Toy Matching in E-commerce

Kai Lin
kl5ev@virginia.edu

Erzhen Hu
eh2qs@virginia.edu

Yinzhu Jin
yj3cz@virginia.edu

Fuxiao Liu
fl3es@virginia.edu

Abstract

This project is to detect categories from the toy product description and names. The data set is an Amazon (toy) data set with manufacturer specific model, names and description of children's toys. The expected outcome includes labeling scheme and using CRF and bi-LSTM to measure the performance of our category extraction. As for the annotation, we performed the labeling and used product description decomposed into sequence of tokens labeled with BIO encoding, and the output of learning algorithm on a product description would be a sequence of labels.

Three diffrent kinds of methods are used for our task. For the first method, We tried several traditional machine learning models like svm, logistic regression, and the linear svm for the second task. Linear svm gets the highest classification accuracy. Second model is CRF with hand-crafted features. And the last model is bi-directional LSTM. Note that there are too many categories in such a small dataset, which explains why even the best model results in relatively low accuracy.

1 Problem Statement

This project presents a name entity extraction to detect categories from the toy product description and names. The data set is an Amazon (toy) data set with manufacturer specific model, names and description of children's toys. The problem we are going to solve is to figure out how to form groups of identical categories sold by different sellers, and on the other hand, we want to be precise in the creation of categories to ensure they obtain identical products sold by different sellers. Hence, if the categories are formed by extraction, the model will present the customers exactly one item page with a single title, description and price. But if the groups are not homogeneous, then a product may be showed on a page that it does not belong to.
To make the process specific, let x be a product description and let $(x_1, x_2, .., x_n)$ be a particular tokenization x_t of x. Given a category η, our task for the second method is to discover the function:

$$f(x_t) = f(x_1, x_2, ..x_n) = (x_i, x_{i+1}, ..x_k)$$

,where $(x_i, x_{i+1}, ...x_k)$ is the tokenization of η. And for other methods, we need to discover the function:

$$g(x_t) = g(x_1, x_2, ..x_n) = \eta$$

If we look at the product description, for example, in the toy data

$$"WhistleBalloonsColour : AssortedSize : 35cm/14" Packaging : 10pcs"$$

, We will seek to output the categories

$$f(x_1, x_2, ..x_9) = (x_2) = (Balloons)$$

We build the training set consisting of the product descriptions and names, annotated with a category attribute value, and compare Conditional Random Fields, Hidden Markov models, etc to a number of baseline machine learning approaches.

Preprint. Under review.

2 Proposed Method

2.1 Literature review

Our literature review includes name entity recognition application with feature selection, as well as the utilization of Deep Recurrent Neural Networks in eCommerce compared with traditional machine learning methods. The Name Entity Recognition(NER) task can be used beyond identifying people, location. Many recent studies (Putthividhya and Hu,2011;More,2016) focused on extracting product attributes from online E-Commerce data. One of the popular generative model based classifiers for NER is Hidden Markov Model, which captures the temporal states by modeling transitions over time. However, methods like SVM for NER have been shown to outperform generative model. Moreover, Conditional Random Fields(Feng and McCallum 2004) has been proposed for sequence labeling problem for NER. Additionally, Majumder,et al.(2018) demonstrated the potential of Bidirectional LSTM-CRF and Bidirectional LSTMs in E-commerce domainwhich shows the efficacy of these deep recurrent models over previous machine learning benchmarks. To capture the existing grammatical structure of E-commerce data in both product title and product description, some studies(Putthividhya and Hu,2011) used the boostrapped Named Entity Recognition to identify new brands corresponding to spelling variants and some typographical errors.
There are still challenges in product descriptions that, these descriptions may contain lots of named entities unrelated to the product. The number of different attributes that are used by a general retailer to describe products is very large.

2.2 Propose our Overall framework

Here we show how we define the metrics used to measure the performance of the category extraction algorithm.

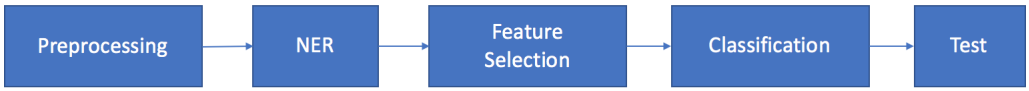
Method1: Word embedding using pretrained glove 50-d vectors concatenate word bag vectors with classification methods(logistic regression, svm, NB)

Method2: Here we implement the CRF model with extracted features(word parts, POS tags, lower/title/upper flags, features of nearby words), the NER tag (B-category name, I-category name, O)+linear-chain CRF, we got the state features and transition features from the CRF model, we set 3-fold cross validation to get optimized results.

Method3: Bi-LSTM: word embedding using pretrained glove 50-d vectors, tag embedding initialized with one-hot representation, then learned along the training.

For the preprocessing part, we have extracted the last hierarchy of the Amazon categories' hierarchical structure and marked 169 categories with frequency less than 25 as a new category, "Others". we also implemented train test split on this phase.

As shown in figure below, This part will introduce our overall framework.



2.3 Relationships between the results and proposed project

We have figured out our overall design of the whole project and now we are moving forward as expected. We have finished pre-processing the Amazon (toy) data set and extracted the columns we need, and applied several NER methods. For example, using NLTK and spacy, even though the current analysis of the noun did not perform well, we continue figuring out how to improve the accuracy.

3 Expected Outcome

The expected outcome includes labeling scheme and use CRF and bi-LSTM to measure the performance of our category extraction. As for the annotation, we are doing the labeling and used a product description decomposed into a sequence of tokens labeled with BIO encoding, and the output of learning algorithm on a product description would a sequence of labels.
B: beginning token of the category name
I: Intermediate token of a category name. If the length of the sub-sequence is greater than 1, then the label of each token except the first will be labeled as I-category.
O: Indicates that the token is not part of the category/brand name
The post-processing might be implemented to avoid zero output for the categories.
In the final report, we will report the output and compared Conditional Random Fields, Hidden Markov, etc for model performance using F_1 measure.

4 The Progress Of The Proposed Project

4.1 Preprocessing

Amazon categories have hierarchical structure. As an example, one of the categories is "Hobbies > Model Trains Railway Sets > Rail Vehicles > Trains". We consider only the last hierarchy for simplicity, i.e. "Trains" in this example. Samples with missing category are deleted. Our data-set contains 9312 valid samples with 236 different categories. Many of categories only have a few corresponding samples. This may cause significant unbalance for experiments. Therefore, we marked 169 categories with frequency less than 25 as a new category, "Others". This new category has 685 samples, which is reasonable considering the dataset size.

The next step is to split dataset into training set, validation set and test set. To ensure the category distribution is similar in each set, we performed stratified sampling. Training set roughly contains 60% samples, and validation set and test set each contains about 20% samples.

We represent the input text in two different ways. First method is traditional Bag of Words representation model. For those words with very low frequencies, we mark them as "UNK". For the neural models and some other machine learning models, we use GloVe word embedding vectors. It is popular choice for NLP task via deep learning. Since our dataset is not big enough, we will use pre-trained GloVe embedding.

4.1.1 NER and Feature Selection

We have now dig deeper into Hidden Markov Chain, CRF, Hidden Markov Chain; CRF; Bi-LSTM; CNN With k-fold.

4.1.2 results and outcome

Table1 shows basic statistics of prepossessed data. As an example, one of samples is:
"*Trains; CLASSIC TOY TRAIN SET TRACK CARRIAGES LIGHT ENGINE BOXED BOYS KIDS BATTERY; Technical Details Manufacturer recommended age:3 years and up. Additional Information ASINB00E5MNXJ4 ...*"
The second and the third column of samples are product name and description, which will be used as input for the model. The first column is the category, the label our model aims to predict.

Dataset	Sample number
training set	5555
validation set	1864
test set	1893
total	9312

Table 1: Dataset Statistics

Figure 1 shows the NLTK pos tag for one line of record of Product Description. Figure2 shows tree structure of the pos tag. Figure 3 shows the spacy for one record, although current precision is not

well enough, we are continue exploring how to improve the precision. We use NLTK and SpaCy to achieve Named Entity Recognition.

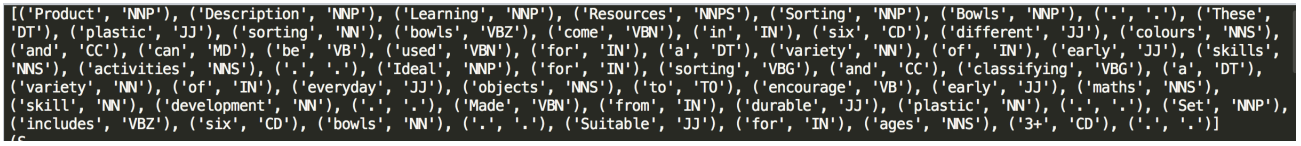


Figure 1: NLTK pos tag

Figure 2: treestructure

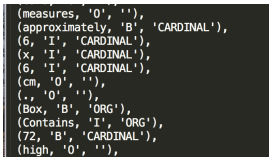


Figure 3: Spacy

4.2 Text classification with pre-trained word embedding

Here, we tried word embedding using pretrained glove 50-d vectors concatenate word of bag vectors with SVM of different kernels(linear,rbf,etc), logistic regression, Naive Bayes, etc. the results showed that linear SVM get the best result as shown in the Figure 4

Model	Precision	Recall	F1score	Time
SVM (kernel='linear', c=0.1)	0.309	0.26131479	0.27502532	0.2s

Figure 4: Results of SVM

4.3 NER with Conditional Random Fields (3-fold cross validation)

We tried linear-chain CRF here for the pre-defined categories, the results are shown in Figure 5, the transition and state features from Conditional Random fields has been shown in Figure 8

4.4 Bi-LSTM with pretrained glove 50d vectors

Majumder,et al.(2018) demonstrated the potential of Bidirectional LSTM-CRF and Bidirectional LSTMs in E-commerce domain which shows the efficacy of these deep recurrent models over previous machine learning benchmarks, here we tried Bi-LSTM but the preliminary results is not that good.

Model		Precision	Recall	F1_score	Time
CRF with cv ('c1':0.42139673260345256,'c2':0.0067941467019329285 (Best CV score: 0.9367260174060519)	Micro average	0.954	0.955	0.955	1h 32min 3s (150 fits)
	Macro average	0.247	0.154	0.176	
CRF with cv ('c1': 0.00001, 'c2': 0.0067941467019329285 (Decreasing c1 to get l1 regularization)	Micro average	0.998	0.998	0.998	4min 20s
	Macro average	0.319	0.205	0.236	

Figure 5: Results of CRF

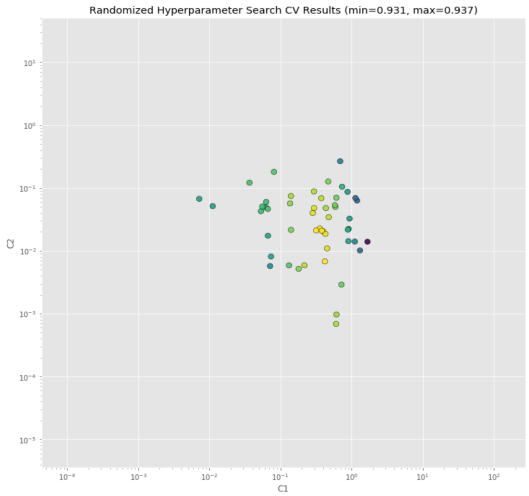


Figure 6: Cross validation results (the darker the better)

From \ To	O	B-Chess	B-Hand Puppets
O	2.572	0.029	-0.014
B-Chess	0.826	-0.135	0.0
B-Hand Puppets	0.074	0.0	-1.335

y=O top features		y=B-Chess top features		y=B-Hand Puppets top features	
Weight[?]	Feature	Weight[?]	Feature	Weight[?]	Feature
+7.708	postag:NNS	+4.051	word.lower():chess	+4.369	+1:word.lower():wry
+6.323	postag:NNP	+3.019	+1:word.lower():chess	+3.337	-1:word.lower():art
+6.006	bias	+2.974	-1:word.lower():champion	+3.119	-1:word.lower():over
+5.402	postag:NNPS	+2.452	-1:word.lower():woods	+2.970	-1:word.lower():take
+4.615	-1:postag:NN	+2.413	-1:word.lower():?	+2.827	-1:word.lower():giraffe
… 876 more positive …		+2.147	-1:word.lower():chess	+2.811	-1:word.lower():curiosity
… 873 more negative …		+1.978	+1:postag:VBN	+2.791	+1:word.lower():art
-2.939	-1:word.lower():chapter	+1.936	+1:word.lower():recognized	+2.773	+1:word.lower():monkey
-2.967	+1:word.lower():beaten	+1.775	+1:word.lower():men	+2.757	word.lower():dimension
-2.968	+1:word.lower():d6	+1.745	word[-2:]:ss	+2.719	-1:word.lower():dad
-3.652	word.lower():spinning	… 69 more positive …		… 153 more positive …	
-4.510	postag[:2]:NN	… 5 more negative …		… 14 more negative …	

State features		
4.368741	B-Hand Puppets	+1:word.lower():wry
4.164948	B-Packs & Sets	+1:word.lower():heartgold
3.996605	B-Art Sand	+1:word.lower():magic
3.985012	I-Dolls	+1:word.lower():soft
3.977346	B-Balloons	+1:word.lower():balloons
3.857763	B-Card Games	+1:word.lower():facet
3.810840	B-Slot Cars	-1:word.lower():mm
3.704961	B-Dice & Dice Games	+1:word.lower():d6
3.391243	B-Card Games	-1:word.lower():u.s

Figure 7: The model learns that token “champion” ,”woods”, ”recognized” ,”supplied”, ”officially” is likely to be at the beginning of the category “chess”; wry could be at the beginning of category Hand Puppets

Model	Precision	Recall	F1score
Bi- ISTM	0.003	0.014	0.003

Figure 8: Results of the Bi-LSTM

5 Discussion and Future Work

We tried several different methods. The first one is using pre-trained word embedding to improve the classification performance with traditional machine learning models. Here we tried several models like svm, logistic regression, and the linear svm. SVM performs better than other text classification methods after pre-trained word embedding. The second one is conditional random fields with hand-crafted features. It gets a high micro average $f1_score$ but a low macro average $f1_score$ due to the zeros of many categories, decreasing the C1 might be a good way but will influence the model quality here. Bi-directional LSTM with word embedding and POS tag as input did not perform good in this experiment. The main challenge for us is still that there are too many categories in such a small dataset. Also, the category distribution is not balanced. The reason why the SVM performs better than other text classification methods with pre-trained word embedding might be that sentences are not grammatically complete, e.g. (The Hobbit - An Unexpected Journey - Bilbo Baggins Bobble Head Figure) As for the future work, since it was a sub-set from Amazon dataset, it would be better if we could try on larger datasets to compare the mdoels. The absence of syntactic structure in such short pieces of text makes extracting attribute values a challenging problem.

References

[1] Putthividhya, D.P. and Hu, J., 2011, July. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 1557-1567).* Association for Computational Linguistics.

[2] More, A., 2016. Attribute extraction from product titles in ecommerce. *arXiv preprint arXiv:1608.04670*.

[3] Peng, F., Feng, F. and McCallum, A., 2004, August. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics (p. 562).* Association for Computational Linguistics.

[4] Majumder, B.P., Subramanian, A., Krishnan, A., Gandhi, S. and More, A., 2018. Deep Recurrent Neural Networks for Product Attribute Extraction in eCommerce. *arXiv preprint arXiv:1803.11284*.