Product Phrase Extraction from e-Commerce Pages

Artem Vovk
avovk@google.com
Google
Mountain View, CA

Dmitrii Tochilkin*
dmitry.tochilkin@skolkovotech.ru
Skolkovo Institute of Science and Technology
Russia

Pradyumna Narayana
pradyn@google.com
Google
Mountain View, CA

Kazoo Sone
sone@google.com
Google
Mountain View, CA

Sugato Basu
sugato@google.com
Google
Mountain View, CA

ABSTRACT

Analyzing commercial pages to infer the products or services being offered by a web-based business is a task central to product search, product recommendation, ad placement and other e-commerce tasks. What makes this task challenging is that there are two types of e-commerce product pages. One is the single-product (SP) page where one product is featured primarily and users are able to buy that product or add to cart on the page. The other is the multi-product (MP) page, where users are presented with multiple (often 10-100) choices of products within a same category, often with thumbnail pictures and brief descriptions — users browse through the catalogue until they find a product they want to learn more about, and subsequently purchase the product of their choice on a corresponding SP page. In this paper, we take a two-step approach to identifying product phrases from commercial pages. First we classify whether a commercial web page is a SP or MP page. To that end, we introduce two different image recognition based models to differentiate between these two types of pages. If the page is determined to be SP, we identify the main product featured in that page. We compare the two types of image recognition models in terms of trade-offs between accuracy and latency, and empirically demonstrate the efficacy of our overall approach.

CCS CONCEPTS

• **Information systems → Computational advertising**; **Content match advertising**; **Information extraction**; *Structured text search.*

KEYWORDS

information extraction; web page classification; text classification; neural networks; natural language processing; computer vision

*This work was conducted while doing internship at Google.

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316608

ACM Reference Format:
Artem Vovk, Dmitrii Tochilkin, Pradyumna Narayana, Kazoo Sone, and Sugato Basu. 2019. Product Phrase Extraction from e-Commerce Pages. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3308560.3316608

1 INTRODUCTION

The total volume of e-commerce sales in the US is estimated to be around $500 billion annually today and it is still showing double-digit growth year-over-year [5]. In this growing e-commerce domain, Natural Language Processing (NLP) is playing an increasing role in different types of tasks. Some representative tasks include: (a) Product Matching, where the description of an item offered by the seller has to be automatically mapped to the appropriate product listing category — the NLP challenges here include handling heterogeneous ways of describing the same data (ranging from unstructured sparse text to rich structured text), erroneous information extraction leading to product duplicates, bundling multiple different products together in the same offer, etc., which have been recently tackled by deep neural network models [17]; (b) Automatic title generation for product pages, where titles on product browse pages have to be generated automatically — the main NLP challenge here is handling multiple languages, which has been tackled recently by sequence to sequence neural networks [13]; (c) Review summary and user profile generation — the core NLP task here is extracting representative sentences from product reviews and user profiles, for which recently neural networks have been used for aspect extraction and embedding [15]; and (d) Sentiment analysis from product review data, where the challenge lies in detecting the polarity of user sentiments in reviews by handling ambiguous text at different levels (e.g., sentence or paragraph level) — different machine learning models like Naive Bayes, Random Forest and Support Vector Machines have been applied on this task and compared [9]. In addition, on the advertising front, Pryzant et. al (2018) recently used different neural network modeling approaches (e.g., direct residualization, convolutional adversarial techniques) to identify words in search advertising text that can influence users' click behavior, while accounting for confounders [16].

One area that has not been well studied to the best of the authors' knowledge is deep understanding of the content generated by advertisers, such as identifying products and services offered in different types of e-commerce web pages. There are billions of

commercial web pages, offering a wide variety of products ranging from apparel goods to real estate, and billions of dollars are spent by consumers while interacting with these pages. As the market and the number of items offered on e-commerce platforms grows, it is becoming increasingly important to have a good understanding of the products being offered on a web page. In recent work, Zheng et al. [20] automatically extract product attributes from natural language text given a product name — they extract the relevant attributes from product profiles using bidirectional LSTM models for sequence tagging and CRF models for enforcing tagging consistency. A similar modeling approach using LSTM-CRF had been used earlier by Kozareva et al. [12] for extracting product brand name, model, etc. from text segments — in this case, however, the authors analyzed shopping queries for this task, not product pages.

In our work we distinguish between two kinds of e-commerce web pages: single-product pages (or sometimes called leaf pages) and multi-product pages (or sometimes called non-leaf pages). In single-product (SP) pages, the page's primary focus is on a single, specific product, typically with a large image of a product that is accompanied by product descriptions and user reviews, with an option to purchase the item by adding it to the shopping cart. In multi-product (MP) pages, the page's primary focus is to give users multiple choices of products that belong to the same category, so that users can browse through wide selection of products until they find an item that interests them, at which time the user can visit the corresponding SP page with more details of the chosen product and possibly make a purchase.

In SP pages, identification of the product phrase is relatively straightforward, although there are still challenges, e.g., similar products featured on the same page can easily confuse any product phrase identification algorithm. In MP pages, identification of the exact products being offered is less useful as the advertisers' primary intention usually is to show users a collection of similar products, so that users can make an informed decision.

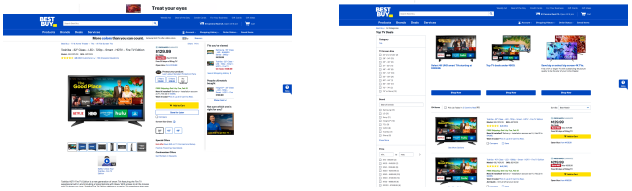
The classification of these two page types is of significant importance in e-commerce tasks, as it relates to how users can be matched to the commercial contents. For example, if the user has a clear intent of looking for a specific product, it makes sense for an internet search engine or a merchant to match them with a SP page for that product; whereas when the user is still in exploration or early phase of shopping, perhaps it makes more sense to match them with a MP page.

In this paper, we introduce three different approaches to classify product page type using different features and models. The following sections in the paper describe these approaches in detail, and outline the experiments we performed to analyze the trade-offs between various techniques.

2 APPROACH

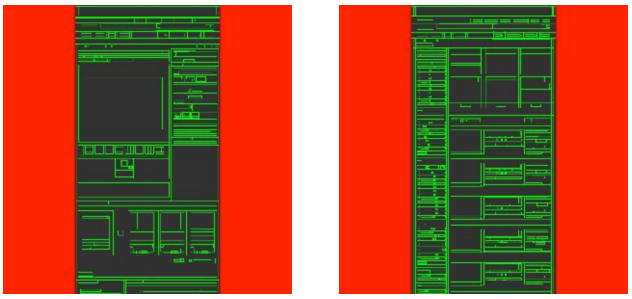
2.1 Page Type Classification

One approach of classifying page types is to render a web page as done by a browser to create an image of a page as seen by users. The goal here is to get an accurate representation of how users see web pages. However, the drawback is that the full rendering of each web page is quite resource intensive. In addition, it is not clear if the model is attributing the classification to certain products or other visual elements, rather than learning more abstract concepts. Figures 1(a) and 1(b) shows typical SP and MP pages to provide concrete examples.



(a) Single-product (SP) page (b) Multi-product (MP) page

Figure 1: Screenshot images. One product is prominently shown in the center of the page in SP page while each product is featured in one row of the block and users can scroll down to see more of similar products in MP page.



(a) Single-product (SP) page (b) Multi-product (MP) page

Figure 2: Wireframe images. A large block can be seen in the image in SP page while multiple similar blocks can be observed in MP page.

An alternate approach described in this paper is to generate lightweight web page structure representation. The idea is to depict the visual structure of the e-commerce pages without full scale rendering. The goal of this approach is to achieve the performance of image-based classification using high quality screenshots (as in our Web Render Service model), using a less resource-intensive page rendering mechanism that can make training and serving these models more efficient.

We first use internal tools to identify the render box of all blocks, which use different methods to identify the positions of the blocks on the screen [8, 10]. Then we use the OpenCV library [2] to render wireframe images of the render boxes. This allows us to create a simple wireframe representation of a web page that is unaffected by embedded images or other contents, and can classify the page purely by the page layout alone. Figures 2(a) and 2(b) show typical wireframe image of the SP and MP pages presented in Figures 1(a) and 1(b) respectively.

For both the image (screenshot) and wireframe representations, we used the Inception V2 model [19] trained on the JFT-300M dataset, which has more than 375M noisy labels (more than 19000 classes) for 300M images [7, 11, 18]. The last fully connected layer

weights of this model are randomly initialized and is fine-tuned using RMSProp optimizer, with learning rate of 5e-6, momentum (set to 0.9) and a random batch of size 48. The finetuning is stopped after 200,000 steps.

2.2 Main Product Phrase Classification

Once we know that the page has one main product (i.e., is a SP page), the next task is to identify what the product is. The purpose of the next model we will discuss is to classify phrases extracted from a web page as describing the product of the page or not. On a page that sells 'Nike Huarache Shoes', good product phrases could be for example 'Nike Huarache', while non-product phrases could include 'free delivery', 'heel support', 'Nike shoes review', etc. Since product/service names are usually noun phrases, only noun phrases extracted from the landing page are considered here. In this model, we use the candidate text (main product phrase), context sentence (a sentence containing the candidate text) and other metadata / features about the page to predict if the given phrase is a main product on the page. We also use the page title, topic of the page, url and salient terms found on the page as features. All text inputs are individually embedded using pre-trained word2vec embeddings (e.g., [14]). After that we used a CNN of width 2 to 5 each of which has 64 filters and 1-max pooling is performed over each feature map. On top of that we added a hidden layer of size 128 and before the last fully connected layer we fused all individual hidden layers into one layer by concatenating them.

2.3 Label/Dataset Generation

In this study, we make use of schema.org annotations [4] in both page type classification and main product phrase classification. According to the schema.org website, they provide "a collection of shared vocabularies webmasters can use to mark up their pages in ways that can be understood by the major search engines". Even though not all web pages have schema.org annotations, currently more than 1 million domain use Product annotations [3].

First, as we want to differentiate SP pages and MP pages, the most natural choice of the label is to count the number of products featured on the page and assign SP label if there is only one product and assign MP label if there are more than one, among all product pages. One way to achieve this is to use schema.org annotations. We collected some sample of search landing pages that contain https://schema.org/Product annotations — web pages that contain only 1 Product annotation are marked as a SP pages while the ones that contain more than 2 Product annotations are marked as MP pages. We discarded pages with exactly 2 Product annotations, as we have noticed from the eyeballing the data that there are a lot of mislabeled cases with 2 annotations. We also performed a human evaluation to estimate the quality of this labeling approach. We annotated 1k examples of each bucket. The results are summarized in Table 1 and they confirmed our concerns regarding pages with 2 Product annotations.

To avoid domain over-fitting, we limit to 5 pages per domain in our training data as well as we split our train/dev/test(80/10/10) sets by domain. In this way we were able to collect approximately 1.7 million labeled examples, 78% of which is SP and the remaining 22% is MP.

N	SP/MP	Label
1	93.0% / 4.3%	SP
2	45.3% / 51.4%	–
3+	31.4% / 61.0%	MP

Table 1: N is number of annotations and SP/MP shows human evaluated fraction of SP pages vs MP pages in each group. For example, among sample of web pages that has exactly 1 Product annotations, 93% of them are labeled as SP by human.

For the main product phrase classification, we use the phrase from the name property of product annotation as positive data. For the negative data, we use any other noun phrase found on the same web page. However, treating only the exact literal string match as the only truth is restrictive — it suffers lack of sufficient positive data as well as leads to many false negative results as well. In this study, we apply some rules to relax this. Let us suppose that the annotated phrase is "iconic silver charm **bracelet**" where the root of the phrase, bracelet, is shown in bold. The parse tree of this phrase looks like this, "[iconic] [[silver] charm] bracelet" showing "iconic" modifies "bracelet" as well as "silver" modifying "charm" and "charm" modifying "bracelet". Given this, any noun phrase sharing the same root as the annotated phrase, such as "PANDORA **bracelet**" or "iconic **bracelet**", would be considered positive. In cases where noun phrases contain an overlapping noun(s) but not sharing the same root, e.g., "silver charm" where "charm" is the root of the phrase, we discard these phrases as neither positive or negative examples. Finally, a noun phrase not sharing any noun or root token would be considered a negative example. This is demonstrated in Table 2. The dependency tree of each phrase is obtained using [6].

Phrase	Label
iconic silver charm **bracelet**	P
iconic **bracelet**	P'
silver charm **bracelet**	P'
PANDORA **bracelet**	P'
silver **charm**	I
iconic **store**	N

Table 2: Label of "P" is positive from the annotation (main product phrase), "P'" indicates induced synthetic positive (treated equally as true positive data) and "N" is negative. "I" is indefinite and discarded from training or evaluation. Root of each phrase is indicated by bold.

While generating training data we kept max 50 phrases per class/domain. We collected 85.7 million English phrases, 11.2% of which are product phrases and the rest 88.8% are non-product phrases.

3 RESULTS & DISCUSSIONS

First we evaluate our page type classification models using a sample of search landing pages in the held-out set.

Table 3 shows the precision, recall, F1 of the classifiers using threshold of 0.5 for the MP page class. Despite its simplicity, the wireframe image model presents comparable performance to the fully rendered image recognition model. The wireframe image model is multiple orders of magnitude faster than the full screenshot image model due to its simpler rendering, as seen in Table 3. We have observed memory consumption at the inference time is also significantly smaller in the wireframe model, offering superior characteristics for deploying in a production environment.

Approach	Precision	Recall	F1	Avg. Latency (ms)
Full Image	**0.89**	**0.88**	**0.89**	7,950
Wireframe Image	0.84	0.81	0.82	**230**[1]

Table 3: MP page classification using two different image recognition models. Wireframe approach, despite its simplicity, shows comparable metrics to full image rendering results at about 35 times faster processing time. Threshold of 0.5 is used.

Additionally we measured the quality of wireframe image classifier on 2k human labeled dataset (Table 4). We can see a significant drop in MP class precision, which can be explained by relatively low quality of our synthetically generated MP labels (as shown in Table 1) .However for our study we care more about the precision of SP classifier, which is relatively high (0.91).

Class	Precision	Recall	F1
Multi-product	0.69	0.83	0.75
Single-product	0.91	0.81	0.86

Table 4: Quality of classifier on 2k human labeled dataset. Threshold of 0.5 is used.

Next, we investigate how this page type classifier performs in each vertical. Table 5 shows the result of the classifier sliced by page vertical for the top 5 verticals that represent about 65% of the dataset. The vertical taxonomy and page type classification is taken from [1]. The model consistently perform well across different verticals. In particular the model has high precision (89% or higher) in identifying SP pages in most verticals. In fact, we see the average precision in bottom 5 categories is no different than the one in the bottom 5 categories (0.90), and out of 28 top level verticals we looked at, it shows 0.90 or higher precision on 22 verticals. This is useful for our current study as we are interested in identifying pages with a single product focus and be able to identify that product. Note that this evaluation is performed only on the pages that have the schema.org annotations.

Now armed with a precise selection of SP pages, we apply the main product phrase classification model to the web pages identified as SP pages. Table 6 shows the classification accuracy of the main product phrase classifier on a held out set.

We studied effects of different features used in the model by removing one or more features at the inference time. Table 7 shows

[1]This excludes pre-computation time of simplified layout rendering, since it can be easily cached in the production environment.

Vertical	P (MP)	R (MP)	P(SP)	R(SP)
/Shopping	0.87	0.85	0.92	0.93
/Home & Garden	0.83	0.78	0.90	0.93
/Autos & Vehicles	0.88	0.84	0.89	0.91
/Business & Industrial	0.79	0.73	0.89	0.91
/Computers & Electronics	0.81	0.75	0.89	0.92

Table 5: Slicing the web page type classification results by vertical. P indicates precision and R indicates recall, measure on respective class of MP or SP. Only the top 5 verticals from Google Cloud Content Categories [1] are shown, which represents approximately 65% of all eval data. All the results are from wireframe model.

Approach	Precision	Recall	F1
Product Phrase	.96	.94	.95
Non-Product Phrase	.94	.96	.95

Table 6: Main product phrase classifier results. Threshold of 0.5 is used.

this results. First, not surprisingly, it is clear that the phrase itself alone is not sufficient to differentiate the main product or not. There are some noun phrases that are clearly not that of a product. However there are many product phrases that are not the main product on the page, such as related items, which phrase itself is not sufficient to differentiate.

Ablated Feature	Loss in Precision
All but Phrase	-.23
Context	-.22
All but Phrase & Context	-.01
Page title	.00
Url	.00

Table 7: Feature ablation study at the inference time. Most of the predictive power comes from the phrase and its context sentence, and in fact only with these 2 features, it reaches to within 1 % of the best results.

Second, it is surprising that from other features only context sentence mattered for the classification. We took another look at our data and noticed that in a lot of cases context sentence for the product phrases is the product phrase itself. This is mostly due to the way we generated training data by using schema.org annotation: usually the annotated product phrase is a highlighted standalone sentence on the web page. This is not the effect we were looking for, since we also wanted to get the product phrases that located within sentences.

To overcome this issue we retrained our model without context sentence feature (Table 8). The quality of the new model is comparable to the previous one, we only lost 1 percent point in F1 product phrase score.

After that we did the ablation study again (Table 9). This time the most predictive features were the phrase itself as well as the

Approach	Precision	Recall	F1
Product Phrase	.95	.93	.94
Non-Product Phrase	.94	.96	.95

Table 8: Main product phrase classifier results without context sentence. Threshold of 0.5 is used.

title of the page. This result is aligned with our expectations, since the page title usually contains some highlights of the page, and for product pages the main highlight is the product itself.

Ablated Feature	Loss in Precision
All but Phrase	-.19
All but Phrase & Title	-.02

Table 9: Feature ablation study at the inference time of the model without context sentence. Most of the predictive power comes from the phrase and page title

4 CONCLUSION

In this paper, we presented two different image-based page structure detection models and a main product phrase classification model. The page type classifier model can be used to help us identify if the page focuses on a single product or multiple products. This allows us to extract a relevant product phrase from web pages where there is a single product focus. The main product phrase classifier can help us identify what that focused product is, among all the products mentioned on the page.

We would next like to combine the wireframe image recognition model with textual signals to understand the content better in the context of the presentation structure. In the future, we plan to combine this approach with information retrieval to detect and extract the most relevant information from a web page, with the help of web page structure.

ACKNOWLEDGMENTS

Authors would like to thank Ramakrishnan Srikant for thorough review of the manuscript.

REFERENCES

[1] 2018. Content Categories | Cloud Natural Language API | Google Cloud. https://cloud.google.com/natural-language/docs/categories
[2] 2019. OpenCV Library. https://opencv.org/
[3] 2019. "Product - schema.org". "http://schema.org/Product"
[4] 2019. schema.org. https://schema.org/
[5] 2019. U.S. Census Bureau News - QUARTERLY RETAIL E-COMMERCE SALES 3rd QUARTER 2018. https://www.census.gov/retail/mrts/www/data/pdf/ec_current.pdf
[6] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally Normalized Transition-Based Neural Networks. In *Association for Computational Linguistics*. https://arxiv.org/abs/1603.06042
[7] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint* (2017), 1610–02357.
[8] Michael Xu Erik Hendriks and Kazushi Nagayama. 2014. Understanding web pages better. https://webmasters.googleblog.com/2014/05/understanding-web-pages-better.html
[9] Xing Fang and Justin Zhan. 2015. Sentiment analysis using product review data. *Journal of Big Data* 2, 1 (16 Jun 2015), 5. https://doi.org/10.1186/s40537-015-0015-2
[10] Google. 2018. Understand rendering on Google Search. https://developers.google.com/search/docs/guides/rendering
[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
[12] Zornitsa Kozareva, Qi Li, Ke Zhai, and Weiwei Guo. 2016. Recognizing Salient Entities in Shopping Queries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL*.
[13] Prashant Mathur, Nicola Ueffing, and Gregor Leusch. 2018. Multi-lingual neural title generation for e-Commerce browse pages. (2018).
[14] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. http://arxiv.org/abs/1301.3781
[15] Christopher Mitcheltree, Veronica Wharton, and Avneesh Saluja. 2018. Using Aspect Extraction Approaches to Generate Review Summaries and User Profiles. (2018).
[16] Reid Pryzant, Sugato Basu, and Kazoo Sone. 2018. Interpretable Neural Architectures for Attributing an Ad's Performance to its Writing Style. In *The 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 125–135.
[17] Kashif Shah, Selcuk Kopru, and Jean David Ruvini. 2018. Neural Network based Extreme Classification and Similarity Models for Product Matching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. Association for Computational Linguistics. http://aclweb.org/anthology/N18-3002
[18] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *CoRR* abs/1707.02968 (2017). arXiv:1707.02968 http://arxiv.org/abs/1707.02968
[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*. http://arxiv.org/abs/1512.00567
[20] Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. OpenTag: Open Attribute Value Extraction from Product Profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '18)*.