I need an outfit for a beach wedding that I’m going to early this summer. I’m so excited — it’s going to be warm and exotic and tropical. I want my outfit to look effortless, breezy, flowy, like I’m floating over the sand! Oh, and obviously no white! For a tropical spot, I think my outfit should be bright and colorful.
fashion concepts jointly in two languages: a style language used to describe outfits, and an element language used to label clothing items. This model answers Barthes’ question: identifying the elements that determine styles. It also powers automated personal stylist systems that can identify people’s styles from an outfit they assemble, or recommend items for a desired style (Figure 1).

We train a polylingual topic model (PLTM) on a dataset of over half a million outfits collected from a popular fashion-based social network, Polyvore! Polyvore outfits have both free-text outfit descriptions written by their creator and item labels (e.g., color, material, pattern, designer) extracted by Polyvore. These two streams of data form a pair of parallel documents (style and element, respectively) for each outfit, which comprise the training inputs for the model.

Each topic in the trained PLTM corresponds to a pair of distributions over the two vocabularies, capturing the correspondence between style and element words (Figure 2). For example, the model learns that fashion elements such as “black,” “leather,” and “jacket” are often signifiers for styles such as “biker,” and “motorcycle.”

We validate the model using a set of crowdsourced, perceptual tasks: for example, asking users to select the set of words in the element language that is the best match for a set in the style language. These tasks demonstrate that the learned topics mirror human perception: the topics are semantically coherent and translation between elements and styles is meaningful to users.

This paper motivates the choice of model, describes both the Polyvore outfit dataset as well as the training and evaluation of the PLTM, and illustrates several resultant fashion applications. Using the PLTM, we can explain why a clothing item fits a certain style: we know whether it is the collar, color, material, or item itself that makes a sweater “smart.” We can build data-driven style quizzes, predicting style preferences from a user’s outfit. We even describe an automated personal stylist which can provide outfit recommendations for a desired style expressed in natural language. Polylngual topic modeling can help us better understand our fashion theories, and support a rich new set of interactive fashion tools.

**MOTIVATION**

As more fashion data has become available online, researchers have built data-driven fashion systems that process large-scale data to characterize styles [6, 11, 12, 19, 23], recommend outfits [20, 21, 27, 32], and capture changing trends [8, 10, 29]. Several projects have used deep learning to automatically infer structure from low-level (typically) vision-based features [15, 28]. While these models can predict whether items match or whether an outfit is “hipster,” they cannot explain why. For many applications, models predicated on human-interpretable features are more useful than models that merely predict outcomes [9]. For example, when a user looks for a “party” outfit, an explanation like “this item was recommended because it is a black miniskirt” helps her understand the suggestion and provide feedback to the system.

This paper presents a fashion model that maps low-level elements to high-level styles, adapting polylingual topic modeling to learn correspondences between them [18]. Both sets of features (elements and styles) are human interpretable, and the translational capability of PLTMs can power applications that indicate how design features are tied to user outcomes, identifying peoples’ styles from the elements in their outfits and recommending clothing items from high-level style descriptions.

In addition to their translational capabilities, PLTMs offer a number of other advantages. Unlike systems built on discriminative models [21, 27, 32], PLTMs support a dynamic set of styles that grows with the dataset and need not be specified a priori. Moreover, topic modeling represents documents as distributions over concepts, allowing styles to coexist within outfits rather than labeling them with individual styles [6, 11, 12, 20]. Finally, the model smooths distributions so that systems can support low frequency queries. Even though there are no “wedding” outfits explicitly labeled “punk rock” in our dataset, we can still suggest appropriate attire for such an event by identifying high probability fashion elements associated with “wedding” (e.g., “white,” “lace”) and “punk rock” (e.g., “leather,” “studded”), and searching for clothing items which contain them.

To build a PLTM for fashion, we require data that contains both style and element information. Researchers have studied many sources of fashion data, from independent fash-
Happy Valentine’s Day! Have a nice time with your boyfriends, and don’t forget about people who are alone, weekend, romance, love, hugs, blush, valentinesday, personalstyle, #sweaterweather.

We collected text and image data for 590,234 outfits using a snowball sampling approach [7] from Polyvore’s front page, sampling sporadically over several months between 2013 and 2015. Our collection includes more than three million unique fashion items, with an average of 10 items per outfit. We collected label data for 675,699 of those items, resulting in a repository of just over 4 million item labels.

Figure 3: Polyvore outfits (left) are described at two levels: high-level style descriptions (e.g., “#valentinesday”) and specific descriptions of the items’ design elements (e.g., “red cardigan,” “lightweight shirt”). For each outfit, we process these two streams of data into a pair of parallel documents (right).

**POLYVORE OUTFIT**

Happy Valentine’s Day

Happy Valentine’s Day! I have a nice time with your boyfriends, and don’t forget about people who are alone, weekend, romance, love, hugs, blush, valentinesday, personalstyle, #sweaterweather.

**PLTM DOCUMENTS**

**STYLE**

happy, love, hugs, blush, valentinesday, personalstyle, sweaterweather

**ELEMENT**

red, short, sleeve shirts, white, tshirt, mango, oxford, lightweight, stack tops, heel, shoes

Retro sunglasses, Heart sunglasses, Hippie glasses

We leverage these two streams of data to construct a pair of parallel documents: one containing words from the title and description text to identify words that should be added to the style vocabulary, keeping hashtags such as “summerstyle” and discarding common English words that are irrelevant to fashion.

The element vocabulary is drawn from the repository’s set of Polyvore item labels. We learn frequent bigrams, trigrams and quadgrams such as “Oscar de la Renta” or “high heels” via pointwise mutual information [14]. The element vocabulary comprises these terms and any remaining unigram labels not added to the style vocabulary. After processing the repository’s text, the style vocabulary has 3106 terms and the element vocabulary 7231.

Using these vocabularies, we process each outfit’s text data into a pair of parallel documents: one containing words from the style vocabulary, and a second containing words from the element vocabulary (Figure 3, right). Both documents describe the same set of items in two different languages: an outfit might be “goth” in the style language, but the words used to describe it in the element language might be “black,” “velvet,” and “kambriel.” These parallel documents become the training input for the PLTM.

**FASHION DATA**

Polyvore is a fashion-based social network with over 20 million users [25]. On Polyvore, users create collections of fashion items which they collage together. Such collages are common in fashion: mood boards are frequently used “to communicate the themes, concepts, colors and fabrics that will be used” in a collection [24]. True mood boards are rarely “wearable” in a real sense, but on Polyvore collages typically form a cohesive outfit.

Polyvore outfits are described at two levels: specific descriptions of the items’ design elements (e.g., “black,” “leather,” “crop top”) and high-level style descriptions, often of the outfit as a whole (e.g., “punk”). We leverage these two streams of data to construct a pair of parallel documents for each outfit, which become the training inputs for the PLTM.

**Polyvore Outfit Data**

Polyvore outfit datasets contain an image of the outfit, a title, text description, and a list of items the outfit comprises (Figure 3). Titles and text descriptions are provided by users and often capture abstract, high-level fashion concepts: the use of the outfit; its appropriate environment, season, or even mood. In addition, each outfit item has its own image and element labels provided by Polyvore (Figure 3, bottom left). These labels are typically low-level descriptions of the item’s design elements, such as silhouette, color, pattern, material, trim, and designer.

Polyvore outfits are described at two levels: specific descriptions of the items’ design elements (e.g., “black,” “leather,” “crop top”) and high-level style descriptions, often of the outfit as a whole (e.g., “punk”). We leverage these two streams of data to construct a pair of parallel documents for each outfit, which become the training inputs for the PLTM.

**Representing Outfits in Two Languages**

With the outfit and item data collected, we create two vocabularies to process outfits into parallel style and element documents. The style vocabulary is created by extracting terms from the repository’s text data relating to style, event, occasion, environment, weather, etc. Most of these words are drawn from the text produced by Polyvore users since they annotate outfits using high-level descriptors; however, we also include Polyvore item labels that describe styles (e.g., “retro” sunglasses). We manually process the 10,000 most frequent words from the title and description text to identify words that should be added to the style vocabulary, keeping hashtags such as “summerstyle” and discarding common English words that are irrelevant to fashion.

The element vocabulary is drawn from the repository’s set of Polyvore item labels. We learn frequent bigrams, trigrams and quadgrams such as “Oscar de la Renta” or “high heels” via pointwise mutual information [14]. The element vocabulary comprises these terms and any remaining unigram labels not added to the style vocabulary. After processing the repository’s text, the style vocabulary has 3106 terms and the element vocabulary 7231.

Using these vocabularies, we process each outfit’s text data into a pair of parallel documents: one containing words from the style vocabulary, and a second containing words from the element vocabulary (Figure 3, right). Both documents describe the same set of items in two different languages: an outfit might be “goth” in the style language, but the words used to describe it in the element language might be “black,” “velvet,” and “kambriel.” These parallel documents become the training input for the PLTM.

**FASHION TOPICS**

To capture the correspondence between the fashion styles and elements exposed by the Polyvore dataset, we adapt polylingual topic modeling. Polylingual topic modeling is a generalization of LDA topic modeling that accounts for multiple vocabularies describing the same set of latent concepts [18]. A PLTM learns from polylingual document “tuples,” where each tuple is a set of equivalent documents, each written in a different language. The core assumptions of PLTM are that all documents in a tuple have the same distribution over topics and each topic is produced from a set of distributions over words, one distribution per language.

We train a PLTM to learn latent fashion topics jointly over the style and element vocabularies. The training input consists of the repository of Polyvore outfits, where each outfit is represented by a pair of documents, one per language. The key insight motivating this work is that these documents represent the same distribution over fashion concepts, expressed with
Generative Process

An outfit’s set of fashion concepts is generated by drawing a single topic distribution from an asymmetric Dirichlet prior

$$\theta \sim \text{Dir}(\theta, \alpha),$$

where $\alpha$ is a model parameter capturing both the base measure and concentration parameter.

For every word in both the style language $S$ and element language $E$, a topic assignment is drawn

$$z^S \sim P(z^S | \theta) = \prod_n \theta_{z^S}$$
$$z^E \sim P(z^E | \theta) = \prod_n \theta_{z^E}.$$

To create the outfit’s document in each language, words are drawn successively using the language’s topic parameters

$$w^S \sim P(w^S | z^S, \Phi^S) = \prod_n \phi_{w^S | z^S}$$
$$w^E \sim P(w^E | z^E, \Phi^E) = \prod_n \phi_{w^E | z^E},$$

where the set of language-specific topics ($\Phi^S$ or $\Phi^E$) is drawn from a language-specific symmetric Dirichlet distribution with concentration parameter $\beta^S$ and $\beta^E$ respectively.

Inference

We fit PLTMs to the outfit document tuples using Mallet’s Gibbs sampling implementation for polylingual topic model learning [16]. To learn hyperparameters $\alpha$ and $\beta$, we use MALLET’s built-in optimization setting. Each PLTM learns a distribution over style words for each topic ($\Phi^S$), a distribution over element words for each topic ($\Phi^E$), and a distribution over fashion topics for each outfit in the training set. Since choosing the optimal number of topics is a central problem in topic modeling applications, we train a variety of PLTMs with varying numbers of topics and conduct a series of perceptual tests to select the most suitable one. Figure 4 illustrates topics drawn from a model trained with 25 topics, expressing each topic in terms of high probability words in both the style and element languages.

Translation

PLTMs were not originally intended to support direct translation between languages [18]. However, in domains where word order is unimportant, given a document in one language, PLTMs can be used to produce an equivalent document in a different language by identifying high probability words. For example, given a document $w^E$ in the element language, we can infer the topic distribution $\theta$ for that document under the trained model. Since the topic distribution for a document will be the same in style language, we can produce an equivalent outfit in the style language $w^S = \theta \cdot \Phi^S$ by identifying high probability words in that language.

UNDERSTANDING FASHION TOPICS

To evaluate the trained PLTMs, we ran a set of crowdsourced experiments. These perceptual tests validate the suitability of the trained PLTMs for translation-based applications in a controlled setting.

Topic Coherence

To measure topic coherence in each language, we adapted Chang et al.’s intruder detection task [5]. The task requires users to choose an “intruder” word that has low probability from a set of the most-likely words for a topic. The extent to which users are able to identify intruder is representative of the coherence of the topic.

We performed a grid search with PLTMs trained with between 10 and 800 topics. For each trained model, we sampled up to 100 topics and found the 5 most probable words for each. An intruder topic was chosen at random, and an intruder.
truder word sampled from it. Mechanical Turk workers were shown the six words and asked to choose the one that did not belong (Figure 5).

Figure 6 shows the results of this task. Users were able to identify intruder words in the element and style languages with peak median accuracies of 66% and 50%, respectively, significantly above the baseline of random selection at 16%. The coherence peak for the element language occurred between 35 and 50 topics; the peak for the style language occurred between 15 and 35.

In both tasks, accuracy was highest for a relatively small number of topics. However, there is a tradeoff between semantic coherence and fashion nuance. With fewer topics, the model clusters fashion concepts with similar looking words and high semantic coherence: “summer,” “summerstyle,” “summeroutfit,” “summerfashion.” As the number of topics increases, topics are split into finer-grained concepts, and the semantic coherence within each topic falls off more quickly. Figure 7 illustrates this phenomenon, where the last topic shown in Figure 4 has split into two (“western” and “military”).

Figure 7: While the intruder detection results suggests using a small number of topics, there is a tradeoff between semantic coherence and fashion nuance. Although the semantic coherence within each topic falls off more quickly, a model trained with 100-topics exhibits finer-grained buckets — separate cowboy and military topics — than a 25-topic model.

Figure 6: Results of the intruder detection experiments: users successfully identified intruders in both the element and style languages compared to a baseline of random selection (dotted line). Peak performance for the style-based tasks occurs at a lower topic number than for the element-based tasks.

Translation

We also measured translational topic coherence through perceptual tasks. Users were shown the top five words from a topic in one language and asked to select the row of words that best matched it in the other language (Figure 8). One row of words was drawn from the same topic as the prompt, while the other three were drawn at random from other topics. Users were shown groups of words (rather than single words) to provide a better sense of the topic as a whole [5]. We restricted this test to models with between 15 and 100 topics, since the word intrusion results showed highest topic coherence in that range.

Figure 8: To measure translational topic coherence, Mechanical Turk workers were shown five likely words from a topic in one language and asked to choose a row of words in the other language that was the best match.

Figure 9 shows the results from this task. Performance was similar in both translation directions, with a peak median agreement with the model of 60% with prompts in the style language, and a peak median agreement of 66% with prompts in the element language, where the baseline of random selection is 25%. Accuracy is again highest for a relatively small number (25–35) of topics.
APPLICATIONS

We describe three translation-based fashion applications powered by the trained PLTMs, illustrating how human-interpretable features can lead to a richer understanding of fashion style. We show how analyzing the topics learned by the model can answer Barthes’ question. In addition, translating an outfit from an element document to a style one powers a style quiz and translating from a style document to an element one supports an automated personal stylist system.

Answering Barthes’ Question

To answer Barthes’ question, we can directly analyze the learned topics (style concepts) to understand which features (words in the element language) act as signifiers. For some topics, the probability mass is concentrated in one fashion element; for others, the distributions are spread across several features. By computing the entropy of the word distributions in the element language,

\[ H(\Phi^E) = - \sum_{i=1}^{n} P(w_i) \ln P(w_i), \]

we can measure which topics are characterized by one (low entropy) or several (high entropy) fashion elements.

Figure 10 (top) shows three topics that have low entropy: a single word determines each style. The next three topics have high entropy, with many equally-important features coming together to create the style. For a “prom” style, “dress” alone signifies; for a “winter” style, many signifiers (“leather”, “long”, “black”, “wool”, “sweater”) come together.

Style Quiz

Fashion magazines often feature “style quizzes” that help readers identify their style by answering sets of questions like “you are most comfortable in: (a) long, flowing dresses; (b) cable-knit sweaters; (c) a bikini” or selecting outfit images they prefer. While these quizzes are fun, the style advice they provide has limited scope and utility.

Figure 10: To answer Barthes’ question, we analyze each topic — style concept — to understand which features — words in the element language — act as signifiers. Some style concepts are determined by one or two elements (low entropy); for others, several elements come together to define the style (high entropy).
Applications built on our model can help users understand their personal style preferences using an open-ended interaction that provides a rich set of styles — and a confidence measure from the model of those style labels — as a result. Users capture their style by creating an outfit they like (Figure 11, left); the set of words for the items in the outfit forms a document in the element language. We can then infer a topic distribution for this document and find the highest-probability words in the style language. We measure confidence for these style labels by computing the inverse of the topic distribution’s entropy.

When an outfit draws from several topics at once, there is no single dominating style. High entropy outfits sometimes appear to be a confusing mix of items; other times users seem to intentionally mix two completely disparate styles (e.g., romantic dresses with distressed jean jackets). Indeed, the user who created the lowest confidence outfit in the repository labeled it “half trash half angel,” evidently having exactly such a juxtaposition in mind!

**Automated Personal Stylist**

While personal stylist systems can provide useful advice on constructing new outfits or updating a user’s wardrobe, existing recommendation and feedback systems typically have limited sets of styles [11, 21, 32] or must connect users to human workers [26, 4, 19]. The learned PLTM allows users to describe their fashion needs in natural language — just as they would to a personal stylist — and see suggestions a stylist might recommend.

We introduce a system that asks users to describe an event, environment, occasion, or location for which they need an outfit in free text. From this text description, the system extracts any words contained in the style vocabulary to produce a new style document. Then, it infers a topic distribution for this new document and produces a set of high-probability words in the element language that fit that document. The top 25 such words are then taken as candidate labels, and compared to each of the 675,669 labeled items in the database. The system measures the goodness of fit of each item using intersection-over-union (IOU) of the two sets of labels

\[
\text{IOU}(l_i, l_j) = \frac{|l_i \cap l_j|}{|l_i \cup l_j|}.
\]

The system ranks the items by IOU, groups the results by most frequent label, and presents the resultant groups to the user (Figure 12).

**CONCLUSIONS AND FUTURE WORK**

This paper presents a model that learns correspondences between fashion design elements and styles by training polylingual topic models on outfit data collected from Polyvore. Systems built on this model can bridge the semantic gap in fashion: exposing the signifiers for different styles, characterizing users’ styles from outfits they create, and suggesting clothing items and accessories for different needs.

One promising opportunity to extend the presented model is to leverage streams of data beyond textual descriptors, including vision-based, social, and temporal features. Training a joint model that uses computer vision to incorporate both visual and textual information could well lead to a more nuanced understanding of fashion style. Similarly, mining Polyvore’s social network structure (e.g., likes, views, comments) could enhance the model with information about the popularity of fashion styles and elements [30, 13], or how fashion trends form and evolve through time [10, 8, 29].

While the translation-based experiments described in the paper validate the suitability of PLTMs for fashion applications in a controlled setting, we are eager to perform more meaningful user testing “in the wild.” Deploying the tools described in the paper at scale and monitoring how they are used would allow us to build more personalized and context-aware models of fashion preference. The semantics of fashion change by location, culture, and individual: the “decora” style might not make sense outside of Japan; “western” outfits might only be worn in the United States; individuals may not agree on what constitutes “workwear.” Better understanding how different users interact with our tools is a necessary first step towards making them truly useful, and enabling them to dynamically adapt to different people and contexts.

The framework presented in this paper is not limited to fashion. Design artifacts in many domains contain latent concepts...
Figure 12: A personal stylist interface that recommends fashion items based on natural language input. We extract the style tokens from a user’s description of an outfit, infer a topic distribution over the style document, and compute a list of high probability words in the element language. Users are shown items ranked by intersection-over-union over the top element words.
that can be expressed with sets of human-interpretable features capturing different levels of granularity [1, 17]. This model also offers attractive capabilities: it can infer latent concepts of a design, translate between different feature representations, and even generate new artifacts. In the future, we hope that this framework can power new applications in domains like graphic design, 3D modeling, and architecture.

ACKNOWLEDGMENTS
We thank P. Daphne Tsatsoulis for her early contributions to this work, and David Mimno for his helpful discussions of the PLTM.

REFERENCES