OpenBrand: Open Brand Value Extraction from Product Descriptions

Kassem Sabeh
Free University of Bozen-Bolzano
ksabeh@unibz.it

Mouna Kacimi
Wonder Technology Srl
mouna@wonderflow.ai

Johann Gamper
Free University of Bozen-Bolzano
jgamper@unibz.it

Abstract

Extracting attribute-value information from unstructured product descriptions continue to be of a vital importance in e-commerce applications. One of the most important product attributes is the brand which highly influences customers' purchasing behaviour. Thus, it is crucial to accurately extract brand information dealing with the main challenge of discovering new brand names. Under the open world assumption, several approaches have adopted deep learning models to extract attribute-values using sequence tagging paradigm. However, they did not employ finer grained data representations such as character level embeddings which improve generalizability. In this paper, we introduce OpenBrand, a novel approach for discovering brand names. OpenBrand is a BiLSTM-CRF-Attention model with embeddings at different granularities. Such embeddings are learned using CNN and LSTM architectures to provide more accurate representations. We further propose a new dataset for brand value extraction, with a very challenging task on zero-shot extraction. We have tested our approach, through extensive experiments, and shown that it outperforms state-of-the-art models in brand name discovery.

1 Introduction

Brand name plays a very important role in influencing customers' behaviour (Chovanová et al., 2015; Shahzad et al., 2014). Typically, as customers are aware of the brand, they can deduce knowledge about other product attributes. Let us take the example of the toy shown in Figure 1. The brand of this product is "Gentle Monster". By knowing the brand, customers would have some kind of associations, like this toy would be of *"a soft and smooth wood"*, have *"bright colors"*, and contain *"small pieces which is suitable for older kids"*. So, when shopping for toys, they would pick a particular brand based on the attributes they find important. Such correlations between brands and

Wooden Stacking Board Games 54 Pieces for Kids Adult and Families, Gentle Monster Wooden Blocks Toys for Toddlers, Colored Building Blocks - 6 Colors 2 Dice

Figure 1: An example of a product description.

product attributes make it crucial for e-commerce applications to accurately extract brand names from product descriptions.

Retrieving brand names is addressed in the literature within the general problem of attribute-value extraction from product descriptions (Kovelamudi et al., 2011; Vandic et al., 2012; Ghani et al., 2006; Kozareva et al., 2016; Zheng et al., 2018; Xu et al., 2019). Early approaches rely on rule-based techniques which use domain-specific knowledge to identify attributes and values (Kovelamudi et al., 2011; Vandic et al., 2012; Ghani et al., 2006). Such approaches adopt a closed world assumption requiring the possible set of values to be known beforehand by mean of dictionaries or hand-crafted rules. Consequently, they are not suitable for discovering unseen values such as newly emerging brands. To tackle this problem, most recent approaches model the extraction task as sequence tagging (Kozareva et al., 2016; Zheng et al., 2018; Xu et al., 2019) and solve it using deep learning models such as BiLSTM enhanced by Conditional Random Field (CRF) and Attention layers. These new approaches achieve promising results, however, they limit the representation of their data to word embeddings which can capture context but penalizes generalizability to new brands.

In this paper, we propose to use character level embeddings in sequence tagging models for discovering brand names. In addition to word embeddings, character level embeddings were employed

Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5), pages 161 - 170

in Named Entity Recognition (NER) tasks (Lample et al., 2016) to handle out-of-vocabulary words. The problem of unseen words is particularly emphasized in brands because of sub-branding, brand fragmentation, or simply emerging businesses. Unseen brand names can be completely new, like in brand fragmentation where new brands share the same parent brand maintaining minimal links between the new and the existing identities. For example, *"Audi"* and *"Porsche"* do not have any similarity although they have the same parent brand *"Volkswagen"*. By contrast, sub-branding would maintain stronger links between existing brands and the new generated ones, which can be reflected by similarities in brand names. Examples include *"Uber"* and *"UberPool"*, *"McDonalds"* and *"McCafe"*, or *"Samsung"* and *"Samsung Evo"*. Thus, the use of character level embedding is crucial for capturing variations in brand names and the occurrence of unseen brands.

We summarize the main contributions of this work as follows:

1. We propose OpenBrand, a BiLSTM-CRF-Attention model that combines word embeddings with character level word embeddings. In contrast to previous approaches, we learn character level embeddings based on CNN and LSTM architectures to obtain specific representations of our data.

2. We provide a large real world dataset[1] focusing on brand names to have a thorough analysis of the impact of character level embeddings. We experimentally show that our dataset is challenging on brand name extraction, especially those zero-shot brand values.

3. We empirically demonstrate significant improvements in F1 score over several state-of-the-art baselines on brand name extraction. Additionally, we show that OpenBrand guarantees a better generalizability over new brands and deals more effectively with compound brand names.

2 Problem Statement

In this section, we formally define the problem of open brand value extraction. Given a product title, represented as an unstructured text data, and a target attribute (eg. brand), our goal is to extract the appropriate values for the corresponding attribute from the product title. In this context, we want to discover new values that have not been encountered before. We formalize the attribute-value extraction as per the following definition:

Definition *Given a product title X. The title X is represented as a sequence of tokens $X_t = \{x_1, ..., x_T\}$, where T is the sequence length. Consider a target attribute A. Attribute-value extraction automatically identifies a sub-sequence of tokens from X_t as applicable attribute-value pair: $A_v = \{x_i, x_{i+1}, ..., x_k\}$, for $1 \leq i \leq k \leq T$.*

For example, consider the title for the product given in the example of Figure 1:

X = "Wooden Stacking Board Games 54 Pieces for Kids Adult and Families, Gentle Monster Wooden Blocks Toys for Toddlers, Colored Building Blocks - 6 Colors 2 Dice."

The tokenization of X yields: $X_t = \{x_1, x_2, ..., x_{25}\}$ = {"Wooden", "Stacking", "Board", .., "Dice"}, where $T = 25$. For the target attribute: A = {"Brand"}. We want to extract: Brand = $\{x_{12}, x_{13}\}$ = {"Gentle", "Monster"}.

In order to identify these sub-sequences, the sequence of tokens X_t need to be tagged to capture sequential and positional information. For this purpose, we adopt the sequence tagging model and associate a tag from a given tag-set to the sequence of input tokens X_t. We experimented with different tagging strategies and, inline with previous work in the literature (Xu et al., 2019), we found that the {B,I,O} tagging scheme produced the best results, where "B", "I", and "O" represent the beginning, inside, and outside of an attribute, respectively. (A sequence of "O" tags corresponds to the absence of an attribute). Table 1 shows an input/output example of the {B,I,O} tagging strategy.

Input	Kids	Adult	Families	Gentle	Monster	Wooden	Blocks	Toys
Output	O	O	O	B-Brand	I-Brand	O	O	O

Table 1: Example of an input/output {B,I,O} tag sequence for the brand of a product description.

3 OpenBrand Model

To address the open brand value extraction problem, we propose a BiLSTM-CRF-Attention model with character level embeddings. Figure 2 shows

[1]Data is available at https://github.com/kassemsabeh/open-brand.

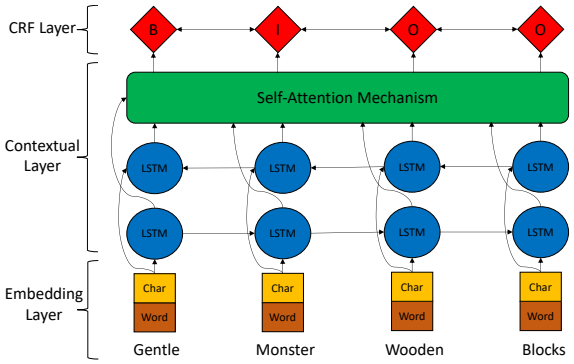


Figure 2: OpenBrand Architecture: BiLSTM-CRF-Attention with character level representations.

our OpenBrand model architecture, which is composed of three main layers: an embedding layer that encodes the input sequence, a contextual layer that captures complex relationships among the input sequence, and an output layer that produces the output labels.

3.1 Embedding Layer

In the embedding layer, we map every word in the product description into a d-dimensional embedding vector. The embeddings of the words are obtained by concatenating the word embeddings and character level embeddings. Word embeddings are obtained from the pre-trained GloVe (Pennington et al., 2014) word representations, which are trained over large unlabeled corpus. Pre-trained word embeddings, such as GloVe and Word2Vec (Mikolov et al., 2013), offer a single representation for each word, which is not useful in the case where words have different meanings depending on the context. To allow our model to learn different representations of embeddings depending on the context, we learn and generate different representations of tokens in the input sequence. For this reason, the weights of our embedding layer are considered to be learnable parameters and not fixed.

An important distinction of our approach, compared to previous work on attribute-value extraction, is that we learn character level features in our model. For character level embeddings, we use two different architectures: CNN-based and LSTM-based character level representations. Learning character level embeddings has the advantage of learning task-specific representations. Convolutional Neural Networks (CNN) are designed to discover position-invariant features and they are highly effective in extracting morphological information (ex. prefix or suffix of words) (Chiu and Nichols, 2016). On the other hand, LSTMs are capable of encoding long sequences, and are thus capable of extracting position dependent character features. These features are crucial to model the relationships between words and their characters. Given a token of our input sequence x_t, the embedding layer maps x_t in to the vector:

$$e_t = [w_t; c_t],$$

where w_t and c_t are the word and character level representations of x_t, respectively. The embedding representation of the whole input sequence X_t would be $\{e_1, e_2, ..., e_T\}$. Figure 3 illustrates the two architectures used to encode the character representations. These character representations are then concatenated with the word embeddings and fed as input to our contextual layer.

3.2 Contextual Layer

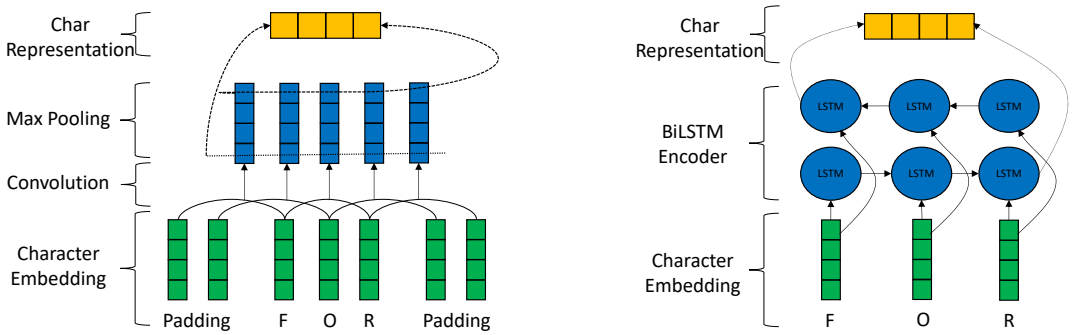
The contextual layer captures contextualized representations for every word in the input sequence. In our model, the input sequence to the contextual layer is the concatenation of the character level representations and word embeddings, both mapped by the underlying embedding layer. In this stage, we employ a BiLSTM contextual layer followed by a self-attention layer.

Long Short Term Memory Networks (Hochreiter and Schmidhuber, 1997) address the vanishing gradient problems of Recurrent Neural Networks and are thus capable of modeling long-term dependencies between tokens in a sequence. Bidirectional LSTM (BiLSTM) can capture both past and future time steps jointly by using two LSTM layers to produce both forward and backwards states, respectively. Given the input e_t (embedding of a token x_t), the hidden vector representations from the backward and forward LSTMs ($\overrightarrow{h_t}$ and $\overleftarrow{h_t}$) is:

$$h_t = \Delta([\overrightarrow{h_t}; \overleftarrow{h_t}])$$

where Δ denotes a non-linear transformation. The hidden representation of the whole input sequence X_t is $H_t = \{h_1, h_2, ..., h_T\}$.

In reality, not all hidden states generated by the BiLSTM layer are equally important for the labeling decisions. A mechanism that allows the output layer to be aware of the important features of the sequence can improve the prediction model. This is exactly what attention does. Attention mechanisms have achieved great success in Natural Language Processing (NLP) and were first introduced



(a) CNN-based character level representation.

(b) LSTM-based character level representation.

Figure 3: Architecture of character level encoders.

in the Neural Machine Translation task (Bahdanau et al., 2015). In the contextual layer, we use a self-attention mechanism to highlight important concepts in the sequence rather than focusing on everything. The model learns to *attend* to the important parts of the input states based on the output produced so far. We first compute the similarity between all hidden states representations to obtain an attention matrix $A \in \mathbb{R}^{T \times T}$ where

$$\alpha_{t,t'} = \sigma(w_\alpha g_{t,t'} + b_\alpha)$$

is the element of matrix A representing the mutual interaction between hidden states h_t and $h_{t'}$. σ is the element-wise sigmoid function, and

$$g_{t,t'} = tanh(W_1 h_t + W_2 h_{t'} + b_g)$$

where W_1, W_2, w_α are trainable attention matrices, and b_g, b_α are trainable biases. The contextualized hidden states can be computed as

$$\widetilde{h}_t = \sum_{t'=1}^{T} \alpha_{t,t'} \cdot h_{t'}$$

The contextualized hidden state of the whole input sequence X_t is $\widetilde{H}_t = \{\widetilde{h}_1, \widetilde{h}_2, ...\widetilde{h}_T\}$.

3.3 CRF Layer

In sequence labeling tasks, it is important to consider the dependencies between output tags in a neighborhood. Conditional Random Fields (CRF) allow us to capture the correlation between labels and model their sequence jointly. For example, if we already know the tag of a token is I, then this increases the probability of the next token to be I or O, rather than being B. We feed the contextualized hidden states $\widetilde{H}_t = \{\widetilde{h}_1, \widetilde{h}_2, ...\widetilde{h}_T\}$ to our output CRF layer to get the sequence of labels with highest probabilities. The joint probability distribution of a tag y given the hidden state $\widetilde{h}_t$ and previous tag y_{t-1} is given by

$$Pr(y|x;\psi) \propto \prod_{t=1}^{T} exp\left(\sum_{k=1}^{K} \psi_k f_k(y_{t-1}, y_t, \widetilde{h}_t)\right)$$

where ψ_k is the corresponding learnable weight, f_k is the feature function, and K is the number of features. The final output label is the label with the highest conditional probability, given as

$$y^* = argmax_y Pr(y_i|x_i;\psi)$$

where $y^* \in \{B, I, O\}$ is the output tag.

In Section 5.2, we will study in detail the effect of the attention and CRF layers on the discovery of brands in comparison with the embeddings layer.

4 Experimental Setup

This section presents the experimental settings of our empirical approach for comparing state-of-the-art models on the task of brand value extraction.

4.1 Dataset

To evaluate the effectiveness of OpenBrand, we have collected a dataset that contains information about products from Amazon. Our dataset is derived from a public product collection - the Amazon Review Dataset (Ni et al., 2019) [2]. The categories of the collected dataset contained a large amount of overlapping brands, which might bias the results of the experiments. Thus, we have selected a subset to have a diverse set of brands with minimal overlapping across categories. We also processed

[2] https://nijianmo.github.io/amazon/index.html

Category	Train	Val	Test
Grocery & Gourmet Food	15679	2239	4479
Toys & Games	44314	6330	12660
Sports & Outdoors	37951	5421	10842
Electronics	33512	4787	9574
Automotive	45132	6447	12894
Total	176588	25224	50449

Table 2: Statistics of AZ-base dataset with five categories.

Category	Samples
Clothing, Shoes & Jewelry	85068
Pet Supplies	10868
Cell Phones & Accessories	78564
Total	174500

Table 3: Number of samples in AZ-new-cat dataset.

the dataset to handle noise, and removed samples with empty values. This led to a dataset comprising over 250k product titles with more than 50k unique values, which we refer to as AZ-base dataset in our experiments. The AZ-base dataset contains information about products in five main categories: *Grocery & Gourmet Food, Toys & Games, Sports & Outdoors, Electronics* and *Automotive*. We randomly sample 70% of the data for training, 10% for validation, and 20% for testing. Table 2 shows the statistical details of the AZ-base dataset.

To further examine the generalization ability of our model, we divide the AZ-base dataset into another training and test split with no overlapping brand values. In other words, none of the values in the test set are encountered during training. We refer to this data split as AZ-zero-shot, as it is designed for evaluating zero-shot extraction. The test set of AZ-zero-shot contains more than 8k new and unique brand values.

In addition, we have also chosen another subset of products from our collected data with another set of categories. The purpose of this dataset is to test the models capabilities in detecting brand values across different category domains. The dataset contains information about products in three new categories as shown in Table 3. We refer to this dataset as AZ-new-cat, as it is designed to evaluate the model on a new set of product categories.

4.2 Models Under Comparison

We implemented and compared three state-of-the-art baseline models on attribute-value extraction.

BiLSTM (Hochreiter and Schmidhuber, 1997) which uses word embeddings from pretrained GloVe (Pennington et al., 2014) for word level representation, then applies BiLSTM to produce the contextual embeddings.

BiLSTM-CRF (Huang et al., 2015) which extends the BiLSTM model by adding a CRF layer on top to model the tagging decisions jointly. This model is considered state-of-the-art sequence tagging model for NER.

OpenTag (Zheng et al., 2018) which adds a self attention mechanism between the contextual BiLSTM layer and the CRF decoding layer. OpenTag is considered the pioneer sequence tagging model for attribute-value extraction.

We compare the above baseline models with the OpenBrand models we proposed in Section 3.

OpenBrand-LSTM In this approach, character level information is obtained by applying a BiLSTM encoder on the sequence of characters in each word. This character level information is used in combination with word-level embeddings as input to the BiLSTM-CRF-Attention model.

OpenBrand-CNN This approach is similar to the above model, but CNNs are used instead of LSTMs to encode character level information in the word sequences.

We use precision P, recall R and F_1 score as evaluation metrics based on the number of true positives (TP), false positives (FP), and false negatives (FN). We use *Exact Match* criteria (Rajpurkar et al., 2016), in our evaluation, with either full or no credit. The implementation details are provided in the Appendix.

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad F_1 = 2 \times \frac{P \times R}{P + R}$$

5 Results and Discussion

We conducted a series of experiments on AZ-base, AZ-zero-shot, and AZ-new-cat datasets under various settings to evaluate the performance of OpenBrand.

5.1 Baseline Performance Comparison

In the first experiment, we compare the performance of OpenBrand with the three state-of-the-art baselines mentioned in Section 4.2 for identifying brand values from product descriptions. Table

Category	Models	P	R	F1
Grocery & Gourmet Food	BiLSTM	70.4	65.9	68.1
	BiLSTM-CRF	74.9	66.0	70.2
	OpenTag	76.0	65.4	70.3
	OpenBrand-LSTM	75.9	**77.5**	71.8
	OpenBrand-CNN	**77.5**	75.4	**76.4**
Toys & Games	BiLSTM	73.7	69.1	71.3
	BiLSTM-CRF	78.9	70.5	74.5
	OpenTag	79.1	70.3	74.5
	OpenBrand-LSTM	80.2	**72.4**	76.1
	OpenBrand-CNN	**81.3**	72.0	**76.4**
Sports & Outdoors	BiLSTM	80.3	75.8	78.0
	BiLSTM-CRF	84.1	75.4	79.5
	OpenTag	84.9	75.0	79.6
	OpenBrand-LSTM	85.7	76.8	81.0
	OpenBrand-CNN	**86.1**	**77.3**	**81.5**
Electronics	BiLSTM	86.2	80.4	83.2
	BiLSTM-CRF	87.8	81.5	84.5
	OpenTag	89.2	79.6	84.2
	OpenBrand-LSTM	89.1	**80.8**	84.8
	OpenBrand-CNN	**89.7**	80.5	**84.9**
Automotive	BiLSTM	88.5	84.3	86.4
	BiLSTM-CRF	90.9	85.0	87.9
	OpenTag	91.6	84.6	87.9
	OpenBrand-LSTM	91.7	85.0	88.2
	OpenBrand-CNN	**91.8**	**85.4**	**88.5**

Table 4: Performance comparison between different models on AZ-base dataset.

4 reports the comparison results of our two models (OpenBrand-LSTM and OpenBrand-CNN) and three baselines across all categories in the AZ-base dataset. From these evaluation results, we can observe that our models substantially outperform the other compared models in all categories. OpenBrand with LSTM character level and CNN character level embeddings are consistently ranked the best over all competing baselines. The overall improvement in F_1 score is up to 6.1% as compared to OpenTag. The main reason for this result is that our model learns both character and word embeddings during training, thus allowing to learn more effective contextual embeddings that are more suitable for the task of extracting brand values.

5.2 Impact of Character level Representations

To understand the effect of character level representations on brand-value extraction, we extend all baseline models with character level embeddings and test them on the AZ-base dataset. Table 5 shows the average F_1 score of baseline models on the AZ-base dataset after adding character level representations. The results show that character level embeddings significantly improve the overall performances of all models. An interesting observation is that character level embeddings improve the model much more effectively than CRF or attention layers. For example, and as shown in the last two rows of Table 5, adding a CNN-representation to a BiLSTM-CRF model improves the model by 1.15%, while adding an attention layer only improves the model by 0.14%.

Model	Base	LSTM-char	CNN-char
BiLSTM	78.56	79.71	**79.73**
BiLSTM-CRF	80.37	81.11	**81.52**
OpenTag	80.51	81.62	**81.85**

Table 5: Effect of character embeddings on the performance of the models (F_1 score).

The experiments also show that using either CNN-char or LSTM-char both lead to an improvement with comparable overall F_1 score. However, CNNs have less training complexity as compared to LSTM models under similar experimental settings. In our experiments, the average training time of models with LSTM-char increased by 59% relative to the baseline BiLSTM-CRF-Att model, while it only increased by 22% with CNN-char, as detailed in Table 6. CNN-char also produces better performances than LSTM-char as shown in Table 5. We conclude that CNN character representations are preferable to LSTM based representations for brand-value extraction.

Model	Average Training Time per Epoch (seconds)	Difference ($\Delta\%$)
BiLSTM-CRF-Att	63	0
+LSTM-char	100	+59%
+CNN-char	77	**+22%**

Table 6: Average training time of our BiLSTM-CRF-Att models computed on a TPU.

5.3 Discovering New Brand Values

We conduct zero-shot extraction experiment to evaluate the generalization ability of our models on unseen brand values. Table 7 reports the zero-shot extraction results. It can be seen that our model achieves better performance than OpenTag on unseen data. This is because our model can leverage the sub-sequence level similarities in brand names between the train set and test set, through the character level embeddings. However, it is clear that the overall performance of all models is worse as compared to the results in Table 4, which is inline

Model	P	R	F1
OpenTag	53.80	33.82	41.53
OpenBrand-LSTM	**56.17**	35.14	43.23
OpenBrand-CNN	55.61	**35.46**	**43.44**

Table 7: Zero-shot extraction results on AZ-zero-shot dataset.

with our expectations as there are no training samples for the zero-shot brand values. This indicates that it is truly a difficult zero-shot extraction task.

To further examine the ability of OpenBrand in discovering brand values in new categories, we train the models on the AZ-base dataset, and test them on the AZ-new-cat dataset introduced in Section 4.1. Table 8 reports the results across three different categories in the AZ-new-cat dataset. It is clear that OpenBrand achieves much better performance with gains up to 2.7% in F_1 score as compared to OpenTag. This indicates that our model has good generalization and is able to transfer to other domains. Also, the results are much better than zero-shot extractions. This is because some data in the training set are semantically related to the brand values in AZ-new-cat and thus they provide hints that guide the extraction. For example, many of the brands in *Cell Phones & Accessories* category (eg. Samsung Galaxy) are sub-brands of products in *Electronics* category (eg. Samsung).

Category	Models	P	R	F1
Clothing, shoes, & Jewelry	BiLSTM	52.6	44.3	48.1
	BiLSTM-CRF	58.5	42.2	49.0
	OpenTag	60.3	43.5	50.5
	OpenBrand-LSTM	63.8	44.7	52.6
	OpenBrand-CNN	**64.5**	**45.2**	**53.2**
Pet Supplies	BiLSTM	49.1	39.4	43.7
	BiLSTM-CRF	55.0	37.3	44.5
	OpenTag	53.9	38.9	45.2
	OpenBrand-LSTM	57.3	**39.8**	**47.0**
	OpenBrand-CNN	**58.2**	38.5	46.3
Cell Phones & Accessories	BiLSTM	81.2	63.8	71.5
	BiLSTM-CRF	80.1	68.0	73.5
	OpenTag	78.3	67.4	72.4
	OpenBrand-LSTM	83.3	**70.7**	**76.5**
	OpenBrand-CNN	**85.2**	67.8	75.5

Table 8: Performance comparison between models on the AZ-new-cat dataset.

5.4 Impact of Brand Entities

We also conducted experiments to explore the relationship between the number of entities that constitute the brand and the performance of the models. Since we use *Exact Match* criteria in our evaluations, detecting brand values with more than one entity becomes very challenging in general. We divide the test set of our AZ-base dataset into four subsets according to the number of entities inside a brand (see Figure 4). While OpenTag achieves good overall F_1 performance with brand values consisting of single entities (88%), it is much worse on brand values with three or more entities (67% and 61% respectively). OpenBrand, on the other hand, still performs well even on brands with two or more entities (71% and 65% respectively).

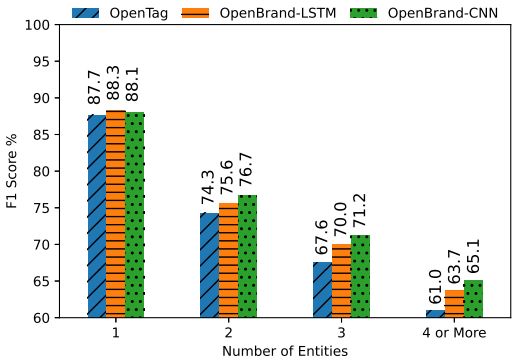


Figure 4: Impact of number of entities on the model performance.

5.5 Discussion

Our experimental results show that, for the task of extracting brand values, OpenBrand outperforms baseline approaches by a significant margin. Besides the general F1 score, the gains can be seen in both precision and recall which go up to 2.2% and 11.5%, respectively. This means that character embeddings do not only help discover more brand values but they also improve the accuracy of the extracted information. Furthermore, the gains in recall are also high for the AZ-new-cat and AZ-zero-shot datasets, reaching 3.3% and 1.46% of improvement respectively. Thus, OpenBrand performs particularly well for unseen data which confirms our initial claim that character embeddings enhance model generalizability.

Another important finding of our study is that the performance of OpenBrand depends on the product category. We can observe that, for the *Automotive* category, the gain in precision is 0.2% while it goes up to 2.2% for the *Toys & Games* category. This is mainly due to an ambiguity problem in the product descriptions of the *Automotive* category.

Some product descriptions might contain values of other brands other than the one that needs to be detected. Let us take the following product description: *"Honda Shadow 750 Aero Cobra Saddlebag Guards Supports"*. This is about a *"Saddlebag Guards Supports"* that is compatible for *"Honda"* cars. The brand of this product is *"Cobra"* but the presence of *"Honda"* in the description can be confusing for the model leading to wrong extractions.

We additionally observe that compound brand values are best handled by OpenBrand. This is due to the fact that the combination of character and word embeddings contributes to more meaningful representations. The results also show that OpenBrand-LSTM tends to perform worse, as compared to OpenBrand-CNN. This is inline with prior observations (Bradbury et al., 2017) that LSTM can be difficult to apply on long sequences of input.

6 Related Work

There has been significant research on the task of attribute-value extraction from product descriptions (Wong et al., 2009). Initial approaches (Vandic et al., 2012) formulated the problem as a classification task relying on supervised learning techniques. (Ghani et al., 2006) use a Naive Bayes classifier to extract values that correspond to a predefined set of product attributes. (Putthividhya and Hu, 2011) focus on annotating brands in product listings of apparel products on eBay. (Kovelamudi et al., 2011) propose a domain independent supervised system that can automatically discover product attributes from user reviews using Wikipedia. Similarly, (Ling and Weld, 2012) propose an automatic labeling process of entities by making use of anchor links from Wikipedia text. Other approaches exploited unsupervised learning techniques like (Shinzato and Sekine, 2013) in their task of extracting attribute-values from e-commerce product pages. Following a similar line, (Charron et al., 2016) use consumer patterns to create annotations for data-driven products. (Bing et al., 2016) focus on the discovery of hidden patterns in costumer reviews to improve attribute-value extraction. The above approaches provide promising results, however they poorly handle the discovery of new values due to their closed world assumption.

The most recent approaches (Kozareva et al., 2016; Zheng et al., 2018; Xu et al., 2019) make instead an open world assumption using sequence tagging models, similarly to NER tasks (Ma and Hovy, 2016; Huang et al., 2015). (Kozareva et al., 2016) use a BiLSTM-CRF model to tag several product attributes for brands and models with hand-crafted features. (Zheng et al., 2018) develop an end-to-end tagging model utilizing BiLSTM and CRF without using any dictionary or hand-crafted features. After that, (Xu et al., 2019) adopted only one global set of BIO tags for any attributes to scale up the semantic representation models of product titles. In this context, (Karamanolakis et al., 2020) proposed a taxonomy aware knowledge extraction model that takes advantage of the hierarchical relationships between product categories. The latest approaches extend the open world assumption also to attributes and use question answering (QA) models (Wang et al., 2020) to scale to a larger number of attributes. Sequence tagging approaches are the most relevant to our work since extracting brand names does not require scalability. However, these models did not exploit character level embeddings which are crucial for improving generalizability. In our work, we enhance such models using different granularities of embeddings.

7 Conclusion

In this paper we have addressed the problem of extracting brand values from product descriptions. Previous state-of-the-art sequence tagging methods faced the challenge of discovering new values that have not been encountered before. To tackle this issue we proposed OpenBrand, a novel attribute-value extraction model with the integration of character level representations to improve generalizability. We presented experiments on real-world datasets in different categories which show that OpenBrand outperforms state-of-the-art approaches and baselines. By exploiting character level embeddings, OpenBrand is capable of learning accurate representations to discover new brand values. Our experiments also show that CNN based representations outperform LSTM based representations in both performance and computation.

A natural extension of this work is to deal with the problem of disambiguation discussed in Section 5.5. To this end, we need to have more training data which helps understating the patterns in a better way. Moreover, we need to extend the tagging model to capture ambiguous product descriptions. This extension can be very important when brand values need to be extracted from other data sources other than concise product descriptions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lidong Bing, Tak-Lam Wong, and Wai Lam. 2016. Unsupervised extraction of popular product attributes from e-commerce web sites by considering customer reviews. *ACM Trans. Internet Technol.*, 16(2).

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. Quasi-recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Bruno Charron, Yu Hirate, David Purcell, and Martin Rezk. 2016. Extracting semantic information for e-commerce. In *The Semantic Web – ISWC 2016*, pages 273–290, Cham. Springer International Publishing.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4.

Henrieta Hrablik Chovanová, Aleksander Ivanovich Korshunov, and Dagmar Babčanová. 2015. Impact of brand on consumer behavior. *Procedia Economics and Finance*, 34:615–621. International Scientific Conference: Business Economics and Management (BEM2015).

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. TXtract: Taxonomy-aware knowledge extraction for thousands of product categories. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8489–8502, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sudheer Kovelamudi, Sethu Ramalingam, Arpit Sood, and Vasudeva Varma. 2011. Domain independent model for product attribute extraction from user reviews using Wikipedia. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1408–1412, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Zornitsa Kozareva, Qi Li, Ke Zhai, and Weiwei Guo. 2016. Recognizing salient entities in shopping queries. In *ACL*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 94–100. AAAI Press.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Duangmanee (Pew) Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 1557–1567, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for

machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Umer Shahzad, Salman Ahmad, Kashif Iqbal, Muhammad Nawaz, and Saqib Usman. 2014. Influence of brand name on consumer choice & decision. *IOSR Journal of Business and Management*, 16:72–76.

Keiji Shinzato and Satoshi Sekine. 2013. Unsupervised extraction of attributes and their values from product description. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1339–1347, Nagoya, Japan. Asian Federation of Natural Language Processing.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Damir Vandic, Jan-Willem Dam, and Flavius Frasincar. 2012. Faceted product search powered by the semantic web. *Decision Support Systems*, 53:425–437.

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. Learning to extract attribute value from product via question answering: A multi-task approach. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 47–55. ACM.

Yuk Wah Wong, Dominic Widdows, Tom Lokovic, and Kamal Nigam. 2009. Scalable attribute-value extraction from semi-structured text. In *ICDM Workshop on Large-scale Data Mining: Theory and Applications*.

Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.

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 & Data Mining*, KDD '18, page 1049–1058, New York, NY, USA. Association for Computing Machinery.

A Appendix

A.1 Implementation Details

Our models are implemented with Tensorflow[3] and Keras[4], and they are trained using TPUs on the cloud. We used the validation set of AZ-base to select the optimal hyper-parameters of our model, while the test set was used to report the final results. During training, optimization is performed with Adam optimizer (Kingma and Ba, 2015) using a $1e^{-3}$ initial learning rate. For all models, we employed pre-trained 100-dimensional word vectors from GloVe (Pennington et al., 2014). All models use a dropout layer (Srivastava et al., 2014) of size 0.3 both before and after the BiLSTM layer. The minibatch size is fixed to 128. The *BIO* tagging scheme is adopted. In the training process, we used the loss score on the validation set to assess model improvement. The models were trained for a total of 100 epochs, and early stopping was applied if there was no improvement for a period of 10 epochs. The average training time for each epoch was also recorded.

Hyper-parameter	**Value**
LSTM Units	{64, **128**, 256}
Character Embedding Size	{10, **30**, 50, 100}
Window Size	{3, 5, **10**}
Number of Filters	{10, **30**, 50}
Trainable Parameters	36420

Table 9: Hyper-parameters for OpenBrand-CNN model.

Hyper-parameter	**Value**
LSTM Units	{64, **128**, 256}
Character Embedding Size	{10, **30**, 50, 100}
Character LSTM Units	{10, **30**, 50, 100}
Trainable Parameters	526170

Table 10: Hyper-parameters for OpenBrand-LSTM model.

Tables 9 and 10 show the selected hyper-parameters in the CNN-based and LSTM-based models respectively, based on the performance on the validation set. These include the character embeddings dimension. The tables also show the total number of trainable parameters for each model. The difference in number of trainable parameters shows that CNNs have less training complexity as compared to LSTM models under similar experimental settings.

[3] https://www.tensorflow.org/.
[4] https://keras.io/.