sample, the sequence of support purchases of a customer C is denoted by $(p_1, p_2, \ldots, p_n)$, where n depends on the sample, and where a purchase $p_i = (T_i, A_i, S_i)$ consists of the timestamp T_i, the article A_i and the size S_i purchased by C. The query article q_{n+1} is defined by (T_{n+1}, A_{n+1}) and consists of time and article information. Note that in practice, many different samples of support / query pairs can be constructed from the list of all purchases of a customer C. Time information is included as input to enable the model to learn how a customer's size and fit preferences or body shape might evolve over time.

The architecture of our model is sketched in Figure 1. The 3 components are an an encoder, a decoder and a size predictor. (1) The encoder takes as inputs the support purchases of a customer, defined as a sequence of processed vectors including article, time and size information, as described in Section 3.1. Each support purchase is encoded into a numerical vector using self-attention to leverage the context of the other purchases. (2) The decoder takes as inputs the sequence of encoded support purchases and a query article, the size of which should be inferred. The query article is a processed vector including article and time information as described in Section 3.1. The decoder uses source-attention to leverage the relevant information contained in the encoded support purchases and encodes the query article. (3) The size predictor applies a linear transformation followed by a softmax operation to convert the encoded query article into size probabilities. In practice, the set of available sizes for a particular article is a small subset of all the sizes known to the model and a size mask is used to normalize the size probabilities appropriately (all the mass is given to the sizes available for a given article). The size actually purchased by the customer for that query article is used as **target size** to compute a categorical cross-entropy loss.

In this paper, a query article is an article actually purchased (and kept) by C, for which we know the timestamp of the purchase and the target size. In contrast, in a real production setting, the query article would be a new article that a customer would be willing to purchase and the timestamp would be the current date. The correct size of this query article for that customer would then be unknown and the model would be asked to produce a recommendation.

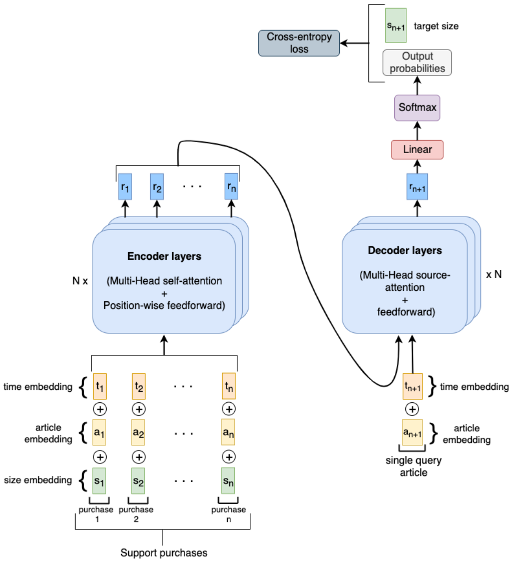


Fig. 1. Attention-based model architecture

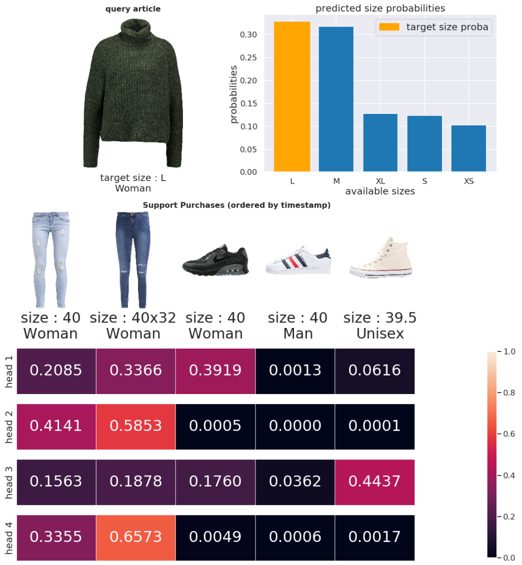


Fig. 2. Category cold-start sample

3.1 Inputs and Embeddings

This section describes how the raw data is transformed so it can be processed by the encoder and the decoder.

Timestamps. The integer representation T_i of timestamps is given by the number of days between the timestamp itself and a fixed reference date. Note that the reference date is arbitrarily set to a date prior to any purchase in the dataset, but its actual value has no impact on the performance.

Articles. Articles A_i are defined by a set of categorical attributes, namely high-level category (textile, shoes, sports, etc.), gender (men, women, unisex), fashion category (jeans, sweaters, sneakers, etc.), brand, season and supplier. Each attribute is one-hot encoded and the resulting vectors are concatenated to produce the article representation of dimension $4,621$.

Sizes. Sizes S_i are one-hot encoded into a $1,162$ dimensional vector- the list of sizes includes all the different size systems present in the dataset: numeric (38, 40), standard (S, M), fractions (41 1/3, 42.5), confection (36-38, 40-42), etc.

Article, timestamp and size are each embedded independently into a numerical vector. For size and article, simple embedding matrices are used to convert the vector representations into vector embeddings $a_i = W_a A_i$ and $s_i = W_s S_i$. For time, we use the positional encoding made of sines and cosines described in [18], where we have replaced the index of the position of an element in the sequence by the time representation (difference in days to a reference date), clamped to a maximum value $M = 1825$ ($\simeq$ 5 years), so that $t_i = f_{\text{pos}}(T_i)$. Note that we have used capital letters for the raw representations and lower case letters for the embeddings. All embeddings are of dimension $d = 256$. The matrices W_a and W_s are learned as part of the model's parameters and are, along with the positional encoding f_{pos}, **shared** across support purchases and query articles. As shown in Figure 1, the representation of a support purchase is the sum of the embeddings for timestamp, article and size. The representation of a query is the sum of the timestamp and article embeddings only.

Keeping the size embeddings separate allows us to map all sizes to a common continuous latent space, the analysis of which is left for future work. Additionally, it allows us to tie the weight of the linear transformation of the size predictor to the size embedding weights, which was shown in [25] to improve performance on language models. Summing up the embeddings is a design choice, concatenating them led to comparable performance. Note that these embeddings are based on generic features such as brand, category, time and not on hashes, so the model can process new articles without fine-tuning. Similarly, a customer is represented by a list of purchases and not by a hash, so new customers can be directly handled.

3.2 Encoder and Decoder Layers

Both the encoder and decoder use $N = 2$ identical layers, with N distinct sets of parameters, stacked on top of one another. Each layer has 2 blocks: for the encoder, 1. a multi-head self-attention block with h=4 heads, 2. a position-wise feed-forward block; for the decoder: 1. a multi-head source-attention block with h=4 heads, 2. a standard feed-forward block (the position is not needed as there is a single query article). The source-attention weights are computed using the encoder representations of the support purchases as both key and values following the scaled dot-product attention used in [18]. The feed-forward blocks are composed of a 2-layer neural network with a hidden layer of dimension $d_{ff} = 512$ and GELUs activation [26], as used in [19]. All blocks have output dimension d and are followed by a residual connection [27] and layer normalization [28]. The depth shown in Figure 1 shows the number of attention heads within a layer, not the N layers.

Let us denote, for each support purchase p_i, $x_i = t_i + a_i + s_i$ the sum of the three embeddings. Then, denoting $f^{\text{enc}}_{\theta_e}$ the encoder function with parameters θ_e, the encoder takes inputs $(x_1, \ldots, x_n)$ and produces the sequence of n contextual representations of the support purchases $(r_1, \ldots, r_n)$:

$$(r_1, \ldots, r_n) = f^{\text{enc}}_{\theta_e}(x_1, \ldots, x_n) \tag{1}$$

We highlight that our model encodes each support purchase using the context of the other associated support purchases, in contrast to other work where support purchases are unaware of one another and have context-free embeddings.

The output of the decoder is a d-dimensional contextual vector representation of the query article computed using the final source-attention weights (one per support purchase) in the decoder. More formally, denoting by $y_{n+1} = t_{n+1} + a_{n+1}$ the sum of the time and article embeddings of the query article, and by $f^{\text{dec}}_{\theta_d}$ the decoder function with parameters θ_d, the contextual representation r_{n+1} of the query article is given by :

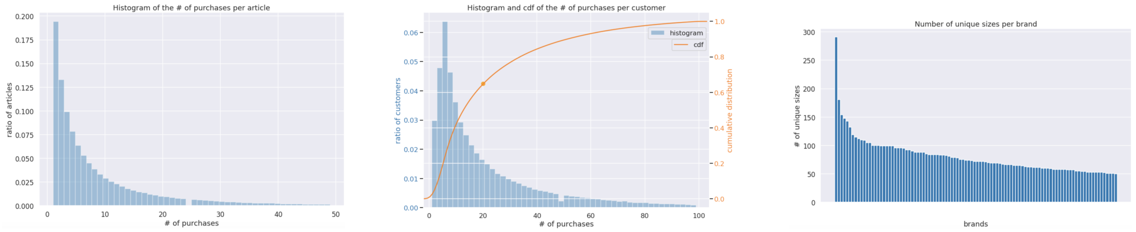
$$r_{n+1} = f^{\text{dec}}_{\theta_d}(r_1, \ldots, r_n,\ y_{n+1}) \tag{2}$$

The representation r_{n+1} is linearly transformed into a vector of logits and a softmax gives the probability distribution over sizes. Following [25], the linear transformation has no bias term and is tied to the size embedding matrix. In other words, it is given by $W_s^T r_{n+1}$ where W_s is the size embedding matrix of shape $d \times n_{\text{sizes}}$ described in subsection 3.1.

As shown at the top of Figure 1, a categorical cross-entropy loss is then computed between the probability distribution over sizes output by the model, and the target size actually purchased by the customer. As mentioned in section 3.1, in practice only few sizes among all possible sizes are available for a particular article. Irrelevant sizes are thus masked out by setting the associated logits to $-\infty$. Following [15–17], masking is done during training **and** at test time.

4 EXPERIMENTAL SETUP

This section presents in detail the data, the training pipeline, and the hyperparameters used for the model.



(a) Purchases per article

(b) Purchases per customer

(c) Unique sizes per brand

Fig. 3. Distribution of brands and purchases per customer and article. Left (a): Histogram of the number of purchases per article for articles with fewer than 50 purchases. Middle (b): Histogram and cumulative distribution function (cdf) of the number of purchases per customer for customers with fewer than 100 purchases. Right (c): Number of available unique sizes per brand for the 100 brands with the highest size diversity. Each vertical bar represents one brand.

4.1 Large-scale anonymized Data

We consider purchase data between 2015 and 2019 from a major fashion e-commerce platform for one European country. Only purchases kept by the customers are considered, the integration of return data is left for future work. This leaves **9M** purchases, **380k** unique customers, **770k** unique articles, **2.2k** different brands and **1,162** unique sizes. This dataset is anonymized and not made public due to various customer privacy challenges and proprietary reasons which lie outside the scope of this work. However the important aspects related to the sizing problem at hand are studied to provide a better understanding of the data. Figure 3b (*resp.* Figure 3a) shows the distribution of the number of purchases per customer (*resp.* per article) in the dataset, and the corresponding cumulative distribution function. We plotted the distribution only for customers with fewer than 100 purchases, and articles with fewer than 50 purchases for readability. This represents roughly 97% of all customers and articles in the dataset. We observe that a large majority of articles were purchased less than 5 times, and that more than 60% of the customers have fewer than 20 purchases in their full history of purchases (orange dot on the cumulative distribution function). In those 20 purchases, articles from both genders can be present, and we often have only a few gender-category pairs represented. This means that, for systems which are not able to predict sizes using information from other categories, it would be impossible to serve recommendations in all fashion categories for many customers, or it would take a high number of purchases for a single customer before they can predict a size for most categories, limitations which our model does not suffer from.

Figure 3c shows the number of available unique sizes per brand for the 100 brands with the highest size diversity in the dataset. We observe that nearly all the brands presented provide more than 50 unique sizes for the customers to select from, thereby naturally creating a high degree of uncertainty for the customer regarding which size to purchase even within one brand. On the other hand, even the brand with the highest size diversity only offers a fraction ($\sim$ 25%) of the $1,162$ available unique sizes in the dataset which adds a great deal of complexity for inter-brand size recommendations since different brands might offer different types of sizes for the same fashion category (e.g. numeric vs fractional for shoes).

4.2 Training, validation and test samples

Train / validation / test split. In order to stay as close as possible to a production setting we follow the same split as in [17]. Purchases are ordered by increasing timestamp and the first 80%, denoted $\mathcal{P}_{\text{train}}$, are used for training, the next 10% for validation ($\mathcal{P}_{\text{val}}$) and the remaining 10% for test ($\mathcal{P}_{\text{test}}$).

Training samples. Purchases in $\mathcal{P}_{\text{train}}$ are grouped by customer, giving one sample per customer. Customers with one purchase only are removed and samples are augmented using the procedure below. For each resulting sample, the last purchase provides the query article and its target size, while all other purchases are used as support.

Data augmentation. As noted in Table 1 in column Train (augment.), the training data (only) is augmented similarly to [15]. First we split each customer's purchase history into sequences of maximum length $L = 40$, and consider each sub-sequence as an independent sample. We then re-split each of those samples $n_{\text{splits}} = 5$ times. To achieve that, for a given sample with n purchases, we select uniformly at random n_{splits} integers $k_1, \dots, k_{n_{\text{splits}}}$ between 2 and n, and then for each of those integers create a new sample by keeping only the first k_i purchases. We keep the original training samples we had before re-splitting and append all the new samples to get the final set of training samples.

Validation samples. Each purchase in $\mathcal{P}_{\text{val}}$ is a query article. The associated support purchases are made of all the purchases in $\mathcal{P}_{\text{train}}$ bought by the same customer.

Test samples. Each purchase in $\mathcal{P}_{\text{test}}$ is a query article q coming from customer c. The associated support purchases are made of subsets of previous purchases from c. Following [17] these subsets vary depending on the chosen scenario.

1. Offline test scenario. Standard test scenario where the support purchases are all of c's purchases in $\mathcal{P}_{\text{train}}$.

2. Online test scenario. The online scenario simulates a real life production environment where information about a customer is not fixed in time and can be updated with each of their new purchase. Here, the support purchases are all of c's purchases in $\mathcal{P}_{\text{train}}$ plus all of c's purchases in $\mathcal{P}_{\text{test}}$ that were made prior to q's timestamp. The first query article associated with c will have only support purchases from the training set, the second query will have a support augmented with the first query article and so on.

3. Offline (+val) / 4. Online (+val) test scenarios. These scenarios mirror scenarios 1 and 2 respectively using $\mathcal{P}_{\text{train}} \cup \mathcal{P}_{\text{val}}$ instead of $\mathcal{P}_{\text{train}}$.

5. Cold-start online test scenario. In this last test scenario, we consider customers who only have purchases in $\mathcal{P}_{\text{test}}$ and not in $\mathcal{P}_{\text{train}}$. It mirrors scenario 2 using the empty set $\emptyset$ instead of $\mathcal{P}_{\text{train}}$. That is we start with an empty support sequence for each customer, and update the supports progressively with each new purchase in $\mathcal{P}_{\text{test}}$.

The performance in all of these 5 different test scenarios is summarised in Table 5. Note that the number of samples is the same for the first 4 test scenarios, and that a sample for a given customer has the same query article and target size in all scenarios, only the composition of the support purchases of the sample differs across scenarios. The number of training, validation and test samples are detailed in Table 1a. The cold-start online scenario on the other hand only has 99k samples as customers who have purchases in $\mathcal{P}_{\text{train}}$ are excluded for this scenario.

4.3 Experimental details

We detail below different parts of the experimental setup we used to produce the results described in section 5. The code for our Transformer architecture was inspired by the PyTorch code of the Harvard Annotated Transformer [29].

Regularization. We use regularization as in the standard Transformer architecture: additionally to layer normalization, we apply dropout [30] with a rate $p_{\text{drop}} = 0.3$ and label smoothing with a smoothing factor $\varepsilon_{ls} = 0.4$. For a concise description of label smoothing, we refer the reader to Section 1.1 of [31]. Smoothing is only applied to the reduced set of available sizes for a particular article, and not to the the set of all possible sizes, allowing us to use a higher value for the smoothing factor. Dropout is applied to the sum of the embeddings before feeding the inputs to the model in the encoder and decoder, and to the output of each sub-block of every layer in both the encoder and the decoder, before the residual connection and the layer normalization.

Optimizer. We use the Adam optimizer [32] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-6}$, and vary the learning rate η_k in function of the iteration k just as in [18], using $n_{\mathrm{w}} = 5,000$ warmup steps : $\eta_k = d^{-0.5} \min(n_{\mathrm{w}}^{-1.5}\, k,\ 1/\sqrt{k})$

Hyperparameters. Table 1b summarizes the values of the model and training hyperparameters. When batching, we pad all support purchases with zeros up to a length of L if needed.

Table 1. Left (a): Number of samples for each split. Right (b): Model and training hyperparameters.

	Train	Train (augment.)	Val	Test
# samples	334,170	2,168,017	784,321	815,405

	N	h	d	d_{ff}	L	M	n_{splits}	batch size
values	2	4	256	512	40	1825	5	256

Hardware and training time. With the values for the hyperparameters described above, we obtain a model with ∼ 3.7 million parameters. Training this model on the ∼ 2 million augmented training samples on a single NVIDIA Tesla V100 GPU took a little less than 3 days for a total of ∼ 120K training steps (15 epochs).

5 RESULTS AND DISCUSSION

In this section, we present and discuss the results of different experiments and analyze them following a similar comparative analysis as in [17] to evaluate how our model performs against different criteria, and then draw conclusions about its advantages compared to other work.

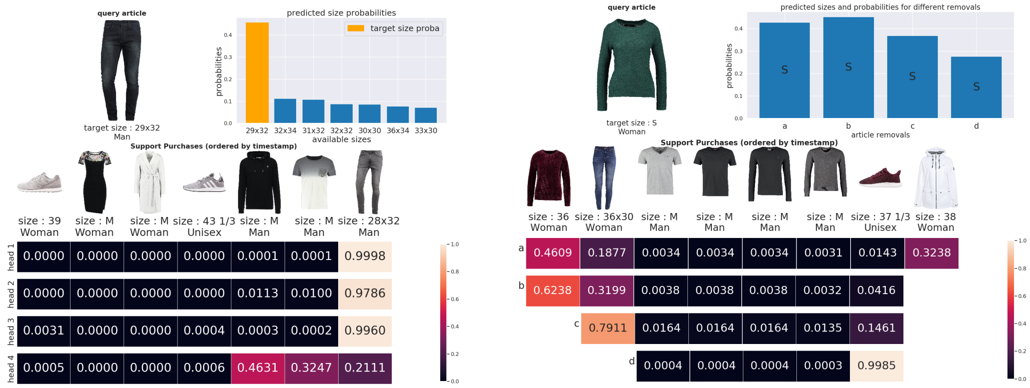
5.1 Overall performance comparison

In this section, we evaluate the model's performance in the offline scenario (defined in subsection 4.1), which we take as our standard performance comparison scenario, compared to the two simple baselines used in [16] and three state-of-the-art methods [12, 15–17], namely: 1. the popularity baseline (for an article, independently of the customer, the most purchased size for that article is predicted), 2.the Bayesian model presented in [12], 3. the Product Size Embedding (PSE) model [15], 4. the Size and Fit Network (SFNet) [16] and 5. MetalSF [17].

The KDE, Bayesian and PSE methods are trained separately for each category-gender pair. We consider 3 distinct fashion categories : lower-garments, upper-garments and shoes, and two different article genders: male and female. This results in 6 different models for each of these methods. Table 2 shows log-likelihood, top-1-2-3 accuracies and micro-averaged AUC for all approaches on more than 815k test purchases from all categories in the offline scenario. We use the same masking of non-available sizes for **all** methods at test time. The attention-based model performs best with a relatively large improvement compared to [12, 15, 16], and a marginal improvement compared to [17]. We show however in sections 5.2.2, 5.4, and 5.5 that the difference in performance increases on the most difficult scenarios. Note that despite the fact that the attention-based model has higher top-1-2-3 accuracies than the other models, it still has lower log-likelihood and micro-AUC than MetalSF [17]. This shows that even though the model predicts sizes more accurately than other models, it is not overly confident in its predictions, which is a typical pitfall of deep learning approaches. This is most likely due to the label smoothing employed in training, which prevents the model from putting too much probability on one single size, making it more robust.

Table 2. Comparison on all categories and size systems in the offline scenario. "log lik." stands for log likelihood, "top-*k*" is top-*k* accuracy and "mAUC" is micro-averaged AUC.

	log lik.	top-1	top-2	top-3	mAUC
popularity	-2.01	0.29	0.53	0.68	0.69
Bayesian [12]	-1.46	0.47	0.72	0.84	0.79
PSE [15]	-1.47	0.53	0.77	0.87	0.82
SFnet [16]	-1.20	0.55	0.79	0.89	0.85
MetalSF [17]	**-1.04**	0.60	0.83	0.92	**0.89**
attention-based	-1.11	**0.61**	**0.84**	**0.93**	0.88



(a) cross-category sample

(b) Removing different articles from the support.

Fig. 4. Size predictions and attention weights. Left (a): on a standard cross-category sample. Right (b): when removing different articles from the support.

5.2 Cross-category performance

One advantage of our model, compared with PSE [15], is that it is able to leverage cross-category information to predict sizes. In addition, we show here that our approach performs significantly better than SFNet [16] and MetalSF [17] in exploiting and explaining these cross-category correlations. This advantage can enhance standard size recommendation by using information from other fashion categories than that of the query article, but more importantly can help with the difficult problem of cold-start category recommendation which [12] and [15] cannot tackle. We show below how our attention-based approach is able to deal with both those settings.

5.2.1 ***Standard cross-category recommendation.*** We start by showing in Figure 4a an example where the model attends to different categories to make a prediction in the context where the category of the query article (men's jeans) is also part of the support purchases (rightmost article in the support). Figure 4a is composed of the following. **Top row**: (*left*) query article with its gender and target size, (*right*) model's output probabilities (only the sizes with a probability higher than 10^{-3} among the top 10 sizes are displayed for readability). **Second row**: support articles with their genders and sizes. **Four bottom rows**: weights of each attention head. We observe that the model overall attends mostly to the men's jeans article present in the support, but also pays attention to the men's sweatshirt (3rd article from the right) and the men's t-shirt (2nd article from the right), thereby showing that it does use information from other fashion categories to predict a size.

5.2.2 ***Category cold-start performance.*** To quantify the advantage of our approach, we focus here on the category cold-start recommendation problem for upper and lower garments. We show that customers who have never bought in one of these categories can be given a better prediction than with other methods if they have shopped in another category. We compare the performance of our method with the popularity baseline (which the Bayesian, KDE and PSE methods would return since they cannot deal with category cold-start recommendation), SFNet [16] and MetalSF [17], and build two datasets (a and b) which are subsets of the test set used in the standard offline scenario described in subsection 4.1. For dataset a (*resp.* b) we keep the samples from the offline scenario for which the query article is an upper-garment (*resp.* lower-garment) and the corresponding customer has no upper-garment (*resp.* lower-garment) purchase in $\mathcal{P}_{\text{train}} \cup \mathcal{P}_{\text{val}}$. Results are reported in Table 3a for upper garments and Table 3b for lower garments. The results show that our approach can leverage cross-category information to predict sizes much more accurately than the popularity baseline, SFNet [16] and MetalSF [17]. An example where our model uses information from other categories to predict a size in the cold-start category setting is shown in Figure 2, which is composed the same way as Figure 4a.

Table 3. Models comparison on a category of interest for customers who have not bought that category in the training or validation sets.

(a) Upper garments (13k test samples).

	log lik.	top-1	top-2	top-3	mAUC
popularity	-1.82	0.31	0.60	0.75	0.64
SFnet [16]	-1.43	0.37	0.62	0.77	0.67
MetalSF [17]	-1.30	0.41	0.69	0.86	0.73
attention-based	-1.60	0.45	0.73	0.89	0.75

(b) Lower garments (15k test samples).

	log lik.	top-1	top-2	top-3	mAUC
popularity	-2.54	0.24	0.45	0.60	0.71
SFnet [16]	-1.79	0.35	0.57	0.71	0.75
MetalSF [17]	-1.60	0.38	0.61	0.76	0.80
attention-based	-1.30	0.40	0.64	0.78	0.81

5.3 Attention adapts to changes in the history

We show here how our model adapts its attention when the context (the support purchases) is modified. In Figure 4b we take an initial purchase history and remove articles to see how the model's attention weights shift. Figure 4b has the same composition as Figure 4a, except for the following differences. **Top row**: (*right*) predicted sizes for different article removals. **Four bottom rows**: averaged attention weights when removing articles from the support. For each removal, the weights of all 4 heads are averaged at each position in the support to reduce figure size. We remove the following articles from the support one by one (in this order top-down): a. No removal, full set of support purchases, b. the white jacket (rightmost article at index 7), c. the burgundy jumper (leftmost article at index 0), and d. the pair of jeans (at index 1). We observe that more cross-category attention is needed when removing the jumper (leftmost article at index 0) from the support, which is from the same fashion category as the query article. It is also worth noting that when removing the jacket, the model becomes more sure of its prediction. We hypothesize that the jacket might run a bit too small which makes the model slightly more uncertain about the correct size to predict for the query article when it is still part of the support purchases. Other than for this jacket, any article removal from the support makes the model increasingly less sure of its prediction because it has to rely only on cross-category information. It is interesting to observe that, even though the model attends to the jeans with the full set of support purchases, the latter gets roughly twice more attention each time an article is removed from the support. This shows that the model is able to use differently the same piece of information depending on what other type of information is available. Note that the pair of sneakers is actually a unisex article, but from the sizes of the other articles it seems like the model understands that it is an article for the female user behind this account, as it starts paying attention to it when other article are removed, whereas none of the 4 male articles receive any attention even though some are jumpers like the query article.

Table 4. Top-1 accuracy on multi-users accounts (> $5k$ test samples per case)

	Bayesian [12]		PSE [15]		SFnet [16]		MetalSF [17]		**attention-based**	
	men	women	men	women	men	women	men	women	men	women
cold-start (no related history)	0.28	0.28	0.28	0.28	0.32	0.30	0.34	0.30	**0.37**	**0.34**
consistent (always same gender)	0.44	0.46	0.50	0.51	0.52	0.54	0.55	0.58	**0.59**	**0.60**
mixed (various genders in history)	0.44	0.47	0.49	0.54	0.48	0.55	**0.55**	**0.61**	0.54	**0.61**

5.4 Performance on multi-user accounts

To evaluate how well our model is able to deal with multi-user accounts we build 6 different experiments keeping subsets of samples from the standard offline setting. 1. (*resp.* 2.) We keep samples where the support has no men's (*resp.* women's) articles and where the query is a men's (*resp.* women's) article (cold-start with no related purchase history). 3. (*resp.* 4.) We keep samples where the support has only men's (*resp.* women's) articles and the query is a men's (*resp.* women's) article (consistent purchase history with always the same gender). 5. (*resp.* 6.) We keep samples where the support has both men and women articles and the query is a men's (*resp.* women's) article (mixed purchase history with various genders). Figure 4a shows a sample from experiment 5, and Figures 2 and 4b show two samples from experiment 6. In all these samples, we observe that the model correctly attends to past purchases of the same gender as the query article. The results of the experiments are presented in Table 4. For experiments 1 and 2 (no related history), the PSE and Bayesian methods return the baseline prediction. The attention-based approach presents a relatively large improvement over the other approaches [12, 15–17] in most experiments, and the similar performance in the consistent and mixed test cases shows that it is not confused by multiple customers behind an account.

5.5 Online performance

The attention-based model we propose proves very valuable in an online scenario as it can digest new customers and new purchases after being trained **without** having to be fine-tuned or modified in any way. In contrast, the design of SFNet [16] imposes that: (1) the network has to be fine-tuned to be able to make predictions for any new customer, (2) whenever a customer already present in the database buys a new article and keeps it, the network has again to be fine-tuned to add this new piece of information within the customer embedding and then use it for further recommendations. This is a major advantage of our model for scalability in a practical setting where both the number of customers and the number of purchases they make grow rapidly.

To quantify the advantage of being able to efficiently process purchases online, we compare the performance of the same trained model on the 5 test scenarios described in subsection 4.1. The results reported in Table 5a are obtained after having trained the model once and fixed all the weights - only the inputs to the model are modified in an online fashion. They show that, as expected, the model is able to leverage additional input information to make more accurate recommendations. In Table 5b, we compare the performance of our model to that of MetalSF [17] in the online cold-start scenario where customers (never seen during training or validation) start with empty support purchases, which are updated one purchase at a time. In this scenario, nearly 70% of the samples have less than 4 purchases in the support. The popularity baseline is used for empty support purchases. Our approach is more accurate than MetalSFin this low number of purchases regime. This is probably due to the attention mechanism's ability to quickly adapt to new purchases and predict accurately for categories / genders which are not part of the support purchases, as demonstrated in Table 3 and Table 4, and visualized in Figure 4b. We leave the analysis of the behaviour on long purchase histories (> 40) for future work but hypothesize that even if the linear regression in [17] benefits from more data to learn from, it could still be affected by outliers while the attention model could filter those out if needed.

Table 5. Effect of online processing (i.e. updating the customer's support set after each purchase) with attention-based models. (a) Standard online vs offline. (b) Online cold-start performance comparison.

(a) Standard online and offline scenarios.

attention-based	log lik.	top-1	top-2	top-3	mAUC
offline	-1.11	0.61	0.84	0.93	0.88
online	**-1.03**	**0.66**	**0.87**	**0.94**	**0.91**
offline + val.	-1.09	0.63	0.85	0.93	0.89
online + val.	**-1.03**	**0.67**	**0.88**	**0.95**	**0.91**

(b) Online **cold-start** (100k test samples).

	log lik.	top-1	top-2	top-3	mAUC
MetalSF [17]	**-1.23**	0.59	0.79	0.88	0.87
attention-based	-1.34	**0.60**	**0.81**	**0.90**	0.87

6 RESULTS ON PUBLIC DATASETS

There are very limited public datasets available for the problem of size and fit, and those (*e.g.* [4]) mainly focus on leveraging customer metadata for the task at hand to predict "fitness" of an article in a given size. As such, the public datasets introduced in [4] are not directly in the scope of this work as methods evaluated on these use either customer or article hashes or customer metadata such as height, age, weight, body measurements to predict too-small, fit or too-big. This is in contrast to our goal, stated in introduction, of building a model for size prediction based solely on the past purchases of a customer without any need for providing sensitive personal data. However, we consider incorporating additional user metadata within our attention-based approach as a future work avenue, and have thus evaluated the top-1 accuracy, log-likelihood and micro-auc of our method on these datasets. The corresponding results with the comparison to LF-ML [4] and SFNet [16] are shown in Table 6. As in [16], since we do not know the splits used in [4, 16], we used 10 random splits and averaged the results. The performance of our approach is comparable to that of the version of SFNet [16] without any customer nor article embedding, which we refer to as SFNet-ne.

Table 6. Performance comparison on the public datasets ModCloth and RentTheRunway

	Entity embedding		Micro-avg AUC		top-1 accuracy		Log likelihood	
Method/Dataset	user id	article id	ModCloth	RentTheRunWay	ModCloth	RentTheRunWay	ModCloth	RentTheRunWay
LF-ML [4]	✓	✓	0.657	0.719	–	–	–	–
SFNet [16]	✓	✓	0.689 ± 0.005	0.749 ± 0.004	**0.690 ± 0.004**	**0.760 ± 0.004**	**-0.758 ± 0.006**	**-0.610 ± 0.008**
SFNet-ne [16]	×	×	0.638 ± 0.007	0.674 ± 0.003	0.683 ± 0.005	0.739 ± 0.002	-0.806 ± 0.009	-0.698 ± 0.006
attention-based	×	×	**0.818 ± 0.003**	**0.857 ± 0.005**	0.683 ± 0.004	0.728 ± 0.009	-0.850 ± 0.011	-0.779 ± 0.015

7 CONCLUSION

In this work, the use of attention models for tackling the size recommendation problem was shown to address several major challenges of current size recommenders, such as dealing with multiple size systems, cross-categorical and multiple gender recommendations. Additionally, the explainability of the predictions made possible by our approach is a big step towards communicating with customers on the emotionally engaged topic of size recommendations. Our approach surpasses the state-of-the-art in large scale experiments, needs only be trained once for all genders and fashion categories and can easily scale to accommodate new customers and purchases. Future work will focus on studying in depth the embeddings learned by the model in a latent sizing space to extract properties of articles, brands, and customers (from a sizing perspective) as well as on analyzing how integrating pre-trained embeddings learned through another method (*e.g.* pre-training with BERT [19] or with a size and fit specific method) can enhance the system's performance. We will also study how the flexibility of our model allows incorporating additional customer metadata when it is available, otherwise leaving the presented model unchanged when it is not.

REFERENCES

[1] Size charts. https://www.adidas.com.sg/help-topics-size_charts.html. Accessed: 2020-01-13.

[2] Ying Yuan and Jun-Ho Huh. *Cloth Size Coding and Size Recommendation System Applicable for Personal Size Automatic Extraction and Cloth Shopping Mall: MUE/FutureTech 2018*, pages 725–731. 01 2019.

[3] Monika Januszkiewicz, Christopher Parker, Steven Hayes, and Simeon Gill. Online virtual fit is not yet fit for purpose: An analysis of fashion e-commerce interfaces. pages 210–217, 10 2017.

[4] Stephan Baier. Analyzing customer feedback for product fit prediction. 08 2019.

[5] Nadia Thalmann, Bart Kevelham, Pascal Volino, Mustafa Kasap, and Etienne Lyard. 3d web-based virtual try on of physically simulated clothes. *Computer-Aided Design and Applications*, 8, 01 2011.

[6] J. Surville and Thierry Moncoutie. 3d virtual try-on: The avatar at center stage. 2013.

[7] Fanke Peng and Mouhannad Al-Sayegh. Personalised size recommendation for online fashion. 2014.

[8] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter V. Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: automatic estimation of 3d human pose and shape from a single image. *CoRR*, abs/1607.08128, 2016.

[9] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. *CoRR*, abs/1805.04092, 2018.

[10] Vivek Sembium, Rajeev Rastogi, Atul Saroop, and Srujana Merugu. Recommending product sizes to customers. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 243–250. ACM, 2017.

[11] Vivek Sembium, Rajeev Rastogi, Lavanya Tekumalla, and Atul Saroop. Bayesian models for product size recommendations. In *Proceedings of the 2018 World Wide Web Conference*, WWW ’18, pages 679–687, 2018.

[12] Romain Guigourès, Yuen King Ho, Evgenii Koriagin, Abdul-Saboor Sheikh, Urs Bergmann, and Reza Shirvany. A hierarchical bayesian model for size recommendation in fashion. pages 392–396, 09 2018.

[13] Rishabh Misra, Mengting Wan, and Julian McAuley. Decomposing fit semantics for product size recommendation in metric spaces. 10 2018.

[14] G. Mohammed Abdulla and Sumit Borar. Size recommendation system for fashion e-commerce. In *KDD Workshop on Machine Learning Meets Fashion*, 2017.

[15] Kallirroi Dogani, Matteo Tomassetti, Sofie De Cnudde, Saúl Vargas, and Ben Chamberlain. Learning embeddings for product size recommendations. In *SIGIR eCom, Paris, France*, July 2019.

[16] Abdul-Saboor Sheikh, Romain Guigourès, Evgenii Koriagin, Yuen King Ho, Reza Shirvany, Roland Vollgraf, and Urs Bergmann. A deep learning system for predicting size and fit in fashion e-commerce. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 110–118. ACM, 2019.

[17] Julia Lasserre, Abdul-Saboor Sheikh, Evgenii Koriagin, Urs Bergmann, Roland Vollgraf, and Reza Shirvany. Meta-learning for size and fit recommendation in fashion. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 55–63, 01 2020.

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

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[21] Eddie S.J. Du, Chang Liu, and D Hutchison Wayne. Automated fashion size normalization. *ArXiv*, abs/1908.09980, 2019.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[23] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[24] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

[25] Ofir Press and Lior Wolf. Using the output embedding to improve language models. *CoRR*, abs/1608.05859, 2016.

[26] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[28] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

[29] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[31] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? *CoRR*, abs/1906.02629, 2019.

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.