

# RecoBERT: A Catalog Language Model for Text-Based Recommendations

Itzik Malkiel<sup>1,2</sup>, Oren Barkan<sup>1,3</sup>, Avi Caciularu<sup>1,4</sup>, Noam Razin<sup>1,2</sup>, Ori Katz<sup>1,5</sup> and Noam Koenigstein<sup>1,2</sup>

<sup>1</sup>Microsoft

<sup>2</sup>Tel Aviv University

<sup>3</sup>Ariel University

<sup>4</sup>Bar-Ilan University

<sup>5</sup>Technion

{itmalkie, orenb, Ori.Katz, Noam.Koenigstein}@microsoft.com

avi.c33@gmail.com

noam.razin@cs.tau.ac.il

## Abstract

Language models that utilize extensive self-supervised pre-training from unlabeled text, have recently shown to significantly advance the state-of-the-art performance in a variety of language understanding tasks. **However, it is yet unclear if and how these recent models can be harnessed for conducting text-based recommendations.** In this work, we introduce RecoBERT, a BERT-based approach for learning catalog-specialized language models for text-based item recommendations. We suggest novel training and inference procedures for scoring similarities between pairs of items, that don't require item similarity labels. Both the training and the inference techniques were designed to utilize the unlabeled structure of textual catalogs, and minimize the discrepancy between them. By incorporating four scores during inference, RecoBERT can infer text-based item-to-item similarities more accurately than other techniques. In addition, we introduce a new language understanding task for wine recommendations using similarities based on professional wine reviews. As an additional contribution, we publish annotated recommendations dataset crafted by human wine experts. Finally, we evaluate RecoBERT and compare it to various state-of-the-art NLP models on wine and fashion recommendations tasks.

## 1 Introduction

Recommendation systems are a major component of content discovery in online stores. Different recommendation systems are employed across a broad spectrum of domains, such as movies, music, groceries, and more. In each case, the recommendation system is associated with a different catalog of items comprising different descriptors, item properties, and metadata. This work deals with the case of generating item-to-item similarities based on item descriptions.

Personalized recommender systems make use of either or both Collaborative Filtering (CF) or Content-Based (CB) information (Aggarwal et al., 2016). CF approaches build models based on users past behavior (Breese et al., 2013; Schafer et al., 2007). On the other hand, CB recommenders use item meta-data such as properties, tags, and descriptions in order to build and match user and item profiles (Brusilovski et al., 2007; Wang et al., 2018c; Lops et al., 2011). A model that utilizes both CF and CB is called a hybrid recommender system.

Item-to-item recommendations are commonly used in large scale recommender systems such as on Netflix (Gomez-Uribe and Hunt, 2015), Amazon (Linden et al., 2003), Xbox (Koenigstein and Paquet, 2013) and many others. Commonly found on product details page (PDP), these non-personalized recommendation lists are known to drive-up purchases as well as user engagement. Similar to personalized recommendation, item similarities can be computed based on user activity, item meta-data or both, using a variety of different models. In a new store, where user data does not exist, item-to-item recommendations are computed using one or more content-based approaches that leverage item meta-data in order to compute item-to-item similarities. The extracted data may include images, videos, textual descriptions, and more.

Textual content-based recommendation systems leverage textual information about items, such as item descriptions and titles. These models usually rely on Natural Language Processing (NLP) models to compute item-to-item similarities. A naive approach to produce recommendations from textual information is to infer similarities by embedding the textual description (and title) of every item in a latent space (Lops et al., 2011; Wang et al., 2018c; De Gemmis et al., 2015). Item embeddings,

that utilize textual descriptions, can be obtained via different types of language models.

Recently, self-supervised pre-training of language models have revolutionized the field of NLP. These techniques first utilize a self-supervised pre-training of a neural-based model using a large corpus of unlabeled text. Then, apply fine-tuning for specific NLP tasks. Among the recent self-supervised pre-trained language models, BERT (Devlin et al., 2018) has emerged as a very powerful method, achieving state-of-the-art results in a variety of NLP tasks such as sentiment analysis (Sun et al., 2019), language inference (Wu and Dredze, 2019; Cui et al., 2019), sentence similarities (Reimers and Gurevych, 2019) and more. BERT pre-training technique incorporates (1) reconstruction of randomly masked words (known as masked language model), and (2) predicting whether two sentences are consecutive (next sentence prediction).

In this work, we build upon BERT and introduce a novel technique for self-supervised pre-training of catalog-based language models. In addition, we introduce an inference technique that utilizes the above model for inferring item similarities that can be used for item-to-item recommendations in cold catalogs. Hence, we name our technique RecoBERT - a BERT model adapted for textual based recommendations.

RecoBERT pre-training leverages self-supervision to its fullest by utilizing a combination of a masked language model along with a title-description model. The latter comprises a learning task that reveals relationships between item titles and descriptions. In some cases, these relations can form a summarization task, for which titles are short sentences that summarize the longer descriptions. In other cases, catalogs may comprise items with implicit titles that incorporate a few words that were crafted for each item at hand. For both cases, the title-description task encourages the model to reveal the underlying connections between titles and descriptions, improves language understanding, and therefore yields more accurate embeddings. This results in an improved text-based item similarity performance in cold catalogs. Importantly, RecoBERT doesn't require item similarity labels nor usage data.

We also introduce a new NLP wine recommendation task, demonstrating RecoBERT's ability to find similar items in very complex domains. The

task utilizes a publicly available dataset comprising 120K elaborate wine descriptions written by wine experts. The goal is to produce wine recommendations for each item in the dataset, in the form of other similar wines. We employed a professional wine sommelier to manually craft 1095 recommendations for  $\sim 100$  wines that form a "ground-truth" test-set for evaluations. For reproducibility, and as an additional contribution, we made these annotations publicly available<sup>1</sup>.

Importantly, the novel wine recommendations task introduced in this work is different and more complex than most NLP tasks usually considered. The wine reviews incorporate domain-specific semantics, taxonomy, and phrases, as well as picturesque descriptions of tastes, aromas, and colors. Arguably, determining similarities between wine reviews is a challenging task, which requires a high level of intelligence and knowledge even to the average human. Specifically, compared to the tasks presented in the GLUE benchmark (Wang et al., 2018b), for which the average adult person can easily solve a query in few seconds, determining the similarity of wines based on their reviews may pose a challenge to most people and takes up to a few minutes even to wine enthusiasts and professionals.

The main contribution of this paper is threefold: (1) We introduce RecoBERT, a self-supervised training for catalog-based language model. (2) We introduce a novel inference technique that yields item-to-item similarities by leveraging RecoBERT, and compare its performance against relevant baselines. (3) we introduce a novel complex NLP task of wine recommendations and publish a matched labeled test set crafted by a professional sommelier.

## 2 Related Work

Recent methods in text-based recommendations suggest a hybrid approach that combines usage data with either traditional or neural-based NLP methods. In (de Souza Pereira Moreira et al., 2018; Zheng et al., 2017), the authors suggest a hybrid approach for recommendations that utilizes both session data (CF) and textual features from articles extracted by a convolutional neural network (CNN). Additionally, in Wang et al. (2015); Djuric et al. (2015) the authors proposed hierarchical Bayesian models for learning a joint representation for textual content and personal ratings, using latent Dirichlet allocation (LDA), deep au-

<sup>1</sup><https://doi.org/10.5281/zenodo.3653403>

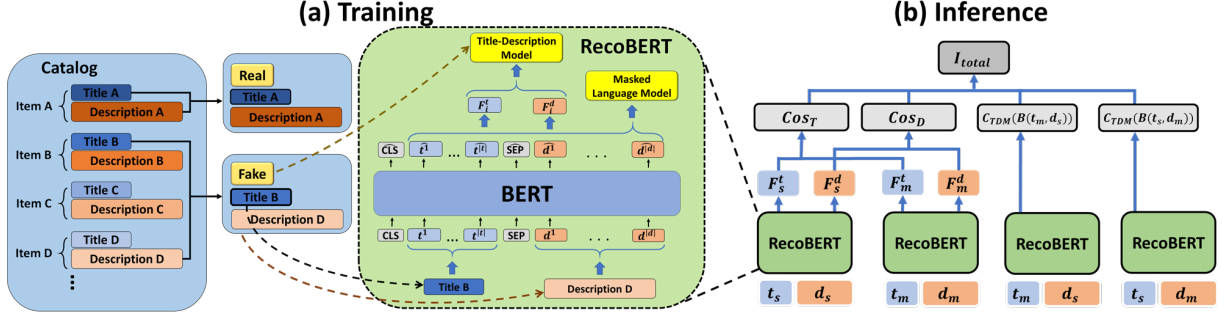


Figure 1: RecoBERT receives title-description pairs corresponding to *positive* (“real”) and *negative* (“fake”) samples, extracted from a given catalog. (a) during training, the title-description pairs are propagated through the BERT backbone and transformed into two feature vectors. These vectors are then fed into the TDM, minimizing a cosine loss between them. (b) in inference, four scores are computed. Two scores propagate the seed and candidate items separately (“real” pairs). The other two scores utilize the TDM head and propagate title-description pairs extracted from both seed and candidate items (“fake” pairs).

toencoders, and word2vec (Mikolov et al., 2013). In contrast to the above methods, the model in this paper doesn’t depend on usage data and hence can be applied to completely cold catalogs. Recently, (Gong and Zhang, 2016) proposed attentive CNN for performing hashtag recommendations for tweets. This method solely depends on text, but requires supervision for similarity. Unlike this method, our model focuses on textual catalogs and doesn’t require item-to-item similarity labels.

A recently proposed family of Transformer-based language models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019) uses multiple attention layers and a two-phase training procedure composed of unlabeled pre-training and supervised fine-tuning. These models show great promise in linguistic tasks, and were shown to exceed human-level baselines in specific tasks such as machine translation (Vaswani et al., 2017a), question answering (Yang et al., 2019), and other related tasks (Wang et al., 2018a). These models utilize *sentence embedding* techniques (Palangi et al., 2016), where a text is encoded into low dimensional vectors that summarize the information in the input text. For example, in universal sentence encoder (USE) (Cer et al., 2018), the authors suggest utilizing vectors extracted from a machine translation model for transfer learning to other NLP tasks.

Lately, Wang et al. (2019); Storks et al. (2019); Aßenmacher and Heumann (2020) claimed that human baselines are being surpassed by Transformer-based models and others that exploit statistical cues in the well-known GLUE set (Wang et al., 2018b). Such models may suffer severe performance degradation when putting to use on real-world problems.

Hence, some argue that the tasks in the GLUE dataset no longer suffice for evaluating language understanding models.

In this work, we propose a new language task that is much more complicated than the semantic similarity tasks in GLUE. Motivated by extracting item similarities for recommender systems, our task is neither composed of single sentences nor sentence pairs. Instead, the goal is to induce semantic similarity between wine items represented by sentence-paragraph pairs. Due to the complexity of the wines domain, as well as the professional language and length of the wine reviews, our novel language understanding task requires a high level of intelligence and knowledge that exceeds the average human level.

### 3 Methodology

Let  $\mathcal{W} = \{w_i\}_{i=1}^W$  be the vocabulary of tokens in a given language. Let  $T$  be the set of all possible sentences generated by  $\mathcal{W}$ , including the empty sentence. Additionally, let  $D$  be the set of all possible paragraphs generated by  $T$ . Let  $\mathcal{C} := \{m_i | m_i \in T \times D\}_{i=1}^C$  be a catalog of items, where each item is associated with a title-description pair (titles are sentences, and descriptions are paragraphs). Given a catalog  $\mathcal{C}$ , the task is to infer a similarity function  $\mathcal{F} : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$ , that scores the similarity between any pair of items  $s, m \in \mathcal{C}$ . In particular,  $\mathcal{F}$  can be used to quantify a similarity score that ranks all the items in the catalog according to their semantic textual similarity with a given seed item  $s \in \mathcal{C}$ .

### 3.1 Model Architecture and Loss Functions

RecoBERT is a function  $\mathcal{B} : T \times D \rightarrow \mathbb{R}^h \times \mathbb{R}^h$ , which utilizes a BERT-Large architecture (Devlin et al., 2019) with a hidden layer size of  $h$ , and incorporates (1) a title-description model (TDM) for scoring the relation between titles and descriptions, and (2) a mask language model (MLM) for specializing in a given domain. A dataset of  $n$  training samples is represented as pairs  $(t_i, d_i) \in T \times D$ , indexed by  $i = 1 \dots n$ . Each pair is associated with a label  $y_i \in \{0, 1\}$ , indicating whether  $t_i$  and  $d_i$  correspond to the same item.

Following the MLM procedure in (Devlin et al., 2019), RecoBERT transforms each  $(t_i, d_i)$  pair into sequences of inputs tokens  $(t_i^j)_{j=1}^{|t|}$  and  $(d_i^j)_{j=1}^{|d|}$ , masks 15% of them and adds the special  $CLS$  and  $SEP$  tokens. This input sequence is then mapped to a sequence of latent embedding tokens by propagating the input through BERT

$$BERT(I(t_i, d_i)) := (\widehat{CLS}_i, (\hat{t}_i^j)_{j=1}^{|t|}, \widehat{SEP}_i, (\hat{d}_i^j)_{j=1}^{|d|})$$

where each latent token corresponds to its matched input token.

Two feature vectors are then computed by  $F_i^t := \frac{1}{|t|} \sum_{j=1}^{|t|} \hat{t}_i^j$  and  $F_i^d := \frac{1}{|d|} \sum_{j=1}^{|d|} \hat{d}_i^j$ . Importantly,  $F_i^t$  and  $F_i^d$  correspond to the title and description of the input, respectively.

It is important to clarify the distinction between BERT and RecoBERT. Bert yields contextualized embeddings (as defined in Eq. 3.1), and can be replaced by any other language model. On the other hand, RecoBERT is defined as:

$$\mathcal{B}(t_i, d_i) = (F_i^t, F_i^d), \quad (1)$$

RecoBERT loss function is composed of two components, a TDM loss, and an MLM loss. The purpose of TDM is to learn the relationship between item titles and descriptions. To this end, we feed the model with both *positive* (“real”) title-description pairs, for which both title and description belong to the same item, and *negative* (“fake”) pairs, where the title and description are taken from two different items.

The TDM loss term utilizes a cosine head  $C_{TDM} : \mathbb{R}^h \times \mathbb{R}^h \rightarrow \mathbb{R}$ , that scores the relation between a title-description pair. Formally,

$$C_{TDM}(F_i^t, F_i^d) = \frac{1 + \text{cosine}(F_i^t, F_i^d)}{2}, \quad (2)$$

and the TDM loss is defined as

$$\mathcal{L}_{TDM} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(C_{TDM}(F_i^t, F_i^d)) + (1 - y_i) \log(1 - C_{TDM}(F_i^t, F_i^d))].$$

The purpose of the MLM is to specialize RecoBERT’s language model on the specifics of the domain and catalog at hand. As we shall see later, this has major significance in complex NLP tasks such as wine recommendations where the semantic meaning of certain words differs from their usual semantic meaning.

The MLM loss follows the paradigm presented in (Devlin et al., 2018), utilizes a classifier  $C_{MLM} : \mathbb{R}^d \rightarrow \mathbb{R}^{|\mathcal{W}|}$  that projects the embedded tokens to the vocabulary space, and applies a softmax function to infer pseudo-probabilities. The MLM loss function can be expressed as  $\mathcal{L}_{MLM} = -\frac{1}{n} \sum_{i=1}^n \sum_{(l,k) \in z_i} \log(C_{MLM}(e_l)_k)$ , where  $z_i$  is a sequence of index pairs  $(l, k)$  that correspond to the  $i$ th training sample,  $l$  and  $k$  are the indices of the masked token in  $BERT(I(t_i, d_i))$  and the vocabulary  $\mathcal{W}$ , respectively. In summary, the total loss for RecoBERT is defined as  $\mathcal{L}_{total} = \mathcal{L}_{MLM} + \mathcal{L}_{TDM}$ .

### 3.2 Training

We split the dataset into a train and validation sets. The validation set is used for early stopping, as we have found it essential, especially for smaller-sized datasets. RecoBERT backbone is initialized by the prescribed weights of the publicly available pre-trained BERT model, while the TDM head is initialized from scratch.

During training, we iterate over the items in the train set, generating *positive* and *negative* samples by switching the description to that of another item with probability  $p_s = 0.5$ . Then, the *positive* and *negative* labels are assigned accordingly. The RecoBERT model and training is illustrated in Fig. 1(a).

### 3.3 Inference

RecoBERT’s inference proceeds by generating four scores. First, we propagate every item  $(t_i, d_i) \in \mathcal{C}$  through RecoBERT, extracting  $F_i^t$  and  $F_i^d$ , as defined in Eq. 1. Then, given a seed item  $s = (t_s, d_s) \in \mathcal{C}$ , and for any item  $m \neq s, m = (t_m, d_m) \in \mathcal{C}$ , we define the two cosine scores denoted by  $Cos_D(s, m) := \text{cosine}(F_s^d, F_m^d)$  and  $Cos_T(s, m) := \text{cosine}(F_s^t, F_m^t)$ . These two cosine scores represent the similarity between (1) the



seed and candidate titles, and (2) the seed and candidate descriptions.

Next, we utilize the learned TDM head to compute additional two cosine scores. Specifically, we propagate the pairs  $(t_m, d_s)$  and  $(t_s, d_m)$  through RecoBERT, extracting  $C_{TDM}(B(t_m, d_s))$  and  $C_{TDM}(B(t_s, d_m))$ , respectively. These two scores approximate the similarity between the candidate title and the seed description, and between the seed title and the candidate description.

Finally, we normalize each score separately, across all candidate items, to have a zero-mean and a unit-variance, and define the total score as follows:

$$\mathcal{I}_{total}(s, m) = \lambda_1 \text{Cos}_D(s, m) + \lambda_2 \text{Cos}_T(s, m) + \lambda_3 C_{TDM}(\mathcal{B}(t_m, d_s)) + \lambda_4 C_{TDM}(\mathcal{B}(t_s, d_m)), \quad (3)$$

where  $\lambda_1 \dots \lambda_4$  are set to 1, and the item-to-item recommendations are obtained by sorting the candidate items according to  $\mathcal{I}_{total}$ , in a descending order. RecoBERT’s inference scheme is depicted in Fig. 1(b).

## 4 Wine Recommendations from Reviews

We introduce a novel NLP recommendation task of finding wine recommendations from reviews. The task is based on a publicly available dataset from Kaggle<sup>2</sup>, and a new test set, annotated by a professional wine sommelier. A common obstacle in evaluating similarity models is the lack of a relevant test-set or ground-truth. Therefore, as part of this paper’s contributions, we made this test publicly available. The Kaggle dataset, together with our annotated ground truth, form a new text-based recommendation task that can be further used by others in the future.

### 4.1 The Wine Dataset

The Kaggle wine dataset comprises of 120K wine titles and reviews. Each title is composed of: (1) winery name, (2) wine year, (3) wine name, and (4) grape variety. The reviews are single paragraphs descriptions written by wine experts, delineating taste, aromas, and other wine characteristics.

The descriptions frequently use a nonliteral, symbolic jargon common with wine enthusiastic and Oenologists. For example, wine sweetness can be

identified by five intensity levels, including bone-dry, dry, off-dry, sweet, and very sweet. These intensity levels substantially affect the similarity between wines. Hence, the task of wine recommendations might be considered as more complex and more difficult than many other classical NLP tasks such as sentiment analysis or question answering. While these classical tasks are relatively very simple for most humans, the wine recommendations task is arguably more difficult and convoluted even to intelligent humans.

Generally, inferring wine similarity requires the solution of the following language understanding challenges:

**1. Characteristic Intensities** Wines comprise different characteristics with different intensity levels.

**2. Characteristic Categories** Taste and aroma are classified into associative categories, and some classes are more distinct than others. For example, *apple* and *citrus* are two distinct categories of taste. Given a wine with a *hint of apple*, a recommendation for a wine with *citrus characteristics* is inadvisable by most professionals. In this example, the additional difficulty stems from the fact that a general (non-specialized) language model may consider “apple” and “citrus” to be relatively close as both are fruits.

**3. Domain-specific Semantics and Taxonomy** Compared to general language, the wine domain incorporates professional jargon with unique phrases, different semantics, and unique taxonomy. For example, the semantic opposite of the word *dry* in the English language is usually the word *wet*, however, in the context of wines, it is the word *sweet*. Similarly, the opposite of *white* is generally *black* where in the wine domain it is the word *red*.

**4. Non-literal Figurative Descriptions** Professional wine reviews incorporate symbolic descriptions that depart from their literal meaning. For example, one reviewer unfavorably described a wine named “Riscal 1860” using the words “Bulky and clumsy”, which implies that the combination between acidity, tannins, alcohol, and sugars, is out of balance.

Fig. 2 presents two representative samples from the dataset. The top example is a red wine, named “Mauchio Reserva”. Its description incorporates domain-specific phrases, such as “tannic”, cate-

<sup>2</sup><https://www.kaggle.com/zynicide/wine-reviews>

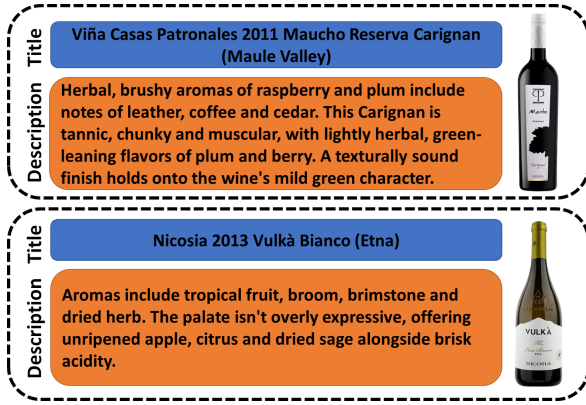


Figure 2: Two items from the wine dataset, each composed of a title-description pair. Images are shown for illustration.

gorial flavors, such as “raspberry”, “plum”, “coffee” and more. The description incorporates figurative terms, such as “chunky and muscular” and “texturally sound finish”. The second example is “Vulkà Bianco”, a white with a relatively straight forward description expressing the different flavor categories, and the intensity level of the “acidity”.

## 4.2 The Expert Annotations Set

Unlike collaborative filtering models, Content-based item to item similarity/recommendation models are very hard to evaluate. Hence, we collected a test set, annotated by professional wine sommeliers, comprising of 1095 wine recommendations to 100 wines. The sommeliers were asked to choose representative “seed” items and annotate each with  $\sim 10$  other wines that share similar characteristics with the seed item. For the sake of reproducibility and as an additional contribution, we made these annotations publicly available<sup>3</sup>.

Fig 3 exhibits one sample from the annotated expert recommendations. As can be seen in the figure, the seed and the recommended item share similar phrases, such as “ripe of black-skinned fruit”, “smooth” and “velvety tannins” from the recommended item, that can be associated with phrases in the seed item, including “black-skinned berry”, “smooth accessible palate” and “supple tannins”.

## 5 Evaluation

We evaluate RecoBERT on two datasets: (1) wines catalog and (2) fashion catalog. For each catalog, we train a separate RecoBERT model using

the Adam optimizer (Kingma and Ba, 2014) and a batch size of 16.

### 5.1 Baseline Models

We compare RecoBERT with the following models:

**Universal Sentence Encoder** (USE) suggests to leverage feature vectors extracted from a Transformer model (Vaswani et al., 2017b) for transfer learning tasks. The Transformer architecture is composed of encoder and decoder networks. During the forward pass, the Transformer receives text in a source language, forwards it through the encoder, outputs a feature vector, feeds it into the decoder, which then generates text in the target language. USE (Cer et al., 2018) utilizes the above intermediate feature vector for transfer learning to other NLP tasks, including semantic textual similarity (such as STS Benchmark (Agirre et al., 2012)), sentiment analysis (Sun et al., 2019), etc. In our work, we employ USE to generate separate embedding for every item title and description.

**Pre-trained-BERT** is the pre-trained BERT-Large model from (Devlin et al., 2018). This model was trained using a large corpus of unlabeled text, to both optimize the masked language model and the next sentence prediction (NSP) task. Since, in most datasets, item similarity labels do not exist, we can not fine-tune this model for the item similarity task. Instead, we utilize the pre-trained BERT model as a feature extractor, and extract the feature vectors  $F_m^t$  and  $F_m^d$  (see Equ. 1), for every item in the catalog.

**Specialist-BERT** is a BERT-Large model that continued pre-training using a domain-specific corpus. Specifically, we create a specialized corpus by extracting the description paragraphs of all the items in the given catalog. Then, we iterate over sentence pairs extracted from the above corpus and continue training the pre-trained BERT with the identical BERT pre-training technique, as presented at (Devlin et al., 2018). We train this model with similar settings used for RecoBERT (i.e. train-validation split, 1.5M training steps, etc.). Feature vectors are extracted in the same way as for the above Pre-trained BERT model.

**MoverScore** employs a contextualized embedding model and a variant of the Earth Mover Distance (Rubner et al., 2000) to measure the similarity between sentence-pairs (Zhao et al., 2019).

<sup>3</sup><https://doi.org/10.5281/zenodo.3653403>

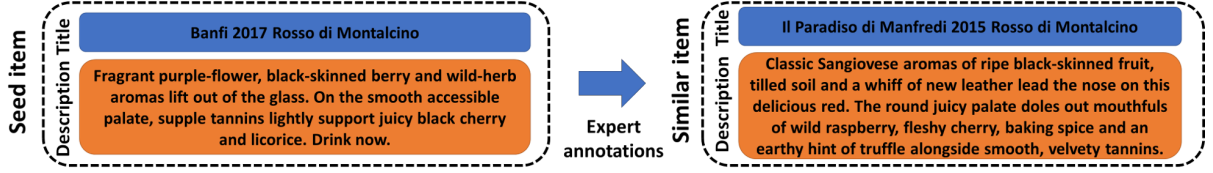


Figure 3: A representative sample from our annotated wine recommendations dataset, crafted by human wine experts.

Given two sentences, MoverScore aligns similar words from each sentence and computes the flow traveling between these words. MoverScore has recently emerged as a promising text similarity metric for text generation tasks, including summarization, machine translation, image captioning, and data-to-text generation. In our experiments, we utilize the MoverScore technique on top of the Specialist-BERT model.

**Inferencing** baseline models, besides the MoverScore, utilize the inference technique presented in section 3.3, by setting  $\lambda_3$  and  $\lambda_4$  to 0 (i.e. applying the sum  $\lambda_1 \text{Cos}_D(s, m) + \lambda_2 \text{Cos}_T(s, m)$ ). For the USE inferencing, we replace the underlying feature vectors  $(F_s^d, F_m^d)$  and  $(F_s^t, F_m^t)$  by those extracted from USE. The MoverScore baseline is applied with its own scoring technique (Zhao et al., 2019), utilizing the EMD between the latent representations of the words.

## 5.2 Quantitative Metrics

**Hit Ratio at  $k$  (HR@ $k$ )** HR@ $k$  is the percentage of the predictions made by the model, where the true item was found in the top  $k$  items suggested by the model. Specifically, a seed-candidate pair is scored with 1 if the candidate item is ranked within the top  $k$  recommendations produced by the model w.r.t. to the seed, otherwise 0. Then the average over all seed-candidate examples in the test set is reported.

**Mean Reciprocal Rank (MRR)** This measure is defined as the average of the reciprocal ranks considering the entire set of ranked items (and not just the top- $k$ ). In contrast to HR, the MRR metric takes into consideration the exact order within the recommendation list.

**Mean Percentile Rank (MPR)** Given a seed item, the percentile rank is the rank that was assigned by the recommendation model to the correct item (to be retrieved), divided by the number of ranked items. This quantity computed for all the

items in the test set and then being averaged.

For more details, we refer the reader to (Resnick and Varian, 1997).

## 5.3 Wine Recommendations Results

For the wine dataset, we compare RecoBERT with all four baselines by three different evaluations. The first two evaluations conduct item similarities by solely relying on item descriptions or item titles (but not both), and ranking 120K wine items. The third evaluation utilizes both item titles and descriptions, ranking the subset of the expert annotated wines.

In Tab.1, we report MPR, MRR, and five HR@ $k$  scores, for each evaluation, using the 1095 expert annotations. In the upper and middle parts of the table, all models solely utilize item descriptions and item titles, respectively. In both evaluations, each model ranked the entire 120K wines in the catalog, for each seed. To make a clean comparison between RecoBERT and the other BERT-based models, in these experiments, we have evaluated all BERT-based models (including RecoBERT) with the same inference score. Specifically, for the descriptions evaluations (upper part) we set all BERT-based models to solely use the  $\text{Cos}_D(s, m)$  score (by configuring  $\lambda_1$  to 1 and setting the other  $\lambda$ s with 0, i.e. we set  $\lambda_2$ ,  $\lambda_3$  and  $\lambda_4$  in RecoBERT to 0, and  $\lambda_2$  to 0 in the other BERT-based baselines). In similar, the titles evaluations (middle part) we solely utilize the  $\text{Cos}_T(s, m)$  score (setting  $\lambda_2$  to 1 and eliminating the rest of the scores). The MoverScore in each section utilizes the textual information associated with its name.

In the bottom part of the table, we report the performance of all models, utilizing both item titles and descriptions, comparing against the full RecoBERT inference, as presented in the section 3.3. In these evaluations, the reported MoverScore<sub>both</sub> separately applies the MoverScore on item titles and descriptions, ranking the items in the catalog by computing the sum of both scores.

The results in the table indicate that RecoBERT

		Model	MPR	MRR	HR@1000	HR@100	HR@50	HR@10	HR@5
120K wines	descriptions	USE $_{\lambda_2 \leftarrow 0}$	77.1%	4.3%	12.2%	4.1%	2.9%	0.7%	0.5%
		Pre-trained-BERT $_{\lambda_2 \leftarrow 0}$	80.8%	6.9%	21.7%	6.5%	4.0%	1.2%	1.0%
		Specialist-BERT $_{\lambda_2 \leftarrow 0}$	96.2%	11.6%	44.5%	17.2%	11.1%	4.5%	3.2%
		MoverScore $_{descriptions}$	96.8%	11.45%	46.84%	19.84%	12.34%	5.93%	4.81%
		RecoBERT $_{\lambda_2, \lambda_3, \lambda_4 \leftarrow 0, 0, 0}$	<b>97.3%</b>	<b>21.0%</b>	<b>64.9%</b>	<b>33.2%</b>	<b>24.9%</b>	<b>10.4%</b>	<b>6.6%</b>
	titles	USE $_{\lambda_1 \leftarrow 0}$	83.2%	5.6%	14.3%	6.5%	3.5%	0.9%	0.6%
		Pre-trained-BERT $_{\lambda_1 \leftarrow 0}$	85.6%	8.9%	23.7%	8.5%	6.2%	2.1%	1.3%
		Specialist-BERT $_{\lambda_1 \leftarrow 0}$	96.6%	13.7%	46.6%	19.4%	13.6%	5.26%	3.5%
		MoverScore $_{both}$	95.6%	14.56%	48.24%	21.25%	14.79%	6.86%	3.75%
		RecoBERT $_{\lambda_1, \lambda_3, \lambda_4 \leftarrow 0, 0, 0}$	<b>97.8%</b>	<b>23.5%</b>	<b>68.4%</b>	<b>35.6%</b>	<b>27.8%</b>	<b>12.5%</b>	<b>8.9%</b>
Expert subset	titles & desc.	USE	72.4%	18.3%	97.3%	34.4%	21.8%	8.2%	4.7%
		Pre-trained BERT	76.8%	24.3%	97.7%	44.5%	32.8%	12.3%	6.6%
		Specialist-BERT	92.3%	35.1%	<b>99.9%</b>	79.6%	59.3%	25.0%	14.7%
		MoverScore $_{both}$	93.5%	54.4%	99.8%	80.2%	67.8%	35.8%	20.7%
		RecoBERT $_{\lambda_3, \lambda_4 \leftarrow 0, 0}$	95.2%	90.3 %	99.9%	84.2%	72.0%	60.6%	23.0%
		RecoBERT	<b>96.3%</b>	<b>91.7%</b>	99.8%	<b>94.9%</b>	<b>89.6%</b>	<b>65.4%</b>	<b>38.6%</b>

Table 1: Recommendations results evaluated on the 120K wines dataset (upper part), and the subset of 1095 annotated items (bottom part).

Model	Average rank
Pre-trained-BERT	3.21
USE	3.58
Specialist-BERT	3.60
MoverScore $_{both}$	3.75
RecoBERT	<b>3.94</b>

Table 2: Expert evaluation for fashion recommendations.

outperforms all other models, in all three categories. Specifically, by solely utilizing item descriptions, RecoBERT results with MPR of 97.3% while the baselines models yield an MPR of 77.1% (USE), 80.8% (Pre-trained-BERT), 96.8% (MoverScore), and 96.2% (specialist-BERT). For MRR, RecoBERT scored 21.04%, while the baseline models ranged between 4.31% (for USE) and 11.6% (for specialist-BERT). In addition, RecoBERT presents superior performance on all HR metrics, sometimes improving by a factor of two, even compared to specialist-BERT and MoverScore which yield the best performance among the baseline models. This can be attributed to the importance of the title-description learning task, and to the benefit gained by the TDM head, which produces more accurate embedding under a cosine metric.

Notably, in the same description-based evaluations (upper part of the table), RecoBERT yields 10.4% in the HR@10 metric. This entails that on average, for each seed, RecoBERT was able to re-

trieve roughly one out of  $\sim 10$  expert annotations, in the top ten recommendations list, by ranking 120K candidate items. Remarkably,  $\sim 10$  annotated items represent  $\sim 0.0083\%$  of the entire catalog.

Additionally, as can be seen in the bottom part of the table, RecoBERT with the full inference yields better performance, by a sizeable margin, compared to all other models, including the same RecoBERT applied with the baseline inference (which solely utilizes the  $Cos_D(s, m)$  and  $Cos_T(s, m)$  scores). The latter is evidence for the benefit of applying the full inference method, which also utilizes the TDM head by propagating title-description pairs extracted from seed-candidate items.

## 5.4 Fashion Recommendations Results

We evaluate RecoBERT on a fashion catalog incorporating 4K items and compare its performance with all four baseline models. Similar to the wines evaluations, all BERT-based models were initialized with the Pre-trained BERT weights and continued pre-training using the text extracted from the fashion catalog. During inference, all models used both item titles and descriptions.

To assess the quality of the recommendations, we report human scoring conducted by a fashion expert. The same test set, composed of 100 seed items, was ranked by all models. The scoring was performed blindly, as the source model for each sample was hidden from the expert. For each seed, the expert ranked the top five recommended items,



Model	MPR	MRR	HR@10
RecoBERT $_{\lambda_3, \lambda_4 \leftarrow 0, 0}$	95.2%	90.3%	60.6%
RecoBERT $_{\lambda_1, \lambda_2 \leftarrow 0, 0}$	88.5%	75.5%	45.8%
RecoBERT $_{\lambda_1 \leftarrow 0}$	92.6%	80.5%	50.1%
RecoBERT $_{\lambda_2 \leftarrow 0}$	88.7%	77.2%	48.5%
RecoBERT $_{\lambda_3 \leftarrow 0}$	95.5%	90.6%	60.8%
RecoBERT $_{\lambda_4 \leftarrow 0}$	95.3%	90.8%	62.3%
RecoBERT	<b>96.3%</b>	<b>91.7%</b>	<b>65.4%</b>

Table 3: Ablation study results

by a total score of 0 to 5, indicating poor to excellent performance

As can be seen in Tab. 2, RecoBERT outperforms all baselines, including the ones that utilize the BERT model that was specialized in the fashion domain. Specifically, RecoBERT has gained a relative improvement of 9.4% and 5.9% compared to specialist-BERT and MoverScore, respectively. See the supplementary materials for more results of RecoBERT applied to the fashion dataset.

### 5.5 Ablation Study

Tab. 3 presents an ablation study for RecoBERT inference, evaluated on the subset of the wine expert annotations. Six variants are considered, each eliminates different scores from RecoBERT inference, by setting their matched  $\lambda$ s with 0. The results, shown in the table, indicate that it is crucial to employ all four scores, in the way it is done in RecoBERT, and that extracting information from both item titles and descriptions is highly beneficial for item similarity performance.

### 5.6 Computational Costs

We report computation times that were measured for RecoBERT training and inference, by utilizing a single NVIDIA V100 32GB GPU using PyTorch framework. For the wines catalog, we trained RecoBERT for 1.5M training steps. This training took  $\sim 5$  days. RecoBERT training on the fashion catalog, comprised 150K steps and took  $\sim 12$  hours. Inferencing RecoBERT with the same GPU allows a throughput of 340 items per second. This enables recommending the entire fashion catalog in  $\sim 7$  hours, and the wines test set in 9.5 hours. Notably, all these computations are applied once, for a given catalog, and can be executed in an offline manner and cached for later use. To further accelerate the computation time of the two  $C_{TDM}$  scores applied through RecoBERT inference, one

can adopt knowledge distillation techniques, such as (Barkan et al., 2019; Jiao et al., 2019; Lioutas et al., 2019), which are beyond the scope of this work.

## 6 Discussion and Conclusions

In this work, we introduce a novel natural language recommendation task along with a novel annotated test set that together contribute to the state-of-the-art research of text-based recommenders and language models. We present RecoBERT - A model for text-based item similarity that (a) mitigates the discrepancy between training and inference phases in the classical BERT model, by operating on sentence-paragraph pairs, (b) refines the backbone language model to provide more accurate embeddings, improving item similarities under the cosine metric, and (c) utilizes matched cosine scores as part of the inference process. In addition, we show that the unique mechanism behind RecoBERT leads to significant improvements over the other baselines and across all metrics.

RecoBERT’s preeminence stems from two properties of its TDM loss: First, feeding title-description pairs allows RecoBERT to apply cross-attention between the tokens of both elements entailing an effective dependency between their embeddings. Second, by leveraging the TDM task, RecoBERT learns an additional task for revealing the underlying connections between item titles and descriptions, which reinforces the model to better specialize in the domain at hand.

In some cases, where titles do not correlate with item descriptions<sup>4</sup>, RecoBERT can be extended to utilize textual tags<sup>5</sup>, by either concatenating the tags with the title, or feeding them as an extra input to RecoBERT.

Compared to other semantic textual similarity tasks, the proposed wine recommendations task, along with our published annotated test set, can shed light on the limitations as well as the key advances of state-of-the-art NLP models for recommendations. In addition, by publishing our annotated wine recommendations dataset, we intend to encourage the community to further explore the boundaries of other NLP models, assessing the ability of machines to understand complex human language.

<sup>4</sup>E.g. in the case of movies, titles are more cryptic and less informative.

<sup>5</sup>For example, in movies, tags can include actors, genre, etc.

## Acknowledgment

As part of this work, we have productized a similar capability in Microsoft, which utilizes the TNLRL language model (Bao et al., 2020) as a backbone. Under this context, we would like to thank Saurabh Tiwary and Saksham Singhal for many helpful discussions and advice.

## References

- Charu C Aggarwal et al. 2016. *Recommender systems*, volume 1. Springer.
- Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *\* SEM 2012: The First Joint Conference on Lexical and Computational Semantics—Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.
- Matthias Aßenmacher and Christian Heumann. 2020. On the comparability of pre-trained language models. *arXiv preprint arXiv:2001.00781*.
- Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*.
- Oren Barkan, Noam Razin, Itzik Malkiel, Ori Katz, Avi Caciularu, and Noam Koenigstein. 2019. Scalable attentive sentence-pair modeling via distilled sentence embedding. *arXiv preprint arXiv:1908.05161*.
- John S Breese, David Heckerman, and Carl Kadie. 2013. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363*.
- Peter Brusilovski, Alfred Kobsa, and Wolfgang Nejdl. 2007. *The adaptive web: methods and strategies of web personalization*, volume 4321. Springer Science & Business Media.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.
- Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.
- Marco De Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. 2015. Semantics-aware content-based recommender systems. In *Recommender Systems Handbook*, pages 119–159. Springer.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- Nemanja Djuric, Hao Wu, Vladan Radosavljevic, Mihajlo Grbovic, and Narayan Bhamidipati. 2015. Hierarchical neural language models for joint representation of streaming documents and their content. In *Proceedings of the 24th international conference on world wide web*, pages 248–255.
- Carlos A Gomez-Urbe and Neil Hunt. 2015. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19.
- Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI’16*, page 2782–2788. AAAI Press.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Noam Koenigstein and Ulrich Paquet. 2013. Xbox movies recommendations: variational bayes matrix factorization with embedded feature selection. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 129–136.
- Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.
- Vasileios Lioutas, Ahmad Rashid, Krtin Kumar, Md Akmal Haidar, and Mehdi Rezagholizadeh. 2019. Distilled embedding: non-linear embedding factorization using knowledge distillation. *arXiv preprint arXiv:1910.06720*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

- Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Paul Resnick and Hal R Varian. 1997. Recommender systems. *Communications of the ACM*, 40(3):56–58.
- Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover’s distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121.
- J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.
- Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News session-based recommendations using deep neural networks. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, DLRS 2018*, page 15–23, New York, NY, USA. Association for Computing Machinery.
- Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.
- Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018b. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Donghui Wang, Yanchun Liang, Dong Xu, Xiaoyue Feng, and Renchu Guan. 2018c. A content-based recommender system for computer science publications. *Knowledge-Based Systems*, 157:1–9.
- Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244.
- Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.
- Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.
- Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 425–434.

# Appendix



Figure 4: Two representative seeds along with their top three recommended items retrieved by RecoBERT. The left column presents RecoBERT’s top three recommendations while ranking the 120K dataset. The right column presents the recommendations for ranking the 1095 expert annotated items. Highlighted recommended items were annotated by the human expert as similar to the seed (“hit”).

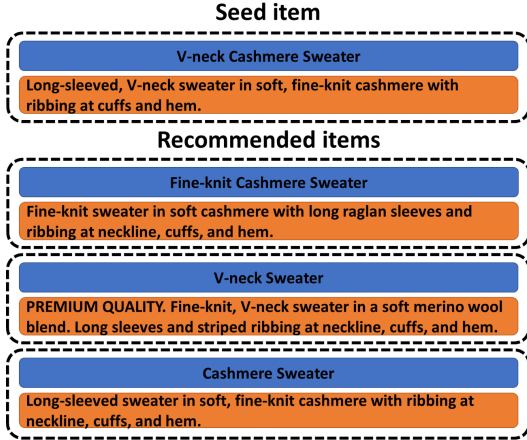


Figure 5: Fashion results. A representative seed item associated with the top three recommended items retrieved by RecoBERT.

## A Qualitative Results

Fig 4 exhibits the top three recommendations for two representative wine seeds from the annotated test set, predicted by RecoBERT, while (1) ranking the entire 120K wines in the catalog (left column), and (2) ranking the subset of 1095 annotated dataset (right column). As can be seen in the figure, the third item retrieved by RecoBERT was also annotated by the human wine expert (“hit”). Specif-

ically, for the “Gamay” seed, the origin of the first recommended wine is a neighbor country of the seed’s origin, and both items share similar berries aromas and body intensity. The second wine is a Pinot Noir, known as Gamay’s “cousin”, and shares similar berries aromas. The third item, which is one out of ten expert annotations for this seed, shares the same seed origin, aromas, and flavors.

For the seed “Domaines Devillard”, the top three recommendations retrieved by RecoBERT were all annotated by the wine expert. For this seed, all the recommendations share the same aging potential. Also, both the seed, first and third items are Cru wines, sharing the same origin and variety. The origin of the second item is Burgundy, which is considered a high-quality vineyard, similar to Cru.

Fig. 5 exhibits a representative seed from the fashion catalog, along with the top three recommendations ranked by RecoBERT. As can be seen in the figure, the top three recommended items for “V-neck Cashmere Sweater” are (1) consistent with the sweaters sub-category, (2) comprise the same material or indicated by a “PREMIUM QUALITY” label, and (3) preserve similar style properties, such as V-neck, and ribbing in a few locations on the sweater.