Product Titles-to-Attributes As a Text-to-Text Task

Gilad Fuchs
eBay Research, Israel
gfuchs@ebay.com

Yoni Acriche
Bravado, USA
yoni@bravado.co

Abstract

Online marketplaces use attribute-value pairs, such as brand, size, size type, color, etc. to help define important and relevant facts about a listing. These help buyers to curate their search results using attribute filtering and overall create a richer experience. Although their critical importance for listings' discoverability, getting sellers to input tens of different attribute-value pairs per listing is costly and often results in missing information. This can later translate to the unnecessary removal of relevant listings from the search results when buyers are filtering by attribute values. In this paper we demonstrate using a Text-to-Text hierarchical multi-label ranking model framework to predict the most relevant attributes per listing, along with their expected values, using historic user behavioral data. This solution helps sellers by allowing them to focus on verifying information on attributes that are likely to be used by buyers, and thus, increase the expected recall for their listings. Specifically for eBay's case we show that using this model can improve the relevancy of the attribute extraction process by 33.2% compared to the current highly-optimized production system. Apart from the empirical contribution, the highly generalized nature of the framework presented in this paper makes it relevant for many high-volume search-driven websites.

1 Introduction

Many online marketplaces have new-listing forms that include both structured and unstructured input types to help sellers describe their listing[1]. While the unstructured part often includes free-text input boxes for title and description, a pictures upload option, etc., the structured part can include the selection of the listing category from a predefined list, or selecting specific attribute-value pairs (e.g. {"Brand":"Apple", "Color":"Black"}). Of the two, structured input often enables marketplaces a more streamline use of the data, since it requires less preprocessing and allows for more direct usage (via search results filters, etc.). On the flip-side, entering such data is more labor intensive for the sellers, and therefore, more expensive to get. This can also be intricate work for sellers since in most cases there are tens of different possible attribute names for every listing, with some attributes having more than one possible value.

To reduce the seller-inflicted cost of entering listing attribute values we set two solution guidelines: (a) sellers should focus on the top attributes that are expected to impact their listing discoverability. This aims to reduce the number of attributes for which seller attention is required and only focus on those which are likely to be used in the buyer-journey of their target audience. And (b), in an effort to further reduce friction, the marketplace should pre-populate a suggested value for each of these top attributes.

To identify the top attributes in a scalable manner we leveraged the rich historical data of buyer behavior on the eBay website. Like many other search-driven websites, eBay allows buyers to curate search results by applying filters on top of the initial results from the free-text-based query. Logging the filtering selections of buyers, alongside with their post-search actions, allows for an opportunity to learn what are the key attributes that buyers value when searching for the right result. For example, a common buyer behavior is to type a general description in the search box, like "handbag", and then to filter the results using more granular attributes, like "Material", etc. (Figure 1). Following this filtering step, the buyer might click on, and potentially purchase, a specific listing that was a part of the filtered results set. Mapping this buyer journey, from search to filtering and listing-click, allows to learn which attributes are most important for the discovery of every listing.

[1] or service; for simplicity we'll continue with the listing notation.

Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5), pages 91 - 98

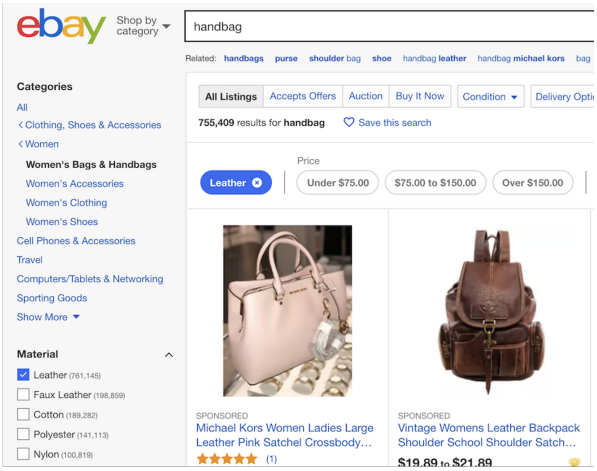

Figure 1: An example of a typical buyer search session. A buyer is searching for "handbag" in the search box (top) and further filters the results by selecting the attribute value "Leather" under "Material" (left).

From a modeling standpoint, to accommodate both of the solution guidelines above, the output set of the model should include the importance ranking of the top attributes and their expected value. As to the model input, in order for the solution to generalize across different downstream tasks, we need to pick a minimal viable data point that all listings have, but yet, that is highly informative. In our case that would be the listing title. Model design can be examined using different lenses; A supervised model based on the historical mapping between listing titles and multiple attribute-value pairs can be modeled as a multi-label text classification (MLTC) task. However, since there is a hierarchical relationship between the attributes and values (since each attribute has a finite list of possible values), the task can also be viewed as a hierarchical multi-label text classification (HMLTC) task. Last, since we care about the importance ranking of the attribute-value pairs, this can also be viewed as a ranking task. Recent Text-to-Text-driven approaches have shown to be highly valuable for various Natural Language Processing (NLP) tasks including MLTC and HMLTC (Nam et al., 2017; Yang et al., 2018; Chang et al., 2018; Li et al., 2018; Lin et al., 2018; Raffel et al., 2019). Inspired by these approaches, we demonstrate using a Text-to-Text framework in a HMLTC ranking task and compare it to other classification models.

Specifically in our case, the use of a Text-to-Text model approach is useful since it allows to produce multiple ranked hierarchical predictions, while separating between the probability score for the attributes and values. This introduces further flexibility to the solution (beyond the scope of the above guidelines) by allowing to report high impacting attributes even if we are uncertain about their expected attribute values. Furthermore, in comparison to approaches such as Named Entity Recognition (NER), a Text-to-Text model does not require the reported top attribute values to exist in the input title. This is useful since sellers are not always mentioning the most valuable attribute values in the listing title. Last, from an empirical standpoint, the Text-to-Text models we trained almost always outperformed models from other approaches (see section 4.2).

To conclude, in this work we suggest a scalable and automatic method for using listing titles to identify the most valuable set of attribute-value pairs by learning from the buyers' filtering behavior. In the next section we describe related work in the field of attribute-value extraction and hierarchical classification tasks. In the following section we describe our data collection methodology and the training procedures used for the four models that we trained. This is followed by a quantitative comparison of the results of the models, and a qualitative evaluation of the results of our best performing one. We conclude by discussing the tradeoffs of our current approach, and describe our plans for future work.

2 Related Work

Various methods are used to automatically extract attribute-value pairs from product-related text. This ranges from manual rules and regular expressions (Petrovski et al., 2014) to more advanced modern learning algorithms (Ghani et al., 2006; Kannan et al., 2011; de Bakker et al., 2013; Melli, 2014; Joshi et al., 2015; Ristoski and Mika, 2016; More, 2016; Petrovski and Bizer, 2017; Majumder et al., 2018; Charron et al., 2016). In contrast to our work, these methods are focusing on extracting the most complete set of attribute-value pairs, or limited to only attribute values which appear explicitly in the product-related text. Apart from (Charron et al., 2016), non of these works have leveraged data from historical user interaction with the attribute-value pairs.

Hierarchical classification has been of wide interest both in computer vision applications and text related tasks. Early work has been focusing on flattening the labels (Cai and Hofmann, 2004; Hayete and Bienkowska, 2005) or on training multiple local

classifiers, where the number of classifiers is dependent on the depth of the label hierarchy (Koller and Sahami, 1997; Sun and Lim, 2001; Cesa-Bianchi et al., 2006). More recent studies aimed to train a single neural network which can learn the label hierarchy complexity (Johnson and Zhang, 2015; Peng et al., 2018; Mao et al., 2019), while others combined both a single global network and multiple local classifiers (Wehrmann et al., 2018). Most recently, several works demonstrated that sequence-to-sequence (Seq2Seq) networks are a promising representation for hierarchical text classification tasks (Nam et al., 2017; Lin et al., 2018). However, less focus was given to using Seq2Seq for the ranking of multiple hierarchical label data structures, which are commonly being used, especially in online marketplaces.

3 Methodology

3.1 Datasets

Our training dataset includes information from two major eBay verticals - "Electronics" and "Fashion", where search-filtering activity is most frequent. The data includes roughly 10M and 3M random entities from Fashion and Electronics (respectively), all from the eBay US website. Each training entity includes a listing title and one matching attribute-value pair which was previously used in a single search filtering session to discover that listing. Since the distribution of attribute-value pairs has a long-tail, we reduced the complexity of the task by truncating the data to include only the top 800 most frequent combinations. Doing so, we kept 90% of all of the filtering activity done by buyers (which is considered sufficient coverage for our use case). We used 5% of the data for validation and and model selection, and an additional 5% for test. For non-hierarchical classification experiments we have concatenated attribute-value pairs to a single token (e.g. {"Color":"Black"} was transformed to "Color:Black"). For Seq2Seq hierarchical classification, we kept the pairs as two separated tokens (e.g. "Color Black"). Separating the tokens allows the Seq2Seq model to natively perform hierarchical classification, as the Seq2Seq decoder's predictions are dependent on the previous predicted tokens (e.g. in case the attribute prediction token is "Color" the next token prediction is likely to be a color name, such as "Black"). All tokens in multi-token attribute names or values were concatenated with an underscore as a delimiter. As duplications in the training set represent a frequent, and therefore more important, listing discovery pattern, the data was not de-duplicated in any way. For example, the title "Color Clash 100% Genuine Leather Snake Ladies Handbag Tote Shoulder Bag" might appear 20 times in the training data, out of which 12 times it will be coupled with the attribute-value pair {"Material":"Leather"}, 6 times with {"Style":"Tote"} and only 2 times with {"Size":"Large"}. The listing titles dataset was pre-processed by transforming the tokens to lowercase and removing known stopwords and non-alphanumeric characters.

3.2 Model Training

For the Text-to-Text approach we trained a Convolution Neural Network (CNN) Seq2Seq model (Gehring et al., 2017) via the Fairseq framework (Ott et al., 2019). For this we used a CNN architecture, following (Gehring et al., 2017), which consists an embedding layer, positional embedding layer, an encoder with 4 convolutional layers, a decoder with 3 convolutional layers and a kernel width of 3. The output of the each encoder convolutional layer is transformed by a non-linear gated linear units (GLU) (Dauphin et al., 2016) with the residual connections linking between the GLU blocks and the convolutional blocks. Each decoder GLU output undergoes a dot-product based attention with the last encoder GLU block output (see also (Gehring et al., 2017) for more details). Training was done with learning rate of 0.25, gradient clipping (clip-norm) of 0.1, dropout of 0.2, maximum number of tokens in a batch (max-tokens) of 4000 and max number of epochs of 15, with a Nesterov Accelerated Gradient (NAG) optimizer (NESTEROV, 1983) on a single GPU. Prior to training, pre-processing was done with "fairseq-preprocess" to build a vocabulary and binarize the data. For predictions, beam search size was set to 5. We trained two versions of the Seq2Seq models - one with attribute-value labels flattened to a single token (Seq2Seq-single), and the other where we kept their hierarchical structure (Seq2Seq-hierarchical), as described in section 3.1 above. Both versions were trained with the same hyper-parameters.

We tested our Text-to-Text modeling approach for attributes prediction against BERT and ULMFiT models, which have both been shown to be highly beneficial for multiple text classification tasks (Howard and Ruder, 2018; Devlin et al.,

2018). Apart from their past success, we also selected BERT and ULMFiT because they allowed us to test two different types of pre-training and fine-tuning approaches, as described below. For the multi-classification BERT model (Devlin et al., 2018), we used the FastBert library[2] which is based on HuggingFace (Wolf et al., 2019). The model that we fine-tuned was bert-base-uncased which includes 110 million parameters, 12 encoder layers consisting of 12 attention heads per layer and 768 hidden units. Fine-tuning was done for a maximum of 3 epochs with a batch size of 16, learning rate of 5e-5, a maximum sequence length of 128, a LAMB optimizer (You et al., 2019; Lan et al., 2019) and using 4 GPUs.

Next, for the multi-classification ULMFiT (Howard and Ruder, 2018) we used eBay's title corpus to fine-tune an English language model (LM) with an AWD-LSTM architecture (Merity et al., 2017a), which is an LSTM model with tuned dropout hyper-parameters that consists of an embedding size of 400, 3 layers and 1150 hidden activations per layer which were pre-trained on the Wikitext-103 dataset (Merity et al., 2017b) and downloaded from fast.ai[3]. The LM fine-tuning was done using the same data that is described in section 3.1, with a batch size of 64, a dropout set to 0.5, for 2 epochs using one cycle policy (Smith and Topin, 2019) and with a maximum learning rate of 1e-2 and 1e-3 for each on a single GPU. Next, a classifier model was trained while using the fine-tuned LM as an encoder, with a batch size of 64, for 3 epochs on 4 GPUs, using one cycle policy, with a discriminative layer training and gradual unfreezing (Howard and Ruder, 2018). During the first epoch only the last layer was fine-tuned, with a maximum learning rate of 1e-2. For the second epoch we fine-tuned the last two layer groups, with a maximum learning rate ranging between 2.5e-3 and 5e-3, and for the last epoch we fine-tuned all of the layers with a maximum learning rate ranging between 2e-5 and 2e-3. The labels for both BERT and ULMFiT were represented as a single token (see section 3.1 above). We also trained a multi-classification model for both, instead of a multi-label one, since we saw that the latter performed significantly worse.

All models were trained on data from eBay's Electronics and Fashion verticals as described at Section 3.1.

[2]https://github.com/kaushaltrivedi/fast-bert

[3]https://docs.fast.ai/index.html

4 Results

4.1 Evaluation Metrics

As commonly used in similar ranking tasks, we computed Precision at k (Prec@k) and normalized Discounted Cumulative Gain at k (nDCG@k or N@k) for model evaluation. Prec@k is defined as follows:

$$Prec@k = \frac{1}{k}\sum_{l=1}^{k} y_{\text{rank(l)}}$$

Where rank(l) is the index of the l-th highest predicted label and $y \in \{0,1\}^L$ is the true binary vector. nDCG@k is defined as follows:

$$DCG@k = \sum_{i=1}^{k} \frac{rel_{\text{i}}}{log(i+1)}$$

$$iDCG@k = \sum_{i=1}^{|REL_{\text{k}}|} \frac{rel_{\text{i}}}{log(i+1)}$$

$$nDCG@k = \frac{DCG@k}{iDCG@k}$$

Where rel_i is the relevance of the result at position i and REL_k represents the list of relevant documents (ordered by their relevance) in the corpus up to position k. The relevance score of each attribute-value pair per listing title is defined as the number of times it was used by buyers to filter the results, prior of clicking that specific listing.

4.2 Quantitative Evaluation

To compare the performance of the different models we computed the ranking accuracy of each using historic attribute-value pairs that were used by buyers to filter their results, prior of clicking a specific listing. As seen in Table 1, the Seq2Seq-hierarchical model outperformed the other models in most of the test criteria. Interestingly, both of the Seq2Seq models (single and hierarchical) outperformed BERT and ULMFiT in almost all of the metrics, which demonstrates the advantage of using a Text-to-Text frameworks in both hierarchical and non-hierarchical learning tasks.

In theory, the results from Table 1 could be purely due to better attribute value prediction by the Seq2Seq-hierarchical model, and not necessarily because of better attribute ranking. Therefore, to further examine the robustness of these results, we

Table 1: Model performance measured by Precision@k (P@k) and nDCG@k (N@k) comparison of the four models - ULMFiT (ULM), BERT, Seq2Seq-single (S2S) and Seq2Seq-hierarchical (S2S-hier) for the Electronics (Elec) and Fashion (Fash) verticals. Best results are marked in bold.

Data	Metric	BERT	ULM	S2S	S2S -hier
Elec	P@1	54	59.4	61.6	**62.7**
	P@3	33.3	37.2	39.4	**40.1**
	P@5	24.4	26.9	28.7	**29.2**
	N@1	50.6	56.2	58.1	**59.5**
	N@3	56.7	63.4	67.2	**68.3**
	N@5	60.5	66.7	71.1	**72.1**
Fash	P@1	61	62.8	62.8	**63.1**
	P@3	33.8	35.4	36.2	**36.3**
	P@5	23	24.1	**25.2**	**25.2**
	N@1	59.4	61.2	61.2	**61.5**
	N@3	64.7	67.7	68.8	**69**
	N@5	67.7	70.9	**72.6**	**72.6**

disconnected the ranking evaluation from the value prediction one, and tested the above models just on attribute ranking. To conduct this comparison we split the models' concatenated attribute-value predictions to attribute and attribute value predictions (i.e. {"Color:Black"} was split to "color" and "black") and re-computed the evaluation metrics only on the former. As seen in Table 2, the models' performance-ranking is overall consistent with previous experiments, with the Seq2Seq-hierarchical model also outperforming for the attribute ranking task.

In addition, from a pure technical perspective, Seq2Seq was the fastest model to train (x15 faster than BERT and x5 faster than ULMFiT), did not require any pre-trained models, and consisted of only a single training step (unlike ULMFiT, which also required an LM fine-tune step).

Table 2: Model performance comparison solely for the attributes ranking task. Best results are marked in bold.

Dataset	Metric	BERT Attr	ULM Attr	S2S Attr	S2S -hier Attr
Elec	P@1	92.4	94	93	**94.6**
	P@3	74	76	76.2	**78**
	P@5	51.8	53.8	56	**57.8**
	N@1	78.9	81.9	79.4	**82.1**
	N@3	82.8	85.2	84.3	**86.6**
	N@5	83.1	85.6	85.7	**87.8**
Fash	P@1	95.7	95.5	95.5	**96**
	P@3	61.9	61.2	63.2	**63.6**
	P@5	40.1	40.2	43.2	**43.5**
	N@1	86.2	85.9	87.2	**88**
	N@3	88.3	88.5	89.5	**90.3**
	N@5	87.7	88.4	90.2	**90.7**

To get a sense of the magnitude of impact that the Seq2Seq-hierarchical model could have on eBay's on-site experience, we compared our results to those from eBay's Attribute Extraction Service (AES). AES is a production system that has been highly optimized over the years, and is in charge of automatically extracting attribute-value pairs from titles that sellers provide. Currently it is mostly reliant on extensively curated rules that got added and optimized over the years. To compare the performance of the two methods we used around 15K attribute-value pairs that were used by buyers to filter search results and to discover a specific listing from the Electronics and Fashion verticals. For each we computed whether the attribute extraction method could automatically provide the relevant attribute-value given only the listing's title. This count was later divided by the number of attribute-value pairs to compute a percentage. As seen in Table 3, Seq2Seq-hierarchical led to an overall 33.2% improvement in relevant attribute-value extraction compared to AES.

Table 3: A comparison between eBay's current production system (AES) and the Seq2Seq-hierarchical (S2S-hier) model for the task of relevant attribute-value extraction. The number of attribute-value pairs which were used for the evaluation is denoted as N. For each method we show the percentage of cases that the relevant attribute-value pairs were extracted correctly (as defined by buyer behaviour).

Dataset	N	AES	S2S-hier
Electronics	10,289	58.8%	**71.9%**
Fashion	4,752	40.2%	**67.4%**
Total	15,041	52.9%	**70.5%**

4.3 Qualitative Evaluation

Since Seq2Seq-hierarchical outperformed the other models (Table 1), we focused our qualitative evaluation only on its predictions. Table 4 shows examples of the top predictions of five different listings, ordered by the model likelihood score (descending order).

As seen in Table 4, {"Brand":"Ray-Ban"} was only the 3rd most important attribute-value pair picked by the model for the title "Ray-Ban G-15 Aviator Black Frame Black Classic 58mm". This can be counterintuitive from a domain expertise standpoint, since the latter is clearly a

Table 4: Example of Seq2Seq-hierarchical prediction, including values which are not explicitly mentioned in the title and multi-values attributes. Values are ordered by their importance rank.

Title	Predictions
Ray-Ban G-15 Aviator Black Frame Black Classic	{"Frame Color":"Black", "Lens Color": "Black", "Brand": "Ray-Ban"}
Asus Strix Gaming LGA1151 DDR4 Motherboard	{"Form Factor": "microATX", "Compatible CPU Brand": "Intel"}
DJI Phantom 4 Aerial UAV Drone Quadcopter	{"Camera": "Included", "Features": "4K HD Video Recording"}
Nike Air Max Shoes Men's Size 7-9	{"US Shoe Size (men's)": [8, 8.5, 9, 7.5, 7]}
Men's Slim Fit Coat Jean Denim Jacket Size S-XL	{"Size (men's)": ["M", "L", "XL", "S"]}

more differential attribute-value pair for the category of sunglasses than, for example, {"Frame Color":"Black"}, which was picked first. However, looking at a sample of the search queries that were prior to the filtering steps (not shown here), we see that 93% of them already contained some variation of the term "Ray-Ban" (e.g. "rayban sunglasses", "ray ban sunglasses aviator", "ray-ban aviator"). Therefore most of the search engine's out-of-the-box results already included "Ray-Ban" branded sunglasses, which mitigated the need to further filter by brand. In contrast, only 2% of the queries mentioned the color "black", which explains the frequent buyer behavior of further filtering the results by color after seeing the search results (which included sunglasses from various colors). Such ranking results are in-line with our solution guideline to identify the top attributes that are expected to be used in the listing's buyer-discovery-journey, and therefore, help maximize the listing's chances to be discovered.

In Table 4 we provide further prediction examples which show that our Text-to-Text model does not require the reported top attribute values to be included in the input title. In addition, we evaluated the model's predictions in cases where attributes can include multiple values, like with 'size', and show that the model successfully extracts all of the relevant values from the ranges that appear in the titles. Note that the different likelihood prediction for each size value can serve as proxy to its popularity among buyers.

5 Conclusion

In this paper we demonstrate using filtering behavior data to predict the most relevant listing attribute-value pairs, and the superiority of using a Text-to-Text approach for modeling a hierarchical multi-label text classification (HMLTC) task that combines ranking. We identify several key advantages of this solution framework: First, acquiring the training data we use is a scalable and inexpensive process which does not require manual labor. Therefore, the volume of data collected in high-volume websites is likely to be sufficient for training deep-learning-based models such as Seq2Seq. Second, unlike methods such as NER, using a Text-to-Text approach enables to identify attribute-value pairs that do not necessarily exist in the title, to extract multiple values per attribute (Table 4) and to separately analyze the importance of every possible attribute-value pair. Third, as to the choice of hierarchical modeling, this allows us to separately analyze the likelihood probabilities of the expected attributes and values, which further generalizes the model for additional downstream tasks.

As for classifiers performance, the Seq2Seq models provided better results for most metrics compared to BERT and ULMFiT. Unlike the latter two, the Seq2Seq models didn't use a Transfer Learning approach that leverages a pre-trained Language Models. We suspect that the relatively short length of listing titles (12 tokens on average), combined with the unique jargon in eBay's data, which is hard to fully capture in the fine-tune process, might have negatively impacted the performance of BERT and ULMFiT.

Regardless to the classifier of choice, we keep in mind that the model's attribute ranking is clearly affected by the set of filtering options that were presented to the buyers on the site, and thus, cannot find attribute pairs that have not been historically used for filtering. Therefore, to avoid a closed feedback loop scenario, we would avoid using the model's attribute ranking results as an input to decide these filtering options. Also, to further increase the quality of the attribute ranking we can use a training data that consists of a sample of buyers that were served with a random (or partly random) list of filtering options. Nonetheless, even without this sample, the model can still provide sellers with meaningful information about their potential buyers' current attribute priority ranking.

References

Lijuan Cai and Thomas Hofmann. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, page 78–87, New York, NY, USA. Association for Computing Machinery.

Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. 2006. Hierarchical classification: Combining bayes with svm. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 177–184, New York, NY, USA. Association for Computing Machinery.

Wei-Cheng Chang, Hsiang-Fu Yu, Inderjit S. Dhillon, and Yiming Yang. 2018. Secseq: Semantic coding for sequence-to-sequence based extreme multi-label classification.

Bruno Charron, Yu Hirate, David Purcell, and Martin Rezk. 2016. Extracting semantic information for e-commerce. pages 273–290.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks.

Marnix de Bakker, Flavius Frasincar, and Damir Vandic. 2013. A hybrid model words-driven approach for web product duplicate detection. In *CAiSE*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. of ICML*.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.*, 8(1):41–48.

Boris Hayete and Jadwiga Bienkowska. 2005. Gotrees: Predicting go associations from protein domain composition using decision trees. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 10:127–38.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado. Association for Computational Linguistics.

Mahesh Joshi, Ethan Hart, Mirko Vogel, and Jean-David Ruvini. 2015. Distributed word representations improve NER for e-commerce. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 160–167, Denver, Colorado. Association for Computational Linguistics.

Anitha Kannan, Inmar E. Givoni, Rakesh Agrawal, and Ariel Fuxman. 2011. Matching unstructured product offers to structured product specifications. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 404–412, New York, NY, USA. Association for Computing Machinery.

Daphne Koller and Mehran Sahami. 1997. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, page 170–178, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Wei Li, Xuancheng Ren, Damai Dai, Yunfang Wu, Houfeng Wang, and Xu Sun. 2018. Sememe prediction: Learning semantic knowledge from unstructured textual wiki descriptions.

Junyang Lin, Qi Su, Pengcheng Yang, Shuming Ma, and Xu Sun. 2018. Semantic-unit-based dilated convolution for multi-label text classification.

Bodhisattwa Prasad Majumder, Aditya Subramanian, Abhinandan Krishnan, Shreyansh Gandhi, and Ajinkya More. 2018. Deep recurrent neural networks for product attribute extraction in ecommerce.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical text classification with reinforced label assignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 445–455, Hong Kong, China. Association for Computational Linguistics.

Gabor Melli. 2014. Shallow semantic parsing of product offering titles (for better automatic hyperlink insertion). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1670–1678, New York, NY, USA. Association for Computing Machinery.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017a. Regularizing and optimizing lstm language models.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017b. Pointer sentinel mixture models.

Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. *CoRR*, abs/1608.04670.

Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. 2017. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5413–5423. Curran Associates, Inc.

Y. E. NESTEROV. 1983. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1063–1072, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Petar Petrovski and Christian Bizer. 2017. Extracting attribute-value pairs from product specifications on the web. In *Proceedings of the International Conference on Web Intelligence*, WI '17, page 558–565, New York, NY, USA. Association for Computing Machinery.

Petar Petrovski, Volha Bryl, and Christian Bizer. 2014. Learning regular expressions for the extraction of product attributes from e-commerce microdata. In *Proceedings of the Second International Conference on Linked Data for Information Extraction - Volume 1267*, LD4IE'14, page 45–54, Aachen, DEU. CEUR-WS.org.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Petar Ristoski and Peter Mika. 2016. Enriching product ads with metadata from html annotations. In *Proceedings of the 13th International Conference on The Semantic Web. Latest Advances and New Domains - Volume 9678*, page 151–167, Berlin, Heidelberg. Springer-Verlag.

Leslie N. Smith and Nicholay Topin. 2019. Super-convergence: very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pages 369 – 386. International Society for Optics and Photonics, SPIE.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, page 521–528, USA. IEEE Computer Society.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5075–5084, Stockholmsmässan, Stockholm Sweden. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: Sequence generation model for multi-label classification.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes.