

# Decomposing Occasion Semantics for Clothing Rental Reviews

Jiajun Bao  
jiajunb@umich.edu  
Computer Science

Qinyun Wang  
qinyunw@umich.edu  
Computer Science and Engineering

## 1 Introduction

Customers usually buy a combo of items as whole for one single goal. For example, customers rent a set of clothes for dating rather than a single pair of trousers. Here, we examined how customer reviews discourse information about the occasion that he/she rented the clothes for. Through tests with real users, we demonstrated that humans can also accurately predict the usage occasions from their customer review. Next, we developed a series of computational models for predict these occasions. Finally, we applied associations rule to figure out the clothing type sets where different type of clothing are frequently rented together, and the corresponding occasions when customers rent them.

## 2 Data

We evaluate our models on the RentTheRunWay dataset [6]. RentTheRunWay is an online cloth renting platform that allows women to rent clothes for various occasions. The dataset contains self-reported reviews from the customers, with other side information like product transactions, ratings, product categories, sizes, usage occasions. The statistics of the data set is summarized in Table 1.

Statistic/Dataset	RentTheRunWay
# Transactions	192,544
# Customers	105,571
# Products	30,815
# Customers with 1 Transaction	71,824
# Products with 1 Transaction	8,023

Table 1: General Table Statistics

## 3 Data Analysis

Prior study has shown that the linguistic cues in customer views are sufficient to predict why they bought the product. Here, we ask (1) whether humans could infer the usage occasion of a product from customer reviews and (2) whether this intuition could be recovered with computational models.

### 3.1 Q1: Do customer reviews reflect products' usage occasion?

Customers rent a set of clothes for dating rather than a single pair of trousers. In section, we explored the humans baseline on the task of predicting the usage occasions from their customer review.

**3.1.1 Data.** The dataset for this section contains the occasions and customer reviews for in each transaction from the original dataset. The statistics is provided in tabel 2. We will use this dataset for both section 3.1 and 3.2.

**3.1.2 Challenges.** The quality of customer reviews varies across different users. While some users give highly-qualified reviews, others simply left the review empty or short reviews that are not informative for analysis. It causes much noise for our analysis. To solve this issue, we apply additional heuristics when cleaning data for annotations.

**3.1.3 Experimental Setup.** Given a customer review for a product, annotators are asked to judge where is the product used from seven pre-defined categories listed in table 3. Data was selected using two heuristics to maximize informativeness with respect to the goals of this experiment. First, we sampled a fixed number of reviews from each categories. Second, the probability for a review to be sampled was proportional to its length. The long the review was, the more likely it was sampled. Up to 50 reviews were selected from each category.

**3.1.4 Results.** The human performance of each category is listed in table 3. We showed that it is feasible for human to predict on which occasion is a product used from its customer reviews, with an overall accuracy of 0.60. The "other" category has a pretty low accuracy. We think it may because human tends to give a nontrivial category.

### 3.2 Q2: Can we reveal those cues with computational models?

Given the ability to predict their usage occasions, online platforms can evaluate the similarity score of two products [8]. This study has shown that the linguistic cues in customer reviews imply the usage occasion of products.

However, manually annotating the occasions is expensive and it lacks the ability to generalize to other dataset. One further question is whether we could reveal those cues with computational models.

Given a customer review  $R = \{S_0, S_1, \dots, S_A\}$  for product  $p$ , our task is to predict which occasion category  $C_o$  the product  $p$  belongs to.

#### 3.2.1 Technique.

##### 3.2.1.1 SVMs

SVM-based methods are commonly used as a classifier [3]. They are included in our classes, so we will not go deep in their introduction.

##### 3.2.1.2 Hierarchical Attention Networks (hattn)

Hierarchical Attention Networks are reported in [10]. Given a sentence  $S_i$  with words  $w_{it}$ , we use a bidirectional GRU [1] to summary the word contents from both directions:

$$x_{it} = W_e w_{it}, t \in [1, T]$$

$$\vec{h}_{it} = \overrightarrow{\text{GRU}}(x_{it}), t \in [1, T]$$

$$\overleftarrow{h}_{it} = \overleftarrow{\text{GRU}}(x_{it}), t \in [T, 1]$$

Occasion	#Products	words	vocabulary
wedding	57784	3869480	50598
formal affair	40408	2808939	43789
party	35626	2114963	37989
everyday	16822	752552	21041
other	15388	963699	25252
work	15042	701391	20168
date	7388	422426	15805
vacation	4075	213065	10879

**Table 2: Occasions Statistics: #Products denotes the number of products that are self-reported in the occasion, #words denotes the number of words in reviews for the occasion, #vocabulary denotes the size of the vocabulary of reviews for the occasion.**

Occasion	Accuracy
wedding	0.73
formal affair	0.65
party	0.93
everyday	0.52
work	0.76
date	0.35
vacation	0.61
other	0.25
Overall	0.60

**Table 3: Human baseline: "Overall" denotes the micro-average of the accuracy listed.**

We concatenate the forward hidden state  $\vec{h}_{it}$  and backward hidden state  $\overleftarrow{h}_{it}$  to obtain the representation for the sentence  $S_i$ . Like [10], we employ attention mechanism to reveal words that we important to the meaning of the sentence:

$$u_{it} = \tanh(W_w h_{it} + b_w)$$

$$\alpha_{it} = \frac{\exp(u_{it}^\top u_w)}{\sum_t \exp(u_{it}^\top u_w)}$$

$$s_i = \sum_t \alpha_{it} h_{it}$$

The context vector  $u_w$  is jointly learned during learning process. It indicates the information that can be derived from the its words [9].

### 3.2.1.3 ALBERT

BERT [2] is a pre-trained language model that improves a variety of language understanding tasks. It guides a recent trend of pre-training deep representations from unlabeled text and then applying task-specific fine-tuning. ALBERT [4] inherits a similar paradigm with new design choices, scaling much better compared to the original BERT.

**3.2.2 Experimental Setup.** We use 80% of the whole dataset for training, 10% for validation, and the remaining 10% for test, unless stated otherwise.

In SVMs, we first initialized the embedding layer with CountVec-torizer to tokenize our input text and generate the vector representation. Then we used TF-IDF to generate the importance score for each word in its review text. We set the Regularization parameter as 1, using "rbf" as the SVM kernel with a degree of 3. Then we trained the SVM model until it was convergent.

In h attn, we tokenize the documents with nltk tokenizers [5]. When building the vocabulary, we only retain words that appear more than 5 times in the documents. The limit of sentences per document is 15, and that of words per sentence is 20. We initialized the word embeddings with GLOVE [7] and finetuned them during training. We set the dimension of word-level rnn hidden state to be 50, while that of the sentence-level rnn hidden state is 200. The concatenation of forward and backward GRU give us sentence representations of 100 dimensions. For training, we used a mini-batch size of 128 and an Adam optimizer with a learning rate of  $8 \times 10^{-4}$ . We trained the model for 400 epochs.

In the ALBERT model, we loaded the pretrained **albert-base-v2** model and tokenized the documents with its default SentencePiece tokenizer. We used a mini-batch size of 32 and an AdamW optimizer with a learning rate of  $1 \times 10^{-5}$ , a weight decay of  $1 \times 10^{-5}$ . We trained the model for 5 epochs.

**3.2.3 Observations.** The experimental results of all models are shown in table 4. Our models did learn some non-trivial information from our dataset, recovering a large part of human intuition in which occasion are the items used. The ALBERT model performed the best.

## 3.3 Q3: What types of clothing are likely to be rent together ?

**3.3.1 Data.** The data we used in this question is the types of clothing and customer Id. If different types of clothing are rented at the same time by a single customer, we can believe that there is a potential like between these clothing types. Say a person who rent a dress is likely to rent a sheath together, instead of a mini or a skirt. Therefore the online clothing renting platform can push a more accurate advertisement to the customer based on his/her online shopping cart.

**3.3.2 Technique & Challenges.** Apriori is an algorithm for frequent item set mining and association rule learning over relational

Methods	Date	Everyday	Formal affair	Party	Vacation	Wedding	Work	Other	Overall
SVM	<b>0.42</b>	0.39	0.44	0.41	<b>0.53</b>	0.51	0.49	<b>0.38</b>	0.44
hattn	0.25	0.53	0.49	0.43	0.27	0.72	<b>0.50</b>	0.22	0.52
ALBERT	0.21	<b>0.62</b>	<b>0.53</b>	<b>0.46</b>	0.31	<b>0.74</b>	<b>0.50</b>	0.20	<b>0.54</b>

Table 4: Document Classification Accuracy on the full dataset. "Overall" denotes the micro-average of the accuracy listed.

Table 5: Top 10 clothing type category

Clothing Type	Number of Data
dress	92884
gown	44381
sheath	19316
shift	5365
jumpsuit	5184
top	4931
maxi	3443
romper	3730
jacket	2404
mini	1751

databases. By applying Apriori algorithm we can identify the frequent individual clothing type sets in the whole data set, and extending them to large clothing type sets when those item sets appear sufficiently often in the whole data set. There are two main challenges based on the data set we have. Firstly, the clothing counts per clothing type is quite unbalanced, so the matrix of clothing type by user is mainly sparse. Secondly, most customer only have one single review based on one single item, which is meaningless in the analysis.

### 3.3.3 Experimental Setup.

#### 3.3.3.1 Data Selection

We need to select the required features from the data set. We grouped the original data by the user\_id with and date. Then we used set function to filter out the different clothing types a customer rent on a single day.

#### 3.3.3.2 Data Reduction

We omitted the data with only one type of clothing renting since it would not give us information about the relationship between different clothing types. For each row of the data set represents the clothing types rented together on the same day by the same person. And finally we got 24118 lines of records.

#### 3.3.3.3 Encoding

To make our data set fit the input format of Apriori Algorithm, we need to encoding each row of data with 1s and 0s. For each row we would maintain two sets: a uncommon set for types that not appears in this row, and a common set for types that appears in this row. We filled a new matrix with 1s if items in a row are in the common set, and 0s if items in a row are in uncommon set.

#### 3.3.3.4 Applying Apriori Algorithm

We applied the Apriori by using the mlxtend library in python, which implements a fast and efficient Apriori Algorithm. The minimal support is set to 0.05 and the min\_threshold for confidence is set to 0.2.

**3.3.4 Observations.** The Table(6) shows the statics and relationship between the antecedent type list and the consequent type list. We can find that dress is the most possible consequent with a high confidence score and lift score, which means when people choose other clothing, they are likely to rent a dress together. It make sense because dress type has the largest number in the whole data set. We can also conclude that there is a strong connection between dress and sheath, gown and dress. They have the highest support score as well as confidence score. Online platform can recommend items based on the consequent type given users' current viewing type.

The Figure(1) shows the relationship between support and confidence for each association rule. The data points are sparse in the figure, but we can infer that when the support of a rule is low, the confidence could be high, which meaning the rule is convicting even it is a rare rule.

The Figure(2) shows the relationship between lift and confidence for each association rule. We can find that the lift score is likely to have positive correlation with confidence score. and the lift score is relatively high for the dress type.

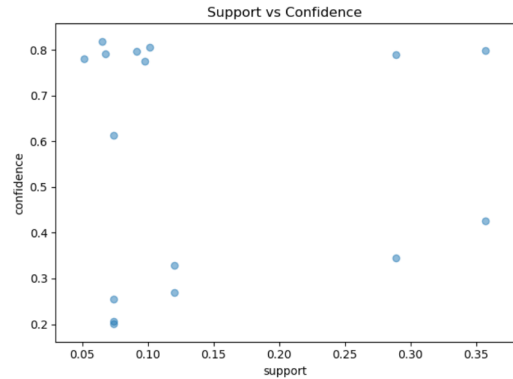
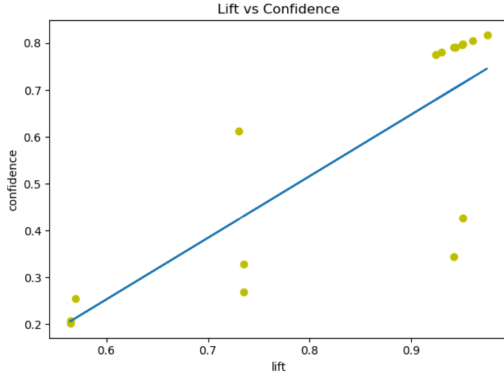


Figure 1: Support vs Confidence

**Table 6: Association Rule Results**

Antecedent	Consequent	Support	Confidence	Lift	Occasion
gown	sheath	0.120	0.269	0.735	wedding
sheath	gown	0.120	0.329	0.735	wedding
dress	sheath	0.289	0.345	0.942	wedding
sheath	dress	0.289	0.790	0.942	wedding
maxi	dress	0.068	0.792	0.942	wedding
romper	dress	0.065	0.818	0.974	party
jumpsuit	dress	0.097	0.776	0.924	party
shift	dress	0.101	0.805	0.960	party
jacket	dress	0.052	0.780	0.930	everyday
gown	dress	0.357	0.798	0.951	wedding
dress	gown	0.357	0.426	0.951	wedding
gown, dress	sheath	0.074	0.207	0.565	wedding
gown, sheath	dress	0.074	0.613	0.730	wedding
dress, sheath	gown	0.074	0.255	0.570	wedding
sheath	gown, dress	0.074	0.202	0.565	wedding

**Figure 2: Lift vs Confidence**

## 4 Conclusions & Discussion

Our project sheds light on how to build more successful recommendation systems for clothes. Our work shows that human could capture linguistic cues from customer reviews to predict why customers bought the product. Theoretically, our work contributes to the understanding of customers’ personal disclosure during purchasing. Moreover, we proposed several computational models to allow this exploration in large scales. Furthermore, we also analyze the inner relationship between different types of items based on the association rules to make a comprehensive recommendation systems.

The challenges we faced includes:

- Dirty data with lots of null, meaningless words, meaningless characters in customer’s reviews.
- Dealing with unbalanced data that would cause bias on the items with high numbers.

- Brainstorming about inner relationship of our data set and pick up the idea to make recommendation system for online renting platform.

We learnt:

- The on-hand data preprocessing skills to deal with the dirty data.
- Practical methods for mining values from large scale datasets like techniques discussed above.

We loves every parts of our project, especially when we learned that we would not have a final exam thanks to this project. Specifically, we enjoy reading creative comments and jokes in customer reviews.

Both members participates in the brainstorming. Jiajun Bao wrote the introduction, section 3.1 and 3.2. Qinyun wrote section 2 and section 3.3. Both members participates in writing the rest parts. Jiajun Bao worked on the hattn and ALBERT methods, and human annotation. Qinyun worked on the data filtering, SVM methods, and associated rule learning.

## References

- [1] CHUNG, J., GÜLÇEHRE, Ç., CHO, K., AND BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR abs/1412.3555* (2014).
- [2] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [3] HEARST, M. A. Support vector machines. *IEEE Intelligent Systems* 13, 4 (July 1998), 18–28.
- [4] LAN, Z., CHEN, M., GOODMAN, S., GIMPEL, K., SHARMA, P., AND SORICUT, R. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations* (2020).
- [5] LOPER, E., AND BIRD, S. Nltk: The natural language toolkit. *CoRR cs.CL/0205028* (2002).
- [6] MISRA, R., WAN, M., AND MCAULEY, J. Decomposing fit semantics for product size recommendation in metric spaces. In *Proceedings of the 12th ACM Conference on Recommender Systems* (New York, NY, USA, 2018), RecSys ’18, Association for Computing Machinery, p. 422–426.
- [7] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543.
- [8] SCHAFER, B., J. B., FRANKOWSKI, D., DAN, HERLOCKER, JON, SHILAD, AND SEN, S. Collaborative filtering recommender systems.
- [9] SUKHBATAAR, S., SZLAM, A., WESTON, J., AND FERGUS, R. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2440–2448.
- [10] YANG, Z., YANG, D., DYER, C., HE, X., SMOLA, A., AND HOVY, E. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (San Diego, California, June 2016), Association for Computational Linguistics, pp. 1480–1489.