Bootstrapping Named Entity Recognition in E-Commerce with Positive Unlabeled Learning

Hanchu Zhang[1] **Leonhard Hennig**[2] **Christoph Alt**[2] **Changjian Hu**[1]
Yao Meng[1] **Chao Wang**[1]

[1]Lenovo AI [2]German Research Center for Artificial Intelligence
{zhanghc9,hucj1,mengyao1,wangchao31}@lenovo.com
{leonhard.hennig,christoph.alt}@dfki.de

Abstract

Named Entity Recognition (NER) in domains like e-commerce is an understudied problem due to the lack of annotated datasets. Recognizing novel entity types in this domain, such as products, components, and attributes, is challenging because of their linguistic complexity and the low coverage of existing knowledge resources. To address this problem, we present a bootstrapped positive-unlabeled learning algorithm that integrates domain-specific linguistic features to quickly and efficiently expand the seed dictionary. The model achieves an average F1 score of 72.02% on a novel dataset of product descriptions, an improvement of 3.63% over a baseline BiLSTM classifier, and in particular exhibits better recall (4.96% on average).

1 Introduction

The vast majority of existing named entity recognition (NER) methods focus on a small set of prominent entity types, such as persons, organizations, diseases, and genes, for which labeled datasets are readily available (Tjong Kim Sang and De Meulder, 2003; Smith et al., 2008; Weischedel et al., 2011; Li et al., 2016). There is a marked lack of studies in many other domains, such as e-commerce, and for novel entity types, e.g. products and components.

The lack of annotated datasets in the e-commerce domain makes it hard to apply supervised NER methods. An alternative approach is to use dictionaries (Nadeau et al., 2006; Yang et al., 2018), but freely available knowledge resources, e.g. Wikidata (Vrandečic and Krötzsch, 2014) or YAGO (Suchanek et al., 2007), contain only very limited information about e-commerce entities. Manually creating a dictionary of sufficient quality and coverage would be prohibitively expensive. This is amplified by the fact that in the e-commerce domain, entities are frequently expressed as complex noun phrases instead of proper names. Product and component category terms are often combined with brand names, model numbers, and attributes (*"hard drive"* → *"SSD hard drive"* → *"WD Blue 500 GB SSD hard drive"*), which are almost impossible to enumerate exhaustively. In such a low-coverage setting, employing a simple dictionary-based approach would result in very low recall, and yield very noisy labels when used as a source of labels for a supervised machine learning algorithm. To address the drawbacks of dictionary-based labeling, Peng et al. (2019) propose a positive-unlabeled (PU) NER approach that labels positive instances using a seed dictionary, but makes no label assumptions for the remaining tokens (Bekker and Davis, 2018). The authors validate their approach on the CoNLL, MUC and Twitter datasets for standard entity types, but it is unclear how their approach transfers to the e-commerce domain and its entity types.

Contributions We adopt the PU algorithm of Peng et al. (2019) to the domain of consumer electronic product descriptions, and evaluate its effectiveness on four entity types: *Product*, *Component*, *Brand* and *Attribute*. Our algorithm bootstraps NER with a seed dictionary, iteratively labels more data and expands the dictionary, while accounting for accumulated errors from model predictions. During labeling, we utilize dependency parsing to efficiently expand dictionary matches in text. Our experiments on a novel dataset of product descriptions show that this labeling mechanism, combined with a PU learning strategy, consistently improves F1 scores over a standard BiLSTM classifier. Iterative learning quickly expands the dictionary, and further improves model performance. The proposed approach exhibits much better recall than the baseline model, and generalizes better to unseen entities.

Algorithm 1: Iterative Bootstrapping NER

Input: Dictionary D_{seed}, Corpus C, threshold K, max_iterations I
Result: Dictionary D^+, Classifier L

$D^+ \leftarrow D_{seed}$;
$C_{dep} \leftarrow dependency\_parse(C)$;
$i \leftarrow 0$;
while *not_converged*(D^+) *and* $i < I$ **do**
  $C_{lab} \leftarrow label(C, D^+)$;
  $C_{exp} \leftarrow expand\_labels(C_{lab}, C_{dep})$;
  $L \leftarrow train\_classifier(C_{exp})$;
  $C_{pred} \leftarrow predict(C_{exp}, L)$;
  for $e \leftarrow C_{pred}$ **do**
    if $e \notin D^+$ *and freq(e)* $> K$ **then**
      $D^+ \leftarrow add\_entity(D^+, e)$;
    end
  end
  $i \leftarrow i + 1$;
end

2 NER with Positive Unlabeled Learning

In this section, we first describe the iterative bootstrapping process, followed by our approach to positive unlabeled learning for NER (PU-NER).

2.1 Iterative Bootstrapping

The goal of iterative bootstrapping is to successively expand a seed dictionary of entities to label an existing training dataset, improving the quality and coverage of labels in each iteration (see Algorithm 1). In the first step, we use the seed dictionary to assign initial labels to each token. We then utilize the dependency parses of sentences to label tokens in a "compound" relation with already labeled tokens (see Figure 1). In the example "hard drive" is labeled a *Component* based on the initial seed dictionary, and according to its dependency parse it has a "compound" relation with "dock", which is therefore also labeled as a *Component*. We employ an IO label scheme, because dictionary entries are often more generic than the specific matches in text (see the previous example), which would lead to erroneous tags with schemes such as BIO.

In the second step, we train a NER model on the training dataset with new labels assigned. We repeat these steps at most I times, and in each subsequent iteration we use the trained model to predict new token-level labels on the training data. Novel entities predicted more than K times are included in the dictionary for the next labeling step. The threshold K ensures that we do not introduce noise in the dictionary with spurious positively labeled entities.

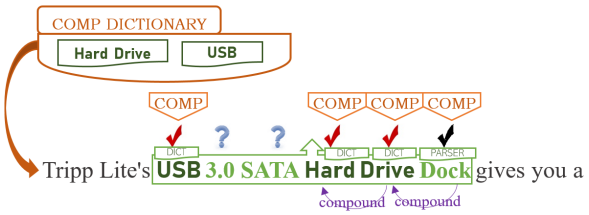

Figure 1: Red check marks indicate tokens labeled by the dictionary, black those based on label expansion using dependency information. The green box shows the true extent of the multi-token *Component* entity.

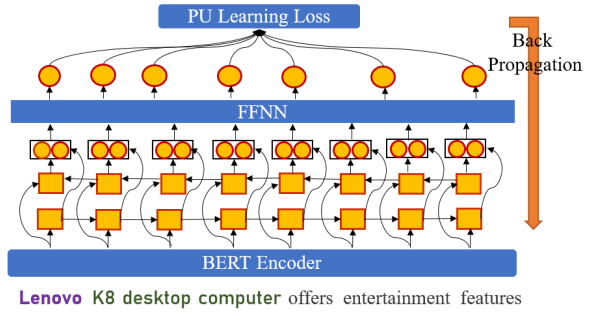

Figure 2: Architecture of the positive unlabeled NER (PU-NER) model.

2.2 PU-NER Model

As shown in Figure 2, our model first uses BERT (Devlin et al., 2018) to encode the sub-word tokenized input text into a sequence of contextualized token representations $\{z_1, ..., z_L\}$, followed by a bidirectional LSTM (Lample et al., 2016) layer to model further interactions between tokens. Similar to Devlin et al. (2018), we treat NER as a token-level classification task, without using a CRF to model dependencies between entity labels. We use the vector associated with the first sub-word token in each word as the input to the entity classifier, which consists of a feedforward neural network with a single projection layer. We use back propagation to update the training parameters in bi-LSTM and final classifier, without fine-tune BERT model.

Dictionary-based labeling achieves high precision on the matched entities but low recall. This fits the positive unlabeled setting (Elkan and Noto, 2008), which assumes that a learner only has access to positive examples and unlabeled data. Thus, we consider all tokens matched by the dictionary as positive, and consider all other tokens to be unlabeled. The goal of PU learning is then to estimate

the true risk regarding the expected number of positive examples remaining in the unlabeled data. We define the empirical risk as $\hat{R}_l = \frac{1}{n}\sum_i^n l(\hat{y}_i, y_i)$ and assume the class prior to be equal to real distribution of examples in the data $\pi_p = P(Y = 1)$, and $\pi_n = P(Y = 0)$. As the model tends to predict the positive labels correctly during training, i.e. $l(\hat{y_i}^p, 1)$ declines to a small value. We follow Peng et al. (2019) and combine risk estimation with a non-negative constraint:

$$\hat{R}_l = \frac{1}{n_p}\sum_i^{n_p} l(\hat{y_i}^p, 1) + max\left(0, \frac{1}{n_u}\sum_i^{n_u} l(\hat{y_i}^u, 0) - \frac{\pi_p}{n_p}\sum_i^{n_p} l(\hat{y_i}^p, 0)\right)$$

3 Dataset

E-commerce covers a wide range of complex entity types. In this work, we focus on electronic products, e.g. personal computers, mobile phones, and related hardware, and define the following entity types: **Products**, i.e. electronic consumer devices such as mobiles, laptops, and PCs. *Products* may be preceded by a brand and include some form of model, year, or version specification, e.g. "Galaxy S8" or "Dell Latitude 6400 multimedia notebook". **Components** are parts of a product, typically with a physical aspect, e.g. "battery", or "multimedia keyboard".[1] **Brand** refers to producers of a *product* or *component*, e.g. "Samsung", or "Dell". **Attributes** are units associated with components, e.g. size ("4 TB"), or weight ("3 kg").

To create our evaluation dataset, we use the *Amazon review dataset* (McAuley et al., 2015),[2] a collection of product metadata and customer reviews from Amazon. The metadata includes product title, a descriptive text, category information, price, brand, and image features. We use only entries in the *Electronics/Computers* subcategory and randomly sample product descriptions of length 500–1000 characters, yielding a dataset of 24,272 training documents. We randomly select another 100 product descriptions to form the final test set. These are manually annotated by 2 trained linguists, with disagreements resolved by a third expert annotator. Token-level inter-annotator agreement was high (Krippendorf's $\alpha = 0.7742$). The test documents contain a total of $27,108$ tokens ($1,493$ *Product*, $3,234$ *Component*, $1,485$ *Attribute*, and 443 *Brand*).

4 Experiments

To evaluate our proposed model (*PU*), we compare it against two baselines: (1) dictionary-only labeling (*Dictionary*), and (2) our model with standard cross-entropy loss instead of the PU learning risk (*BiLSTM*). The *BiLSTM* model is trained in a supervised fashion, treating all non-dictionary entries as negative tokens. The *BiLSTM* and *PU* models were implemented using AllenNLP (Gardner et al., 2018). We use SpaCy[3] for preprocessing, dependency parsing, and dictionary-based entity labeling. We manually define seed dictionaries for *Product* (6 entries), *Component* (60 entries) and *Brand* (13 entries). For *Attribute*s, we define a set of 8 regular expressions to pre-label the dataset. Following previous works, we evaluate model performance using token-level F1 score.

For the estimation of π_p, there are two options to get this prior parameter. The simple way is to treat π_p as constant hyper-parameter, which would not change during the training iterations. Another possible way is suggested in 's work, use a selected value to start bootstraping, then calculate π_p based on the prediction results that produced by model after several iterations. In our work, we choose the former way that make $\pi_p = 0.01$ as a hyper-parameter.

4.1 Results and Discussion

Table 1 shows the F1 scores of several model ablations by entity type on our test dataset. For the iterative experiments, we conduct the iteration 10 times, then collect the best F1 score from evaluation results sequence as one's final score. From the table, we can observe: 1) The PU algorithm outperforms the simpler models for most classes, which demonstrates the effectiveness of the PU learning framework for NER in our domain. 2) Dependency parsing is a very effective feature for *Component* and *Product*, and it strongly improves the overall F1 score. 3) The iterative training strategy yields a significant improvement for most classes. Even after several iterations, it still finds new entries to expand the dictionaries (Figure 3).

[1] Non-physical product features and software, such as "Toshiba Face Recognition Software", or "Windows 7" are not considered as components.

[2] http://jmcauley.ucsd.edu/data/amazon/links.html

[3] https://spacy.io/

Entity Type	Dictionary	BiLSTM	PU	PU+Dep	PU+Iter	PU+Dep+Iter
Component	46.19	65.98	66.89	67.38	68.67	**70.66**
Product	16.78	60.23	60.24	65.05	60.24	**67.07**
Brand	49.74	74.06	74.84	**76.24**	**76.24**	**76.24**
Attribute	7.05	73.30	73.84	**74.14**	**74.14**	**74.14**
All	29.94	68.39	68.95	70.70	69.82	**72.02**

Table 1: Token-Level F1 scores on the test set. The unmodified PU algorithm achieves an average F1 score of 68.95%. Integrating dependency parsing (Dep) and iterative relabeling (Iter) raises the F1 score to 72.02%, an improvement of 42.08% over a dictionary-only approach, and 3.63% over a BiLSTM baseline.

The *Dictionary* approach shows poor performance on average, which is due to the low recall caused by very limited entities in the dictionary. *PU* greatly outperforms the dictionary approach, and has an edge in F1 score over the *BiLSTM* model. The advantages of PU gradually accumulate with each iteration. For *Product*, the combination of PU learning, dependency parsing-based labeling, and iterative bootstrapping, yields a 7% improvement in F1 score, for *Component*, it is still 5%.

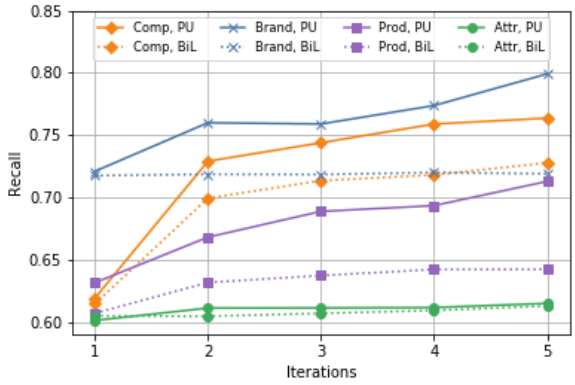

Figure 3: Recall curves of the BiLSTM+Dep and PU+Dep model for *Component*, *Product*, *Brand*, and *Attribute*. PU+Dep boosts recall by 3.03% on average, with a max average difference of 4.96% after 5 iterations.

PU Learning Performance Figure 3 shows that the *PU* algorithm especially improves recall over the baseline classifier for *Components*, *Products* and *Brands*. With each iteration step, the PU model is increasingly better able to predict unseen entities, and achieves higher recall scores than the BiLSTM model. While the baseline curve on *Brands* stays almost flat during iterations, *PU* consistently improves recall as new entities are added into dictionary. For *Attributes*, however, both models exhibit about the same level of recall, which in addition is largely unaffected by the number of iterations.

This suggests that PU learning better estimates the true loss in the model. In a fully supervised setting, a standard classification loss function can accurately describe the loss on positive and negative samples. However, in the positive unlabeled setting, many unlabeled samples may actually be positive, and therefore the computed loss should not strongly push the model towards the negative class. We therefore want to quantify how much the loss is overestimated due to false negative samples, so that we can appropriately reduce this loss using the estimated real class distribution.

Error Analysis Both PU and the baseline model in some cases have difficulties predicting *Attribute*s correctly. This can be due to spelling differences between train and test data (e.g. ”8 Mhz” vs ”8Mhz”), but also because of unclean texts in the source documents. Another source of errors is the fixed word piece vocabulary of the pre-trained BERT model, which often splits unit terms such as ”Mhz” into several word pieces. Since we use only the first word piece of a token for prediction, this means that signals important for prediction of the *Attribute* class may get lost. This suggests that for technical domains with very specific vocabulary, tokenization is important to allow the model to better represent the meaning of each word piece.

5 Related work

Recent work in positive-unlabeled learning in the area of NLP includes deceptive review detection (Ren et al., 2014), keyphrase extraction (Sterckx et al., 2016) and fact check-worthiness detection (Wright and Augenstein, 2020), see also (Bekker and Davis, 2018) for a survey. Our approach extends the work of Peng et al. (2019) in a novel domain and for challenging entity types. In the area of NER for e-commerce, Putthividhya and Hu (2011) present an approach to extract prod-

uct attributes and values from product listing titles. Zheng et al. (2018) formulate missing attribute value extraction as a sequence tagging problem, and present a BiLSTM-CRF model with attention. Pazhouhi (2018) studies the problem of product name recognition, but uses a fully supervised approach. In contrast, our method is semi-supervised and uses only very few seed labels.

6 Conclusion

In this work, we introduce a bootstrapped, iterative NER model that integrates a PU learning algorithm for recognizing named entities in a low-resource setting. Our approach combines dictionary-based labeling with syntactically-informed label expansion to efficiently enrich the seed dictionaries. Experimental results on a dataset of manually annotated e-commerce product descriptions demonstrate the effectiveness of the proposed framework.

References

Jessa Bekker and Jesse Davis. 2018. Learning from positive and unlabeled data: A survey. *CoRR*, abs/1811.04820.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

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.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database : the journal of biological databases and curation*, 2016.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

David Nadeau, Peter D. Turney, and Stan Matwin. 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Advances in Artificial Intelligence*, pages 266–277, Berlin, Heidelberg. Springer Berlin Heidelberg.

Elnaz Pazhouhi. 2018. Automatic product name recognition from short product descriptions. Master's thesis, University of Twente.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419, Florence, Italy. Association for Computational Linguistics.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Yafeng Ren, Donghong Ji, and Hongbin Zhang. 2014. Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 488–498, Doha, Qatar. Association for Computational Linguistics.

Larry Smith, Lorraine K. Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M. Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A. Struble, Richard J. Povinelli, Andreas Vlachos, William A. Baumgartner, Lawrence E. Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel J. Maña-López, Jacinto Mata, and W. John Wilbur. 2008. Overview of biocreative ii gene mention recognition. *Genome Biology*, 9:S2 – S2.

Lucas Sterckx, Cornelia Caragea, Thomas Demeester, and Chris Develder. 2016. Supervised keyphrase extraction as positive unlabeled learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1924–1929, Austin, Texas. Association for Computational Linguistics.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA. Association for Computing Machinery.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Denny Vrandečic and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. *OntoNotes: A Large Training Corpus for Enhanced Processing*.

Dustin Wright and Isabelle Augenstein. 2020. Fact check-worthiness detection as positive unlabelled learning. *ArXiv*, abs/2003.02736.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Guineng Zheng, Subhabrata Mukherjee, Xin Dong, and Feifei Li. 2018. Opentag: Open attribute value extraction from product profiles. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.