PEQ : An Explainable, Specification-based, Aspect-oriented Product Comparator for E-commerce

Abhishek Sikchi
Indian Institute of Technology
Kharagpur, India
sikchi70@gmail.com

Pawan Goyal
Indian Institute of Technology
Kharagpur, India
pawang.iitk@gmail.com

Samik Datta
Flipkart
India
samik.datta@flipkart.com

ABSTRACT

While purchasing a product, consumers often rely on specifications as well as online reviews of the product for decision-making. While comparing, one often has in mind a specific aspect or a set of aspects which are of interest to them. Previous work has used comparative sentences, where two entities are compared directly in a single sentence by the review author, towards the comparison task. In this paper, we extend the existing model by incorporating the feature specifications of the products, which are easily available, and learn the importance to be associated with each of them. To test the validity of these product ranking measures, we comprehensively test it on a digital camera dataset from Amazon.com and the results show good empirical outperformance over the state-of-the-art baselines.

Keywords

Comparison mining; Decision support systems

1. INTRODUCTION

Being able to compare among alternative product models has a direct role to play in decision making by potential buyers. However, it is often hard to arrive at a conclusion of a good choice due to the huge number of brands and models. Websites such as PriceGrabber[1] compare prices of products whereas DPReview[2], CameraDecision[3], etc. give comparative tables of feature specifications as documented by the manufacturer but they do not provide insight into how those features translate into usefulness during the actual use, which can be very specific. On the other hand, user reviews often describe first-hand experiences of using the product but the major drawbacks are the need to read many reviews to get an estimate of agreement between users and the need to manually compare different products. The goal of our work is to fill-in this space and provide a means of comparing products of the same type (e.g., competing cameras). Some of the previous works have focused on comparative sentences in the reviews. While Kessler and Kuhn [3, 4] use approaches such as semantic role labeling and structure alignment to detect product comparison sentences, they do not provide a model to use these for actual comparison. Tkachenko and Lauw [6], on the other hand, takes the product comparison sentences as input and gives a generative model for product comparisons.

However, relying on the language alone can be inapt. For instance, sentences shown in Table 1 differ a lot from other sentences used to compare the same aspects of digital cameras due to the unique choice of words by their reviewers or simply because of typing errors, as in the sentence s_2. On the other hand, one can well hypothesize that in an ideal situation, a rational consumer would opt for a product which has a better set of specifications – a theory supported by psychologists and economists alike. These specifications or attribute values do play a role in determining the intrinsic quality of the product. Specifications have earlier successfully been used along with the product reviews to generate augmented specifications [5].

We propose a model, *PEQ*, that integrates the language along with the products' attributes in order to model comparisons made on the basis of comparative sentences about two entities. Understanding such comparisons have several applications. From the manufacturers' perspective, they could better understand the relative strength or weakness of their products, and hence develop better products. From the consumers' perspective, they could exercise better, informed purchasing decisions by comparing the various features of certain kind of products.

2. THE EXPLAINABLE CHOICE MODEL

The *psychological* theory of choices, as illustrated in [1] and the references therein, posits that the outcome of a product comparison (e.g., digital cameras), hinged on an aspect (e.g., ergonomics), is arrived at by inspecting the products' features. The *economic* theory of choices, as presented in [7] and the references therein, additionally assumes *rational* decision making on the part of the buyer, manifested in the form of *utility* maximization – a buyer and aspect dependent comparison of products' features vis-à-vis prices.

In recent times, with the proliferation of user-generated online product review corpora documenting outcomes of thousands of product comparisons along with accompanying explanatory text, one further wishes to augment the tradi-

[1]https://www.pricegrabber.com
[2]https://www.dpreview.com/products/compare/cameras
[3]https://www.cameradecision.com

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983901

Aspect	Sample sentences
Functionality	s_1 : There's a reason nikon d7000 has almost twice as many buttons/knobs/what-have-you than the D5000: you are meant to use them!
Form factor	s_2 : The TL225 also wieghs a half-ounce more then the TL220 due to the choice of displays.
Image quality	s_3 : Then I bought Canon 40D, and I got the same problem, I even tried Nikon D90, the red dots were reduced, but still very noticeable. s_4: I'm still experimenting, but my gut feeling is that if you were to compare a 16MP DX crop from the center of the D800e frame versus the 16MP D7000 DX at very slow shutter speeds, that the D7000 would have less blur from mirror slap.

Table 1: Sample Comparative Sentences about Digital Cameras

tional choice models with language models for the explanatory text. This greatly aids the interpretability of such comparisons and enables novel decision support systems for e-commerce. Herein, we extend the work presented in [6].

Given a corpus of sentences containing pairwise product comparisons hinged on a specific aspect, our task is twofold: first, we seek to recover the partial order inducing comparative relations on the universe of cited products, with respect to the aspect under consideration, and, second, we infer the outcome of the comparison for every sentence in the corpus. In addition to the above mentioned corpus, we also make use of corpora containing product specifications.

Formally, the review corpus, $\mathcal{S}$, consists of sentences S that compare a pair of products each, $h(S) \in \mathcal{P}$ and $t(S) \in \mathcal{P}$, where $h(S)$ is the product appearing first in S, and $t(S)$ is the product appearing later. The subset of $\mathcal{S}$ involving the pair of products, P_i and P_j, is denoted by S_{ij}. For pairs that are *never* compared in any sentence, $S_{ij} = \phi$.

In what follows, we first model the outcome of product comparisons, relying solely on their specifications, and then model the language one would use to explain those comparisons.

2.1 Intrinsic Goodness

For every pair of products, $(P_i, P_j) \in \mathcal{P} \times \mathcal{P}$, we intend to infer their comparative relationship, with respect to an aspect $A \in \mathcal{A}$, *solely* based on their specifications. To this end, we endow each product $P \in \mathcal{P}$ with an intrinsic *goodness* score, $G_P \in \mathbb{R}$. The outcome of the aforementioned comparison, thus, reduces to the relative values of G_{P_i} and G_{P_j}, modulo an uncertainty that models lack of consensus. In other words, for every sentence $S \in \mathcal{S}$ the outcome of the comparsion is influenced by the relative values of $G_{h(S)}$ and $G_{t(S)}$.

We model the intrinsic goodness to closely resemble the utility, sans an explicit negative coefficient on the price. As an example, while selecting the best digital camera with respect to picture quality, a rational buyer would emphasize relevant specifications like number of megapixels, size of the sensor etc. over and above less relevant features like weight, weather proofing etc. We model this as a linear combination of the specifications including price, with an aspect-dependent weight vector W_A. The intrinsic goodness of each product is thus measured by:

$$G_P = W_A^T X_P \tag{1}$$

where $X_P \in \mathcal{X}$ is the vector encoding the product's specifications. Although we do not explicitly constrain the corresponding component for price in W_A to be negative, in our experiments it often turns out to be so, giving the intrinsic goodness score an utility-theoretic interpretation. An extension that models the non-linearity is discussed in 3.1.

2.2 Rational Choice

For every sentence, $S \in \mathcal{S}$, we let the binary random variable $C_S \in \{0, 1\}$ denote the direction of the comparison, with $C_S = 0$ implying that $h(S)$ is favored over $t(S)$, and vice-versa. The random variable C_S is further assumed to depend on the relative values of the intrinsic goodnesses of the products involved, $G_{h(S)}$ and $G_{t(S)}$. This leads to the *rational* choice model:

$$\begin{aligned}\Pr\{C_S = 0; X_{h(S)}, X_{t(S)}\} &= \Pr\{h(S) \succ_A t(S)\} \\ &= \sigma(\vartheta \times (G_{h(S)} - G_{t(S)}))\end{aligned} \tag{2}$$

Where $\succ_A \subset \mathcal{P} \times \mathcal{P}$ is the partial order induced by the comparison with respect to the aspect $A \in \mathcal{A}$, and ϑ is a tunable parameter.

2.3 Language Model

The outcome of the comparison between $h(S) \in \mathcal{P}$ and $t(S) \in \mathcal{P}$, as embodied in C_S, influences the *language* of the sentence S that we endeavour to model in this section.

We follow the conventions introduced in [6], and posit *two* language models, θ_0 and θ_1, that models the aspect specific sentiment conveying words. We emphasise that our language models are not distributions over words. They are, instead, distributions over *features*, words anchored with respect to $h(S)$ and $t(S)$. We refer the reader to [6] for a detailed exposition.

3. THE *PEQ* GENERATIVE PROCESS

We now describe the generative process that underlies PEQ.

Choice Model. For every comparative sentence, $S \in \mathcal{S}$, that compares two products $h(S)$ and $t(S)$, we first sample the outcome, C_S, according to the choice model $C_S \sim \texttt{Bernoulli}(\sigma(\vartheta \times (G_{h(S)} - G_{t(S)}))$.

Language Model. Depending on the orientation of the choice, we next pick one of $\theta_0, \theta_1 \sim \texttt{Dir}(\alpha)$, from whence the lingustic features, f, is sampled: $f \sim \texttt{Mult}(\theta_{C_S})$.

In line with [6], the joint probability is expressed as follows:

$$\Pr\{F(S) \mid C_S = c\} = \prod_{f \in F(S)} \Pr\{f \mid \theta_c\} \tag{3}$$

Where $F(S)$ enumerates the features present in S.

Armed with the generative process, the comparison outcome for each sentence S, thus, can be obtained as the posterior distribution of C_S, and the partial order among the products can be recovered by comparing the values of G_P. We illustrate the learning algorithm in the next section. The aspect-specific weight vector, W_A, comparison

outcomes $C_S, \forall S \in \mathcal{S}$ and the language models $\theta_c, \forall c \in \{0,1\}$ constitute the hidden variables. The observables comprise of the features $F(S), \forall S \in \mathcal{S}$ and the product specifications $X_P, \forall P \in \mathcal{P}$.

3.1 Learning W_A

Given an assignment of $C_S, \forall S \in \mathcal{S}$, we employ RankSVM, as introduced in [2], to recover W_A. It is a variant of the celebrated Support Vector Machine that learns W_A and respects the pairwise preference constraints posed by $C_S, \forall S \in \mathcal{S}$ in a large-margin setting. The input to this phase are the product specifications, X_P, and the orientation of choices, C_S. Note that extension to the non-linear setting is trivial via kernels. We leave the possibility of augmenting RankSVM to handle chance-constraints, such as those specified by $\Pr\{C_S\}$, to a future work. Once W_A is learnt, we appeal to Equation 1 for deriving the intrinsic goodness of each product.

3.2 Learning C_S

For learning $C_S, \forall S \in \mathcal{S}$, we resort to Gibbs sampling. Following [6], we employ a *collapsed* Gibbs sampler that integrates out θ_0, θ_1 analytically. Letting $\{\mathcal{S}_c \mid \forall c \in \{0,1\}\}$ denote a partition of $\mathcal{S}$, the conditional probabilities can be expressed as:

$$\Pr\{\mathcal{S}_c \mid c; \alpha\} = \frac{\Gamma(\alpha F)}{\Gamma^F(\alpha)} \frac{\prod_{f=1}^{F} \Gamma(\alpha + n(f,c))}{\Gamma(\alpha F + \sum_{f=1}^{F} n(f,c))} \tag{4}$$

Where $n(f,c)$ denotes the frequency of the feature f in $\mathcal{S}_c$, and F denotes the size of the feature vocabulary. We refer the reader to [6] for a detailed derivation.

Fixing $G_P, \forall P \in \mathcal{P}$ (i.e., by fixing W_A), and $C_{S'}$ for every sentence in $\mathcal{S}$ other than S itself, we sample C_S from the following posterior distribution:

$$\begin{aligned} &\Pr\{C_S = c \mid \mathcal{C}_{\neg S}, \cdots\} \\ &\propto \Pr\{C_S \mid G_{h(S)}, G_{t(S)}) \times \prod_{c \in \{0,1\}} P(\mathcal{S}_c \mid c; \alpha) \\ &\propto \exp\{-C_S \vartheta W_A^T (X_{h(S)} - X_{t(S)})\} \times \prod_{c \in \{0,1\}} P(\mathcal{S}_c \mid c; \alpha) \end{aligned} \tag{5}$$

We repeat the process until convergence.

4. EXPERIMENTS

We obtain the corpus of comparative sentences $\mathcal{S}$ from Tkachenko and Lauw [6] which contains reviews from the Digital Cameras category of Amazon. The dataset has comparative sentences extracted from reviews, in which each sentence is annotated with three key information: whether a sentence is comparative, the entities being compared, and the aspect of interest. In total, the number of products being compared within extracted sentences is 180. The four aspects used are functionality, form factor, image quality and price, and they respectively have 457, 78, 129, and 165 comparative sentences. Each aspect is a distinct instance of the problem. The distributions between the two classes (whether the head entity or the tail entity is favored) are relatively well-balanced. We used 500 iterations of the Gibbs sampler to stablize the *PEQ* model. The results are reported in 10-fold cross-validation settings.

4.1 Results

We perform experiments under both supervised and unsupervised settings. Where in the unsupervised setting, only the f's are observed, in the supervised setting, we would consider some C_s variables (corresponding to a subset of labeled sentences) to also have known outcomes. This effects in grouping together sentences of the same label, which would then influence the respective feature distributions, θ_0 and θ_1.

Supervised Configuration. In supervised setting, all the competing algorithms are given a set of labeled (training) data, which is 50% of the total dataset, and the rest as unlabeled (test) data. For sentence-level classification accuracy, each algorithm is required to identify the favored entity for each comparative sentence in the test data, which is essentially a binary classification problem. To measure the performance of an algorithm, we calculate the fraction of correctly classified sentences (over the total number of sentences in the test set). We measure our results against RankSVM model and the CompareGem model formulated in [6] as baseline.

Aspect	CompareGem	RankSVM	PEQ
Functionality	86.8	87.5	**90.2**
Form Factor	72.5	**88.6**	85.7
Image Quality	**68.8**	67.2	64.7
Price	66.7	67.6	**70.5**

Table 2: Classification accuracy for supervised model

For entity ranking accuracy, the majority vote, considering all the sentences that mention a pair is used as crowdsourced benchmark. Therefore, this benchmark reflects how users in general rank these entities.

Tables 2 and 3 show the classification and ranking accuracies for all the competing models in the supervised settings. We see that *PEQ* always performs better than CompareGem in terms of ranking accuracy. For classification accuracy, it performs better than CompreGem in 3 out of 4 aspects. For form-factor, RankSVM outperforms PEQ in both classification and ranking accuracy with a slight margin, and CompareGem by a huge margin. This might be indicative of the fact that specifications are very good indicative for product ranking for the 'Form-factor'.

Aspect	CompareGem	RankSVM	PEQ
Functionality	87.4	88.2	**88.3**
Form Factor	82.7	**90.2**	89.3
Image Quality	75.5	74.4	**76.6**
Price	74.0	66.3	**74.5**

Table 3: Ranking accuracy for supervised model

Unsupervised Configuration. In the unsupervised configuration, no labeled data is used as input. The task resembles clustering into two clusters, rather than classification. We can still use the labels to evaluate this clustering, by computing purity instead. As a baseline, we compare against CompareGem only, since RankSVM cannot be used in the absence of any labels. CompareGem has already been compared to existing methods for both clustering and ranking tasks and found to be superior in most cases. Our model

outperforms CompareGem in almost all cases except for the aspect 'price' as shown in Tables 4 and 5.

Aspect	CompareGem	PEQ
Functionality	70.2	73.6
Form-factor	65.6	65.8
Image-quality	62.5	63.7
Price	57.0	56.4

Table 4: Classification accuracy (purity) for unsupervised model

Aspect	CompareGem	PEQ
Functionality	64.6	68.2
Form-factor	64.4	67.4
Image-quality	59.6	61.7
Price	56.0	54.7

Table 5: Ranking accuracy for unsupervised model

4.2 Feature Analysis

We study the features which are the most discriminative between the two classes, and play important role in the supervised model. A discriminative feature f is one whose conditional probability $\Pr\{c|f\} > 0.8$.

We notice that for functionality, the top feature for $c = 0$, is "h(S)_from_t(S)", while that for $c = 1$ is "from_h(S)_t(S)". Although it involves the same word "from", the different relative positions with respect to the entities make a difference and this underlines the importance of the bag-of-features model. Example sentences with these features are shown below.

- ($c = 0$) On my *D5100*, LiveView autofocus typically took half a second in lower-light conditions, which is no match for the viewfinder, but a huge improvement from the *D5000*.
- ($c = 1$) I like the consistency in the controls as far as moving from a *D7000* to a *D600*.

Other than their relative positions, the actual words that help make up a feature also matter. In Figure 1, we present the most frequently found words in sentences assigned to each class, for the aspect 'Form factor'. Interestingly, we see contrasting features such as "lighter" (for $c = 0$) vs. "heavier" (for $c = 1$). Similarly, for price, we find "less" (for $c = 0$) vs. "more" (for c = 1).

For the specifications which were used as features in the RankSVM model, it was observed that attributes like 'resolution' and 'maximum shutter priority' had more weight for the aspect 'Image quality' whereas 'dimensions' had more relative weight for aspect 'Form factor'.

5. CONCLUSION

In summary, we study the problem of comparative review mining and propose *PEQ*, a new integrated model that learns the products being favored by modelling the intrinsic goodness of the entity as well as the comparative relations at the level of entity pairs. We use Gibbs sampling to infer the sentence-level preferences and RankSVM model to learn the intrinsic goodness in terms of weighted sum of the product specifications. We validate our model on Amazon reviews and compare against existing models and baselines. The model performs better in most cases than the baselines CompareGem and RankSVM. We conclude that the use of the specifications of an entity, which are easily available on the internet leads to a better understanding in comparison of products. This validates the fact that the intrinsic goodness of products is, to a great extent, determined by the values of its attributes. The experiments convincingly show the helpfulness of our model in both supervised and unsupervised configurations.

Figure 1: Word Clouds showing different distributions for the aspect Form factor

6. REFERENCES

[1] D. Görür, F. Jäkel, and C. E. Rasmussen. A choice model with infinitely many latent features. In *Proceedings of the 23rd international conference on Machine learning*, pages 361–368. ACM, 2006.

[2] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[3] W. Kessler and J. Kuhn. Detection of product comparisons – how far does an out-of-the-box semantic role labeling system take you? In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1892–1897, Seattle, USA, October 2013. Association for Computational Linguistics.

[4] W. Kessler and J. Kuhn. Structural alignment for comparison detection. In *Proceedings of the 10th Conference on Recent Advances in Natural Language Processing (RANLP 2015)*, pages 275–281, Hissar, Bulgaria, September 2015.

[5] D. H. Park, C. Zhai, and L. Guo. Speclda: Modeling product reviews and specifications to generate augmented specifications. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 837–845, 2015.

[6] M. Tkachenko and H. W. Lauw. Generative modeling of entity comparisons in text. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 859–868. ACM, 2014.

[7] G. Zhao, M. L. Lee, W. Hsu, and W. Chen. Increasing temporal diversity with purchase intervals. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 165–174. ACM, 2012.