Machine Learning @ Amazon

Rajeev Rastogi
Director, Machine Learning

amazon

Numerous ML Applications

Operations

- Abuse/Fraud
- Address Quality
- Packaging
- Substitutes Prediction
- Payment Instruments Success Prediction

Customers

- Product Recommendations
- Product Search
- Product Ads
- Customer Targeting
- Lending

Sellers

- Demand Forecasting
- Pricing
- Fraud Detection
- Seller Lead Generation

Catalog

- Product Classification
- Duplicate Products Detection
- Product Attributes Mismatch Detection
- Attributes Extraction from Titles/Images

Text

- Named-Entity Extraction
- Reviews Summarization
- Reviews Ranking and Insights
- Fraudulent Reviews

Images

- Visual Search
- Product Image Quality
- Celebrity Faces
- Incorrect Postures
- Brand Tracking

Speech

- Automatic Speech Recognition
- Natural Language Understanding
- Dialog Management

AWS

- Predicting Resource Requirements
- DDoS Detection
- Reputation Computation of MTurk Workers

amazon

Address Quality

Problem

- *Identify and correct poor quality addresses*

Fake addresses

- ***"Pakistan, Sharjah, Lahore"***
- ***"asdf asdf"***

Wrong Pincodes

- ***"28th floor, World Trade Centre, Bangalore, 400001"***
- ***"28th floor, World Trade Centre, Bangalore, 560001"***

Incomplete address

- ***"Near Orion Mall, Malleswaram, Bangalore"***
- ***"Near Bus Stand, Sambhaji Chowk, Nasik"***

Spelling errors

- ***Shop 999 WorliVullage janaata coloniny, 7 1 328***
- ***kodamachalivaristreetpaturutadapalligudem***

Product Packaging

Problem

- *Determine the most cost-effective packaging to use for a product*

SIOC

Poly bag

Bubble package

Variable depth box

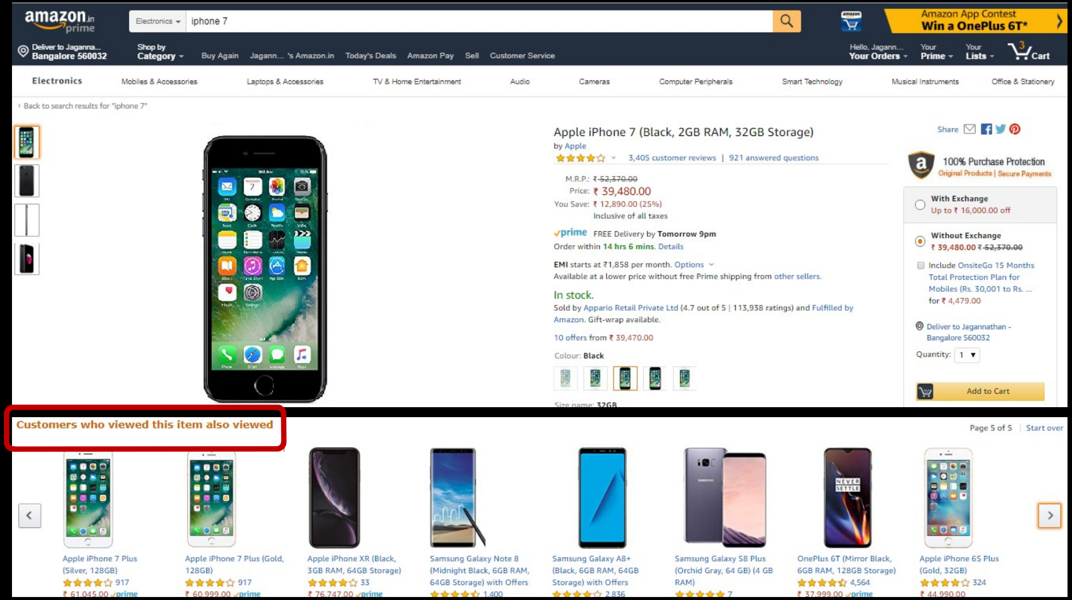
Corrugated box

Low ← Packaging Cost → High

Product Substitutes

Problem

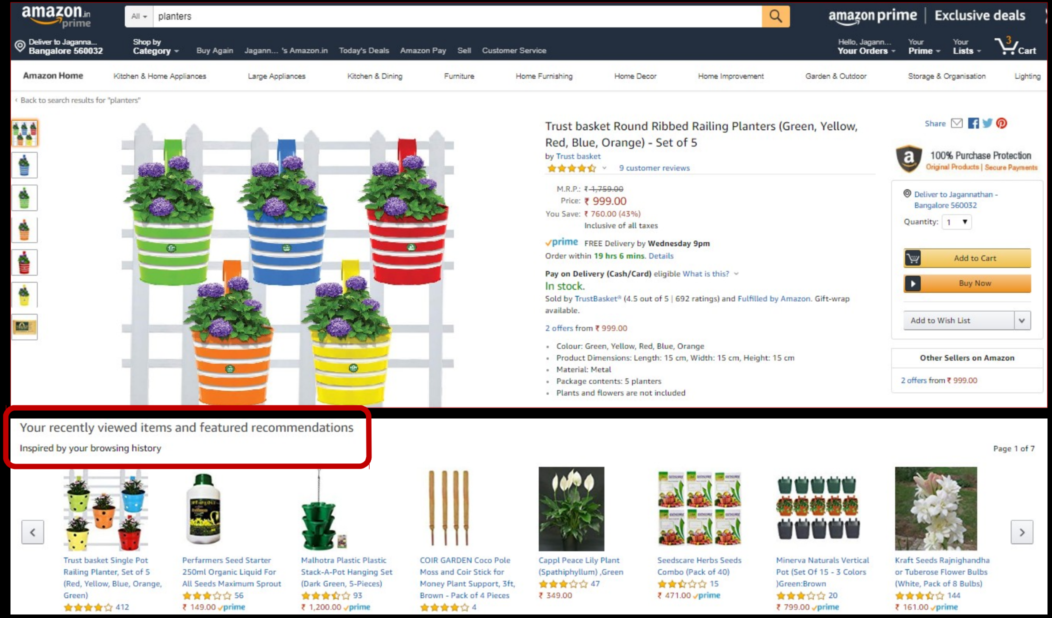
- *Identify substitutes for a product*



Product Recommendations

Problem

- *Recommend products to customers that match their preferences*

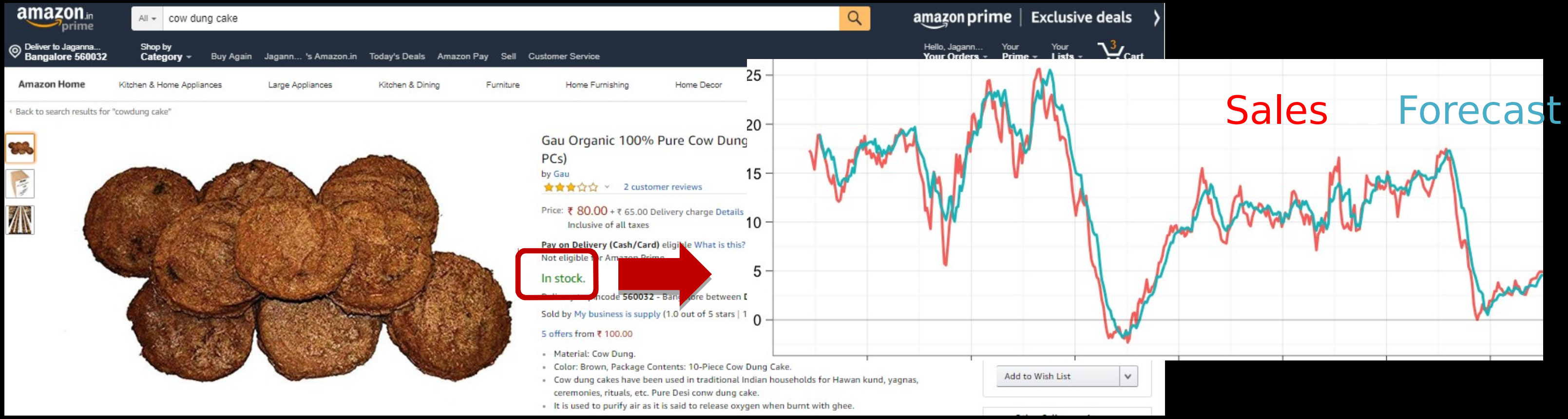


amazon

Product Demand Forecasting

Problem

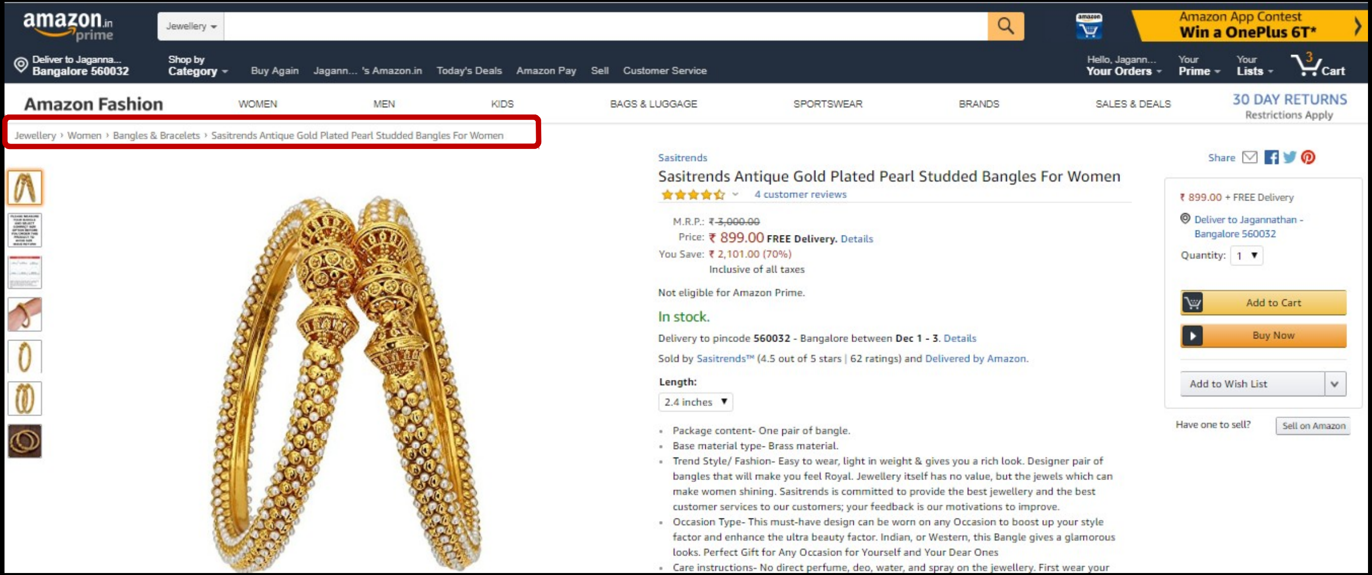
- *Given past sales of a product in every region, predict regional demand up to one year into the future*



Product Classification

Problem

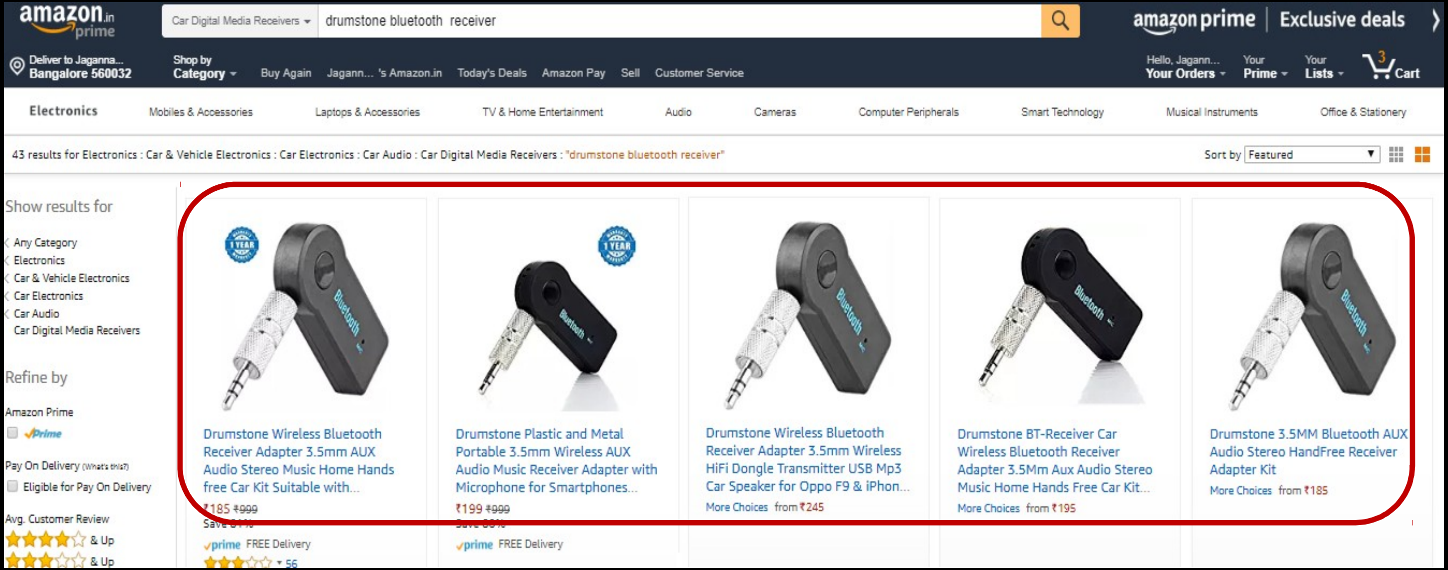
- *Classify each product into the appropriate leaf node in product taxonomy*



Product Matching

Problem

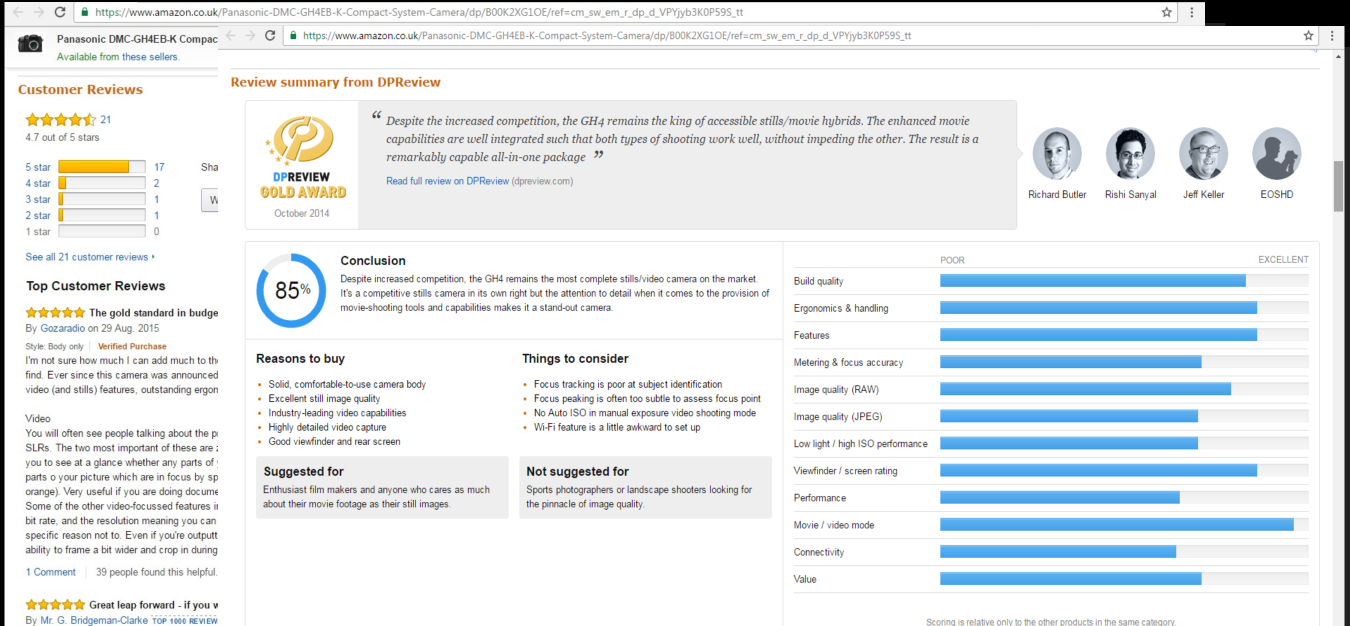
- *Identify duplicate product listings in Amazon catalog*



Insights Extraction from Reviews

Problem

- *Extract fine-grained attribute ratings from product reviews*



amazon

Outline

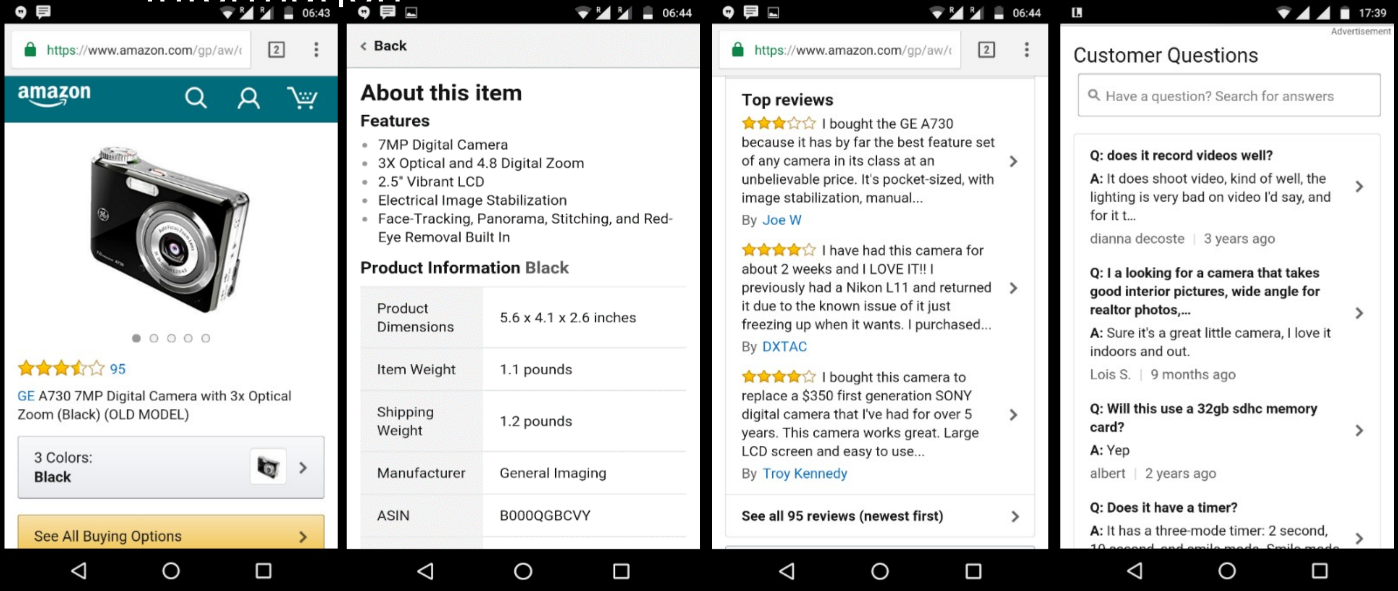
- Question Answering
- Catalog Quality
- Product Size Recommendations

Outline

- **Question Answering**
- Catalog Quality
- Product Size Recommendations

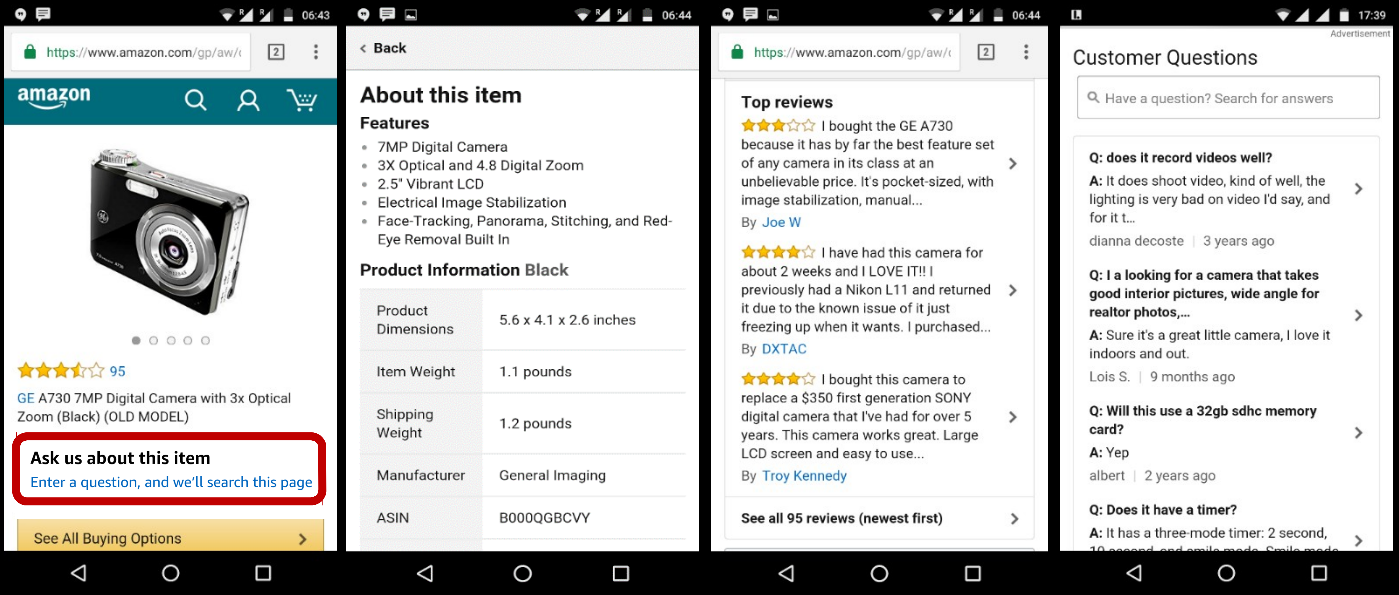
Amazon Product Pages

- Amazon product pages contain a wealth of information

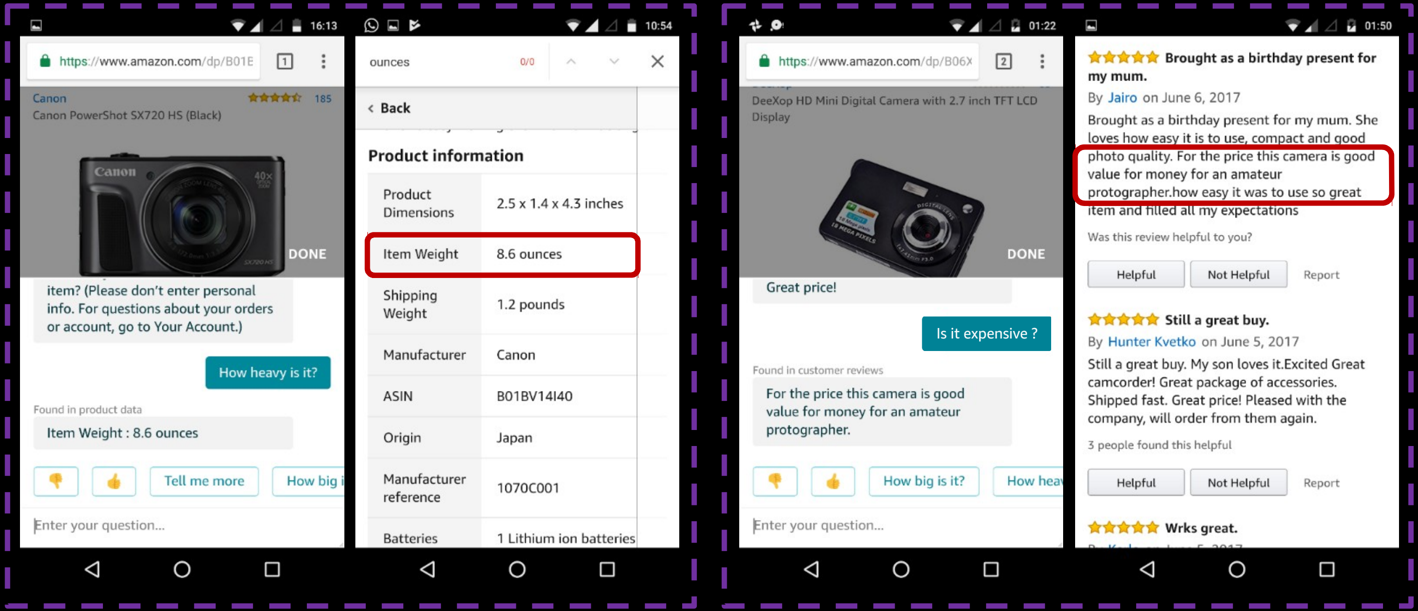


Question & Answering Bot

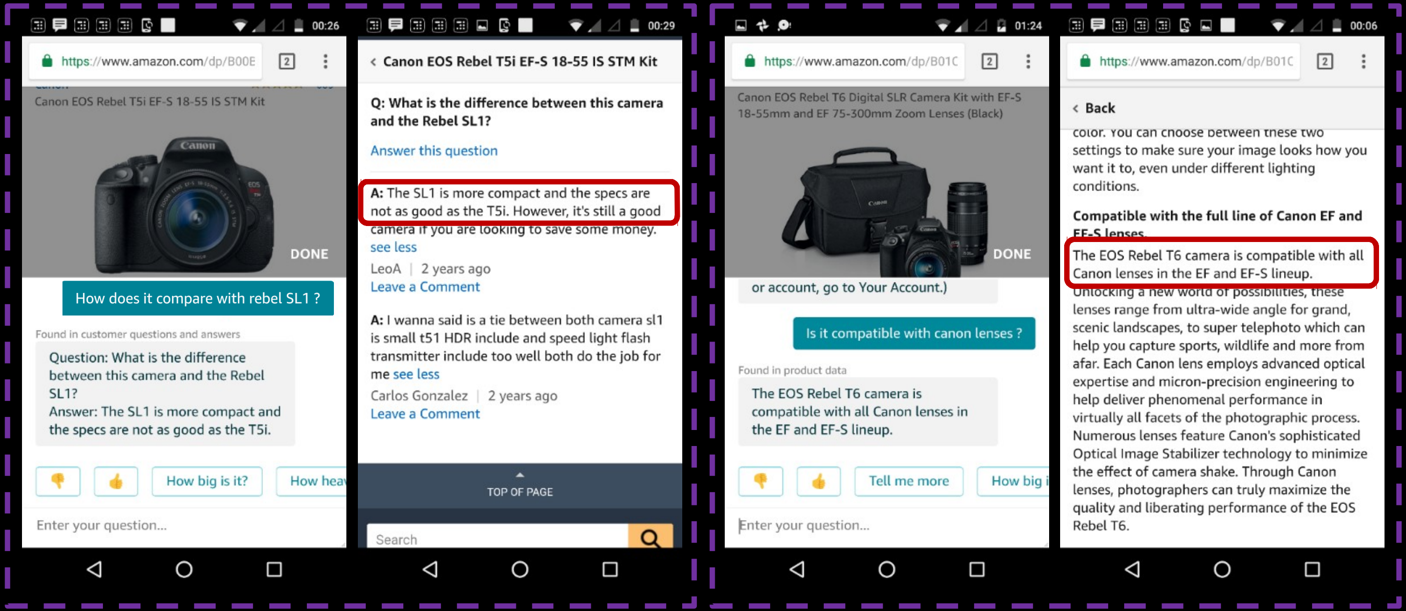
- Question answering interface to make it easy for users to find information on product page



Product Feature Questions



Product Comparison/Compatibility Questions



Key Challenges

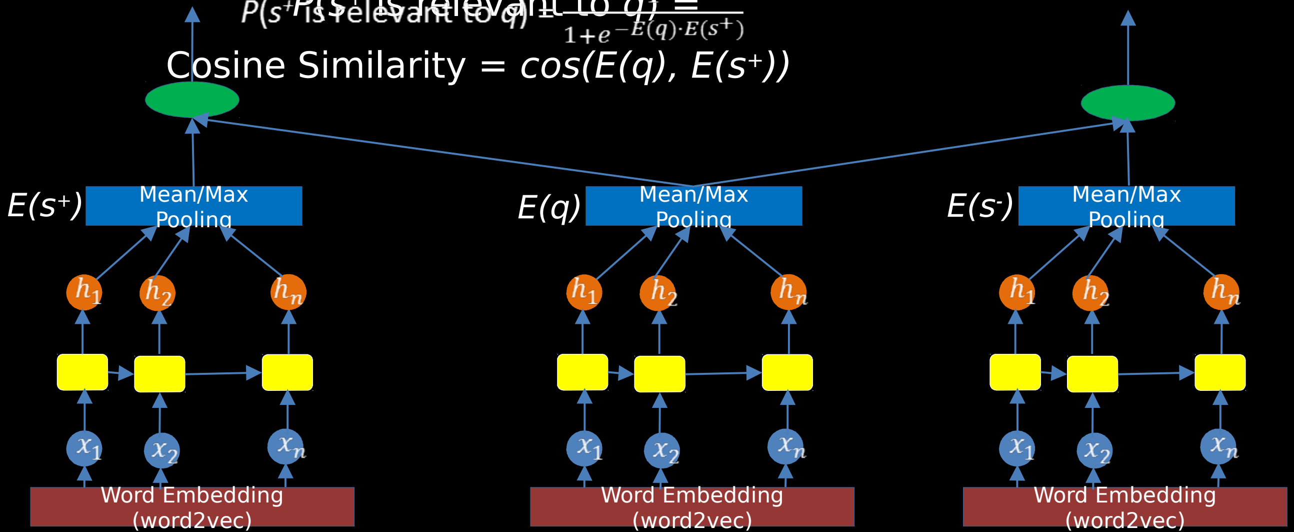
- Question understanding
  - “What is ISO?” vs “What is the ISO [of this camera]?”
- Semantic matching
  - “cost”, “price”, “bang for buck”, “expensive”, “cheap”
- Natural language answer generation
  - e.g. “This is great value for money” for question “Is this expensive?”
- High precision (>90%) requirement
- Data availability
  - “Will this suitcase fit in the overhead of an airplane?”
- Data quality
  - “Dimensions: 1x1x1 inches” for the question “How big is it” on a “Tripod page”

Learning Semantically Rich Representations

- Training examples: <question (q), relevant snippet (s_+), irrelevant snippet (s_-)> triples
- **Triplet network**

$$P(s^+ \text{ is relevant to } q) = \frac{1}{1+e^{-E(q)\cdot E(s^+)}}$$

Cosine Similarity = $cos(E(q), E(s^+))$



s^+: Product is very good for the given price

q: Is it expensive?

s^-: Auto focus is great

Results for Different Loss Functions

- Learn question and snippet representations to minimize the following loss functions:
  - Log loss [EMNLP 2015]
  - Siamese loss [Wang et al. 2014]

    $$\max\{0, \mathrm{M} - (\cos(E(q), E(s^+)) - \cos(E(q), E(s^-)))\}$$

  - Twin loss
- Metric: Precision at rank 1 (P@r1)
- Results:

$$-\cos(E(q), E(s^+))\} + \max\{0, \cos(E(q), E(s^-))$$

Loss function	Baseline	Log loss	Siamese loss	Twin loss
P@r1	56.8%	84.6%	96.1%	97.04%

[Wang et al 2014] Learning fine-grained image similarity with deep ranking, CVPR 2014.

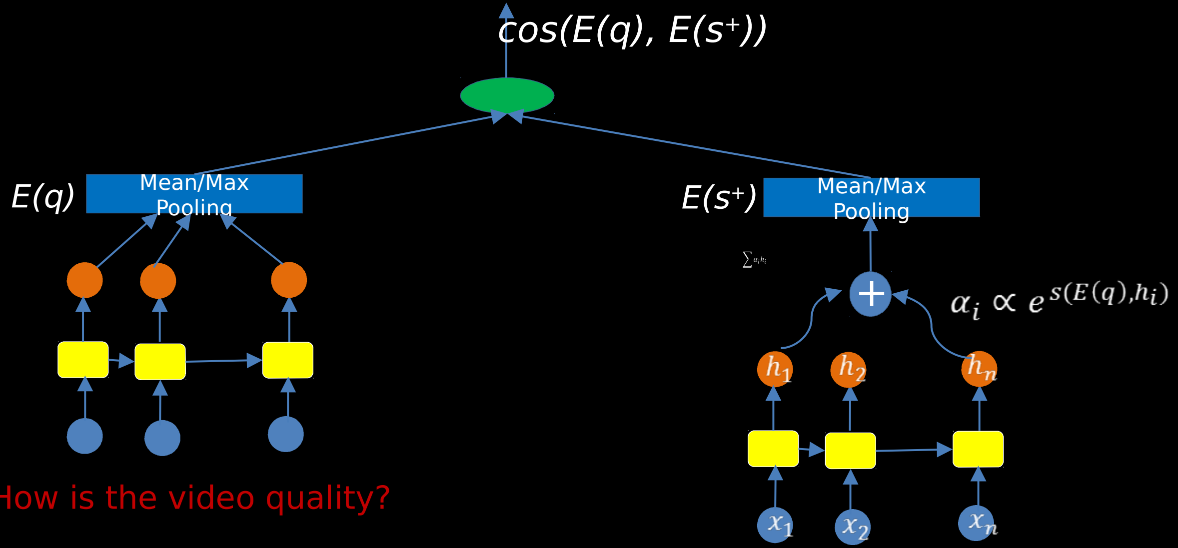
Qualitative Results

Question	Matching Snippet
Is this camera good for pictures at a basketball game?	Works great for **sports** photography
What is the price ?	This item **costs** $100.00. To see tax and shipping, add to cart
How big is it?	Item **dimensions**: 3 x 3.28 x 4.37 inches
How good is stabilization?	EVERY image came out **blurry** (and I held the camera still in a well-lit room).
Will it fit on Olympus air?	Fits very well the Olympus Air OA-01
How much weight can it hold?	Item weight: 2.2 pounds
What is the color of the paper on which the photo is printed?	the color of the camera and the pictures are great.

amazon

Learning Representations with Attention

- Only consider relevant portions of snippets when learning representations [Bahdanau et al. 2015]



q: How is the video quality?

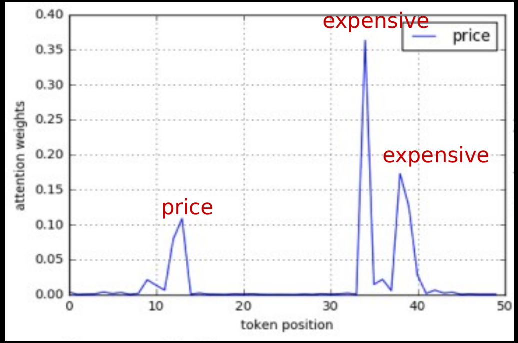
s^+: The camera has good video quality but the price is high

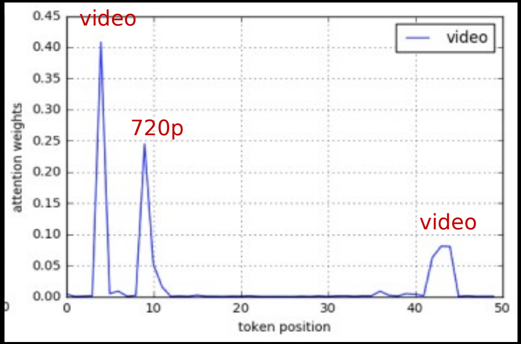
[Bahdanau et al. 2015] Neural machine translation by jointly learning to align and translate, ICLR 2015.

Highlighting Words with High Attention Weights

Review Statement:
The package is good video quality is good for 720p but the price is excellent for what you get - especially if you do not want all the whistles and bells of the more expensive gopro 2-4x more expensive and the quality of video is superb and great.

Questions:
Question1: What is the price?
Question2: How good is the video?





Outline

- Question Answering
- **Catalog Quality**
- Product Size Recommendations

Amazon’s Product Catalog

- Product catalog contains rich metadata on Amazon products
  - Title
  - Image
  - Description
- Must ensure that information i
  - Factually correct
  - Consistent
  - Adheres to pre-set guidelines

Title Defects

Title too long, Spam keywords

Email in Title

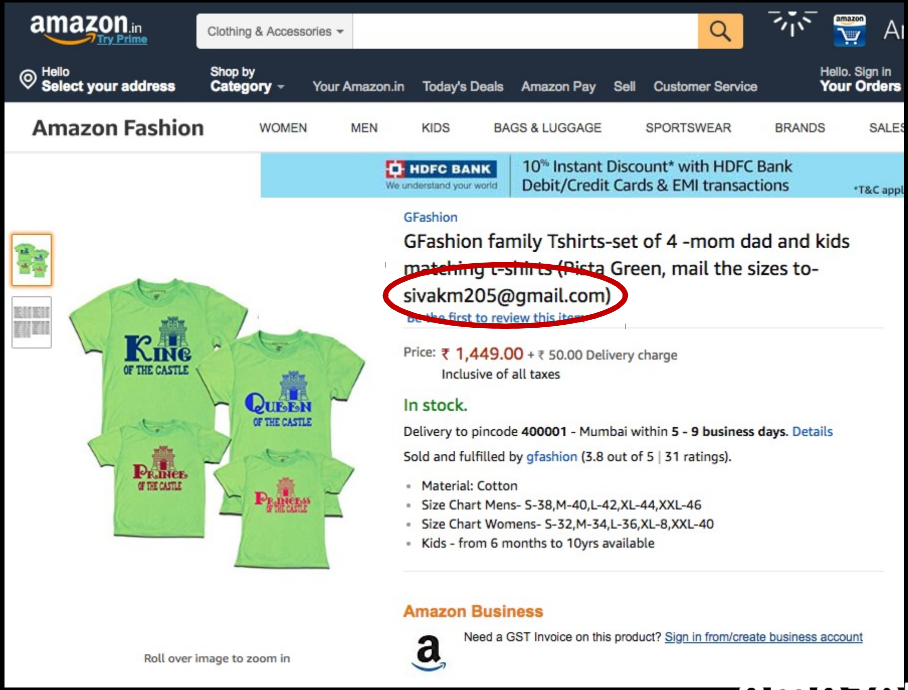
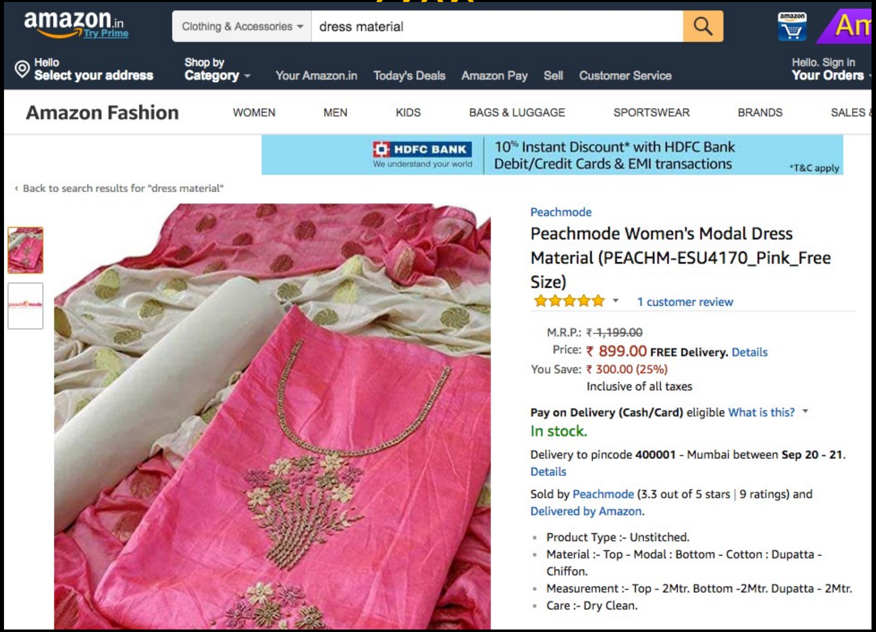


Image Defects

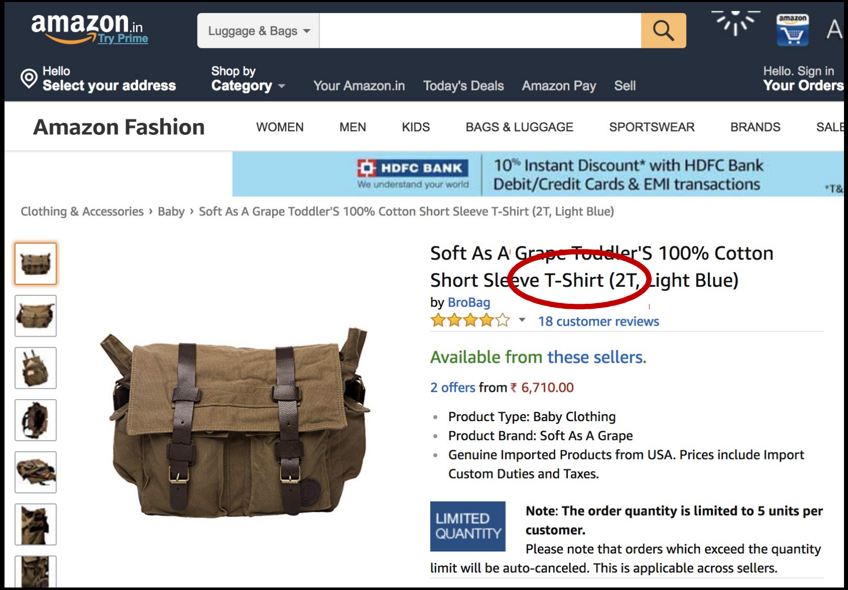
Background not white

Incomplete image, Improper crop

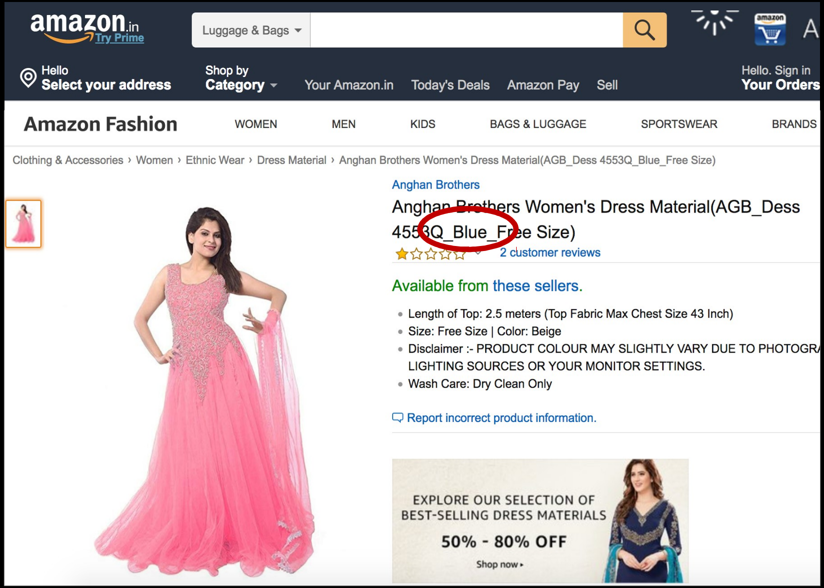


Product Attribute Mismatches

Category Mismatch



Color Mismatch



Text Attribute Extraction

Objective: Extract attributes like brand, color, size, model from product title and description.

Example:

Brand

Samsung Galaxy A8 Brand New Dustin hard Silicon Transparent black Color edge Back CaseCover

Color

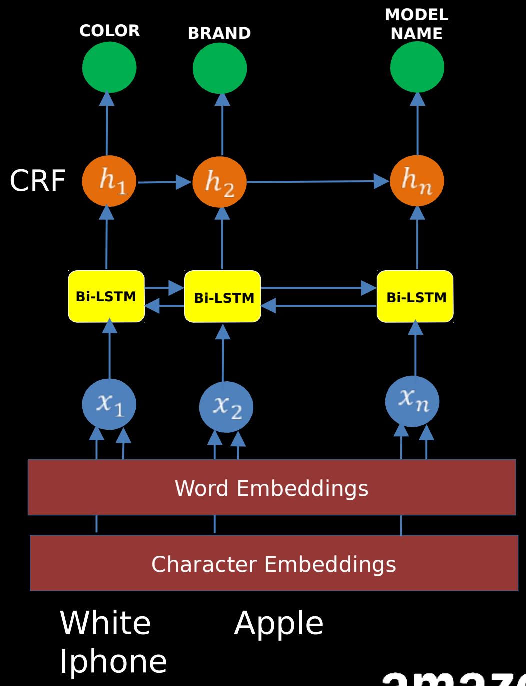
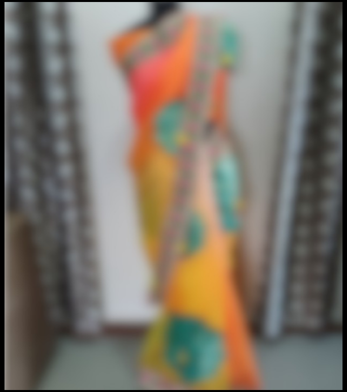


Image Classification/Attribute Extraction

Objective: Detect image defects, extract image attributes such as color, and classify images

Image Defects



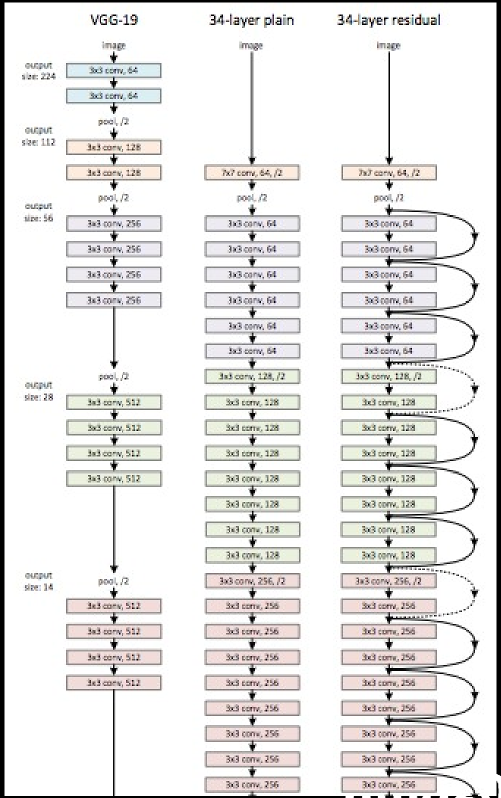
Poor Quality

Attribute Extraction

Color: Red

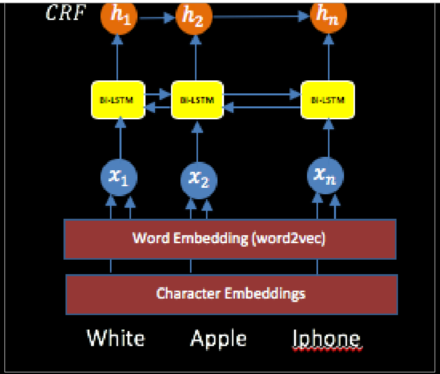
Category Classification

Category: T-Shirt

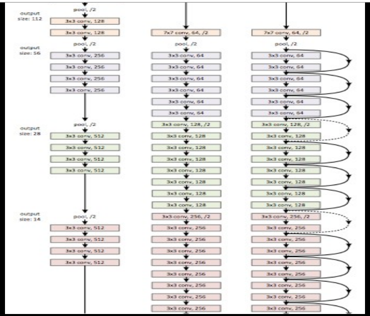


Mismatch Detection

Extractions from Text



Extractions from Image



Brand, Color, Category

Mismatch Logic

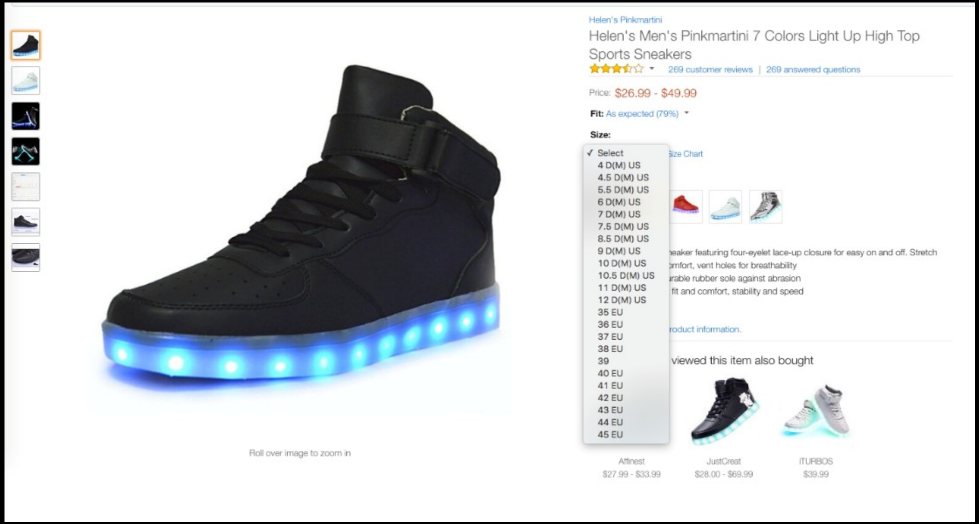
amazon

Outline

- Question Answering
- Catalog Quality
- **Product Size Recommendations**

Size Recommendation Problem

- Given a customer and product, recommend product size that would best fit the customer



Motivation

- No standardization of sizes across brands and locales for product categories such as shoes and apparel
- This leads to users making incorrect purchases, and then returning products
- Products belonging to shoe and apparel categories have high return rates due to fit issues
- Example:
  - Reebok size mapping convention: 6 = 15cm, 7 = 17cm, 8 = 21cm
  - Nike size mapping convention: 6 = 16cm, 7 = 18cm, 8 = 22cm

Key Challenges

- Scale: hundreds of millions of customers and products
- Data sparsity: bulk of users/products have very few purchases
- Cold start scenarios: new customers/products
  - User features: demographics (age, gender), location
  - Product features: catalog size, title, brand, product type
- Multiple personas: each customer *i* may involve multiple personas
  - E.g., family members sharing an account
  - Personas may have widely varying sizes

Our Approach

- Learn true (latent) size for each customer, product
  - True size for customer corresponds to the physical size of the customer (for shoes, it would be the feet size)
  - True size for product corresponds to it’s physical size
- Leverage past customer transactions $T = \{(i, j, y_{ij})\}$
  - y_{ij} takes ordinal values {small, fit, large}

	Adidas (9)	**Nike (8)**	**Reebok (8)**	**Nike (9)**
Customer 1	large		fit	**?**
Customer 2		small		fit
Customer 3	fit		small	**?**

Catalog size

Predict fit outcome

amazon

Our Approach (Contd)

- Notation
  - Latent size for customer i: s_i
  - Latent size for product j: t_j
  - Catalog size for product j: c_j

- Model likelihood of fit as a function of the difference between customer and product latent sizes

$$P\left(y_{ij} = fit\right) \propto f\left(s_i - t_j\right)$$

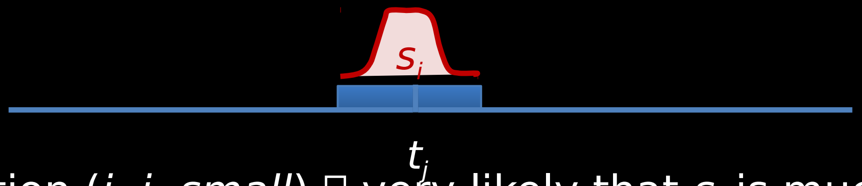
- Recommend product j with highest fit likelihood $P\left(y_{ij} = fit\right)$ to customer i

Bayesian Modeling Benefits

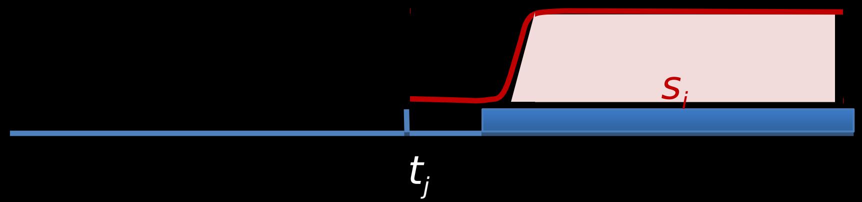
- Handles data sparsity by placing priors on latent size variables
- Models uncertainty in inferred latent sizes
  - Estimates posterior distribution of latent size variables
  - Fit probability is obtained by averaging over posterior size distribution
- Model can capture all the available data
  - Observations: transaction outcomes, customer and product features
  - Hidden variables: latent sizes, customer personas
- Efficient techniques for approximating posterior distributions of latent size variables

Intuition

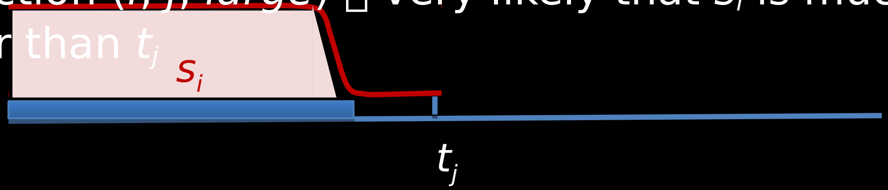
- Transaction (i, j, *fit*) very likely that s_i and t_j are close



- Transaction (i, j, *small*) very likely that s_i is much larger than t_j



- Transaction (i, j, *large*) very likely that s_i is much smaller than t_j



Data Likelihood

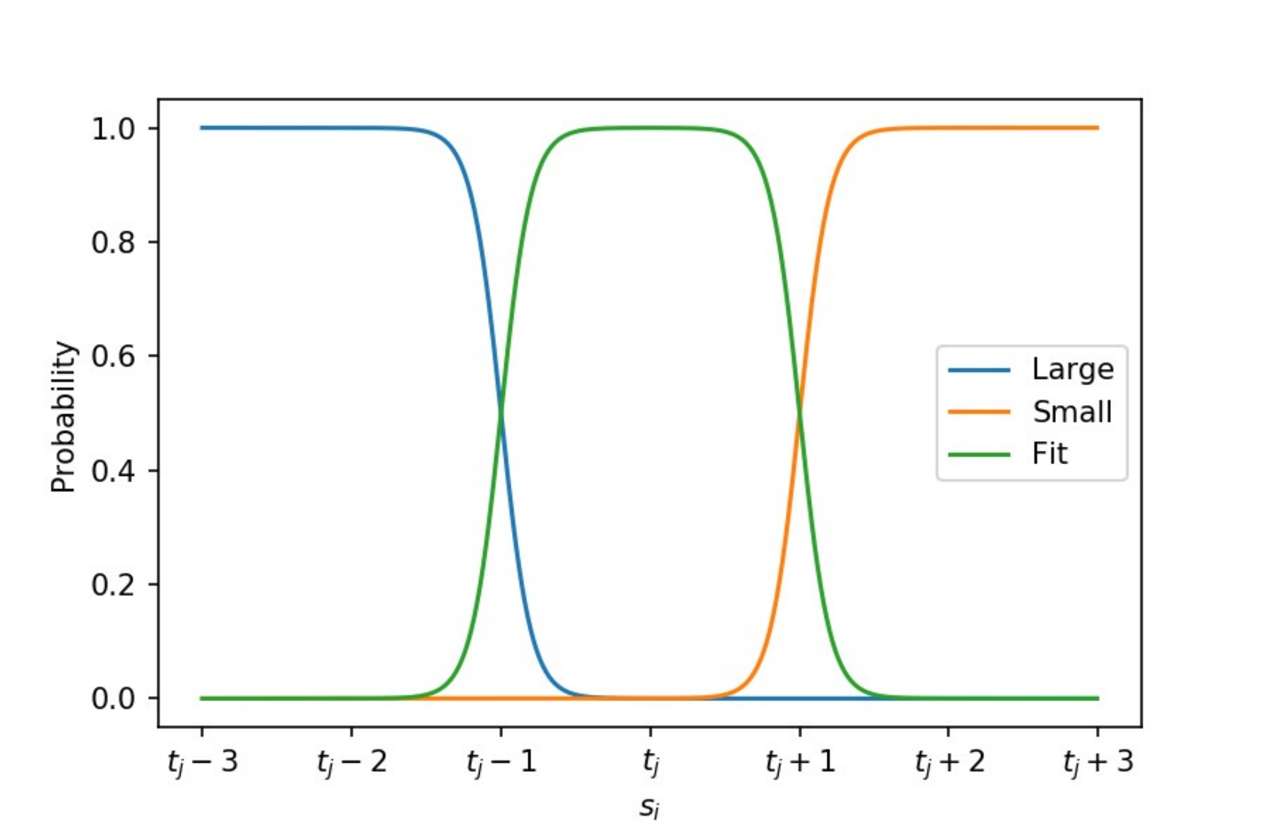
- $$P\left(y_{ij} = small \mid s_i, t_j\right) = \frac{1}{1+e^{-\alpha\left(s_i - t_j\right)+b_1}}$$

$$P\left(y_{ij} = fit \mid s_i, t_j\right) = \frac{1}{1+e^{\alpha\left(s_i - t_j\right)-b_1}} \cdot \frac{1}{1+e^{-\alpha\left(s_i - t_j\right)+b_2}}$$

$$P\left(y_{ij} = large \mid s_i, t_j\right) = \frac{1}{1+e^{\alpha\left(s_i - t_j\right)-b_1}} \cdot \frac{1}{1+e^{\alpha\left(s_i - t_j\right)-b_2}}$$

[Albert & Chib 1993] Bayesian analysis of binary and polytomous response data, JASA 1993.

Data Likelihood



Generative Model

for each customer *i*

for each customer i,

  draw latent size $s_i \sim N(\mu_i, \sigma_s^2)$

for each product j,

  draw latent size $t_j \sim N(c_j, \sigma_t^2)$

for each transaction $(i, j, y_{ij}) \in T$,

  select y_{ij} = *small* with probability $P(y_{ij} = small|.)$

  select y_{ij} = *fit* with probability $P(y_{ij} = fit|.)$

  select y_{ij} = *large* with probability $P(y_{ij} = large|.)$

large|.)

Bayesian Inference

- Let β be the vector of latent sizes

- Let β be the vector of latent sizes

$$\beta = \left(s_1, \ldots, s_c, t_1, \ldots, t_p, 1\right)^T$$

- Posterior distribution

- Posterior distribution

$$P(\beta|T) \propto P(T|\beta) \cdot P(\beta)$$

$$\propto \prod_{(x,y)\in D} \frac{e^{y\beta^T \cdot x}}{1+e^{\beta^T \cdot x}} \cdot \prod_i N(s_i|\mu_i, \sigma_s) \cdot \prod_j N(t_j|c_j, \sigma_t)$$

- Not available in closed form due to logistic likelihood terms and Normal priors

0/1

- Not available in closed form due to logistic likelihood terms and Normal priors

amazon

Polya-Gamma Augmentation [Polson et al. 2013]

- Introduce Polya-Gamma latent variable for every
- Define the joint likelihood distribution
- In [Polson et al. 2013], it is shown that

- Introduce Polya-Gamma latent variable $w \sim PG(0,1)$ for every $(x, y) \in D$

- Define the joint likelihood distribution

$$P(w, y|x, \beta) = \frac{1}{2} e^{\left(y - \frac{1}{2}\right) \cdot (\beta^T \cdot x) - w \cdot \frac{\left(\beta^T \cdot x\right)^2}{2}} \cdot P(w)$$

- In [Polson et al. 2013], it is shown that

$$\int_0^{\infty} P(w, y|\beta, x) dw = \frac{e^{y\beta^T \cdot x}}{1 + e^{\beta^T \cdot x}}$$

[Polson et al. 2013] Bayesian inference for logistic models using Polya-Gamma latent variables, JASA 2013.

Polya-Gamma Augmentation (Contd)

- Let W be the set of Poly-Gamma variables w for $(x, y) \in \boldsymbol{D}$

Let W be the set of Poly-Gamma variables for

$$P(\beta|\boldsymbol{D}) \propto \int P(W, \boldsymbol{D}|\beta) \cdot P(\beta) dW$$

- Approximate the augmented joint distribution $P(W, \boldsymbol{D}|\beta) \cdot P(\beta)$

Approximate the augmented joint distribution

$$\prod_{(x,y)\in\boldsymbol{D}} \frac{1}{2} e^{\left(\left(y-\frac{1}{2}\right)\cdot(\beta^T\cdot x) - w\cdot\frac{(\beta^T\cdot x)^2}{2}\right)} \cdot P(w) \cdot \prod_i N(s_i|\mu_i, \sigma_s) \cdot \prod_j N(t_j|c_j, \sigma_t)$$

Gibbs Sampling Algorithm

- Conditional distribution of

- Conditional distribution of w_i

$$\max\{0, M - (\cos(E(q), E(s^+)) - \cos(E(q), E(s^-)))\}$$

- Conditional distribution for

- Conditional distribution for β_j

))

Predictive Distribution

- Let r samples be drawn from the posterior of β
- Let the l^{th} sample be $\beta^l = (s_1^l, \ldots, s_c^l, t_1^l, \ldots, t_p^l)$

$$P(y_{ij} = fit|\boldsymbol{D}) = \int P(y_{ij} = fit|\beta) \cdot P(\beta|\boldsymbol{D}) d\beta$$

$$\approx \frac{1}{r} \sum_{l=1}^{r} \frac{1}{1+e^{\alpha(s_i^l - t_j^l) - b_1}} \cdot \frac{1}{1+e^{-\alpha(s_i^l - t_j^l) + b_2}}$$

Experimental Results

- Consider 6 real-life shoes datasets with between 10M and 33M transactions
- Baseline model
  - Product size $t_j = c_j$
  - Customer size s_i = Average size of products purchased by customer
  - Logistic regression model with feature (s_i-t_j) to predict outcome
- Bayesian Logit model
  - Predict outcome with highest probability
- Performance metric: weighted AUC
- Results (% improvements over baseline)

Dataset	A	B	C	D	E	F
Bayesian	17.71	18.28	19.7	25.78	20.22	19.42

Leveraging Customer and Product Features

- Means of latent size priors are obtained by performing regression over customer (f_i) and product (g_j) features [AC 2009]

for each customer *i*,

- **Gene**

for each customer i,

  draw latent size $s_i \sim N(w_f \cdot f_i, \sigma_s^2)$

for each product j,

  draw latent size $t_j \sim N(w_g \cdot g_j, \sigma_t^2)$

for each transaction $(i, j, y_{ij}) \in T$,

  select y_{ij} = *small* with probability $P(y_{ij} = small|.)$

  select y_{ij} = *fit* with probability $P(y_{ij} = fit|.)$

  select y_{ij} = *large* with probability $P(y_{ij} = large|.)$

select y_{ij} = *large* with probability $P(y_{ij}$

large|.)

- Perform least squares regression to learn parameters w_f and w_g from customer and product size samples

[Agarwal & Chen 2009] Regression based latent factor models, KDD 2009.

Incorporating Customer Personas

- Latent size for persona *k* of customer *i*: s_{ik}
- Latent variable containing persona involved in transaction (i, j, y_{ij}): z_{ij}
- **Generative model:**

for each customer *i*,

for each customer *i*,

draw persona distribution $\theta_i \sim \mathrm{Dir}(\alpha)$

for each persona *k* draw latent size $s_{ik} \sim N(w_f \cdot f_i, \sigma_s^2)$

for each product *j*,

draw latent size $t_j \sim N(w_g \cdot g_j, \sigma_t^2)$

for each transaction $(i, j, y_{ij}) \in T$,

draw persona $z_{ij} \sim \mathrm{Mult}(\theta_i)$

select $y_{ij} = small, \ldots$ with probability $P(y_{ij} = small \mid z_{ij}), \ldots$

$small|z_{ij}), \ldots$

- Gibbs Sampling algorithm can be extended to draw z_{ij} samples

amazon

Summary

- Learning semantically rich data representations improves predictive accuracy of models
  - Question answering, catalog quality, product search, product recommendations, …
- **Deep Learning** to learn embeddings
  - Allows semantic matching between questions and snippets
  - Loss functions like Siamese loss that aim to maximize difference in class scores perform better
- **Probabilistic Graphical Models** to learn latent sizes
  - Priors handle data sparsity, prevent overfitting
  - Posteriors model uncertainty in data
  - Leverage all the available signals

Summary

- AI, ML at the core of Amazon’s business
  - Address quality, recommendations, product search, catalog quality, question answering, demand forecasting, …
- Learning semantically rich data representations improves predictive accuracy of models
  - Question answering, catalog quality, product search, …
- Deep learning is an effective tool to learn semantically rich data representations