Corpus Development for Indonesian Product Named Entity Recognition using Semi-supervised Approach

Muhammad Akmal
School of Computing
Telkom University
Bandung, Indonesia
dziemboh@gmail.com

Ade Romadhony
School of Computing
Telkom University
Bandung, Indonesia
aderomadhony@telkomuniversity.ac.id

Abstract

We present a study on developing a corpus for Indonesian Product Named Entity Recognition (PRONER). We labeled a small amount of data and implemented a semi-supervised learning approach to label the rest of the data. We used conditional random fields (CRF) as the classifier. The experimental result shows that the corpus accuracy on brand, product type, and product are 89.37%, 44.05%, and 70.49%. The performance on very similar vocabularies is quite good, while we conclude that we need to seek better features and semi-supervised method to recognize unknown tokens.

1 Introduction

Named Entity Recognition (NER) is a part of Information Extraction (IE) that is used to extract entities from a text. A NER system has many advantages, including in market intelligence field, where we can extract Product Named Entity based on a NER approach. Product NER (PRONER) is a NER system that is aimed to recognize product entities in a text. Generally, there are two approaches to accomplish the task of both NER and PRONER. One approach is rule based approach such as (Farmakiotou et al., 2000), and (Kim and Woodland, 2000) and the other approach is machine learning approach such as (Zhao and Liu, 2008), and (Liao and Veeramachaneni, 2009). The machine learning approach need large labeled dataset for training. However, not many labeled datasets are available, especially domain-specific datasets and dataset in other languages than English. In this study, we performed a PRONER labeled corpus development using semi-supervised approach, with Conditional Random Field (CRF) as the method in automatic labeling process.

2 Related Work

Previous work on Indonesian NER corpus includes (Alfina et al., 2017) and (Luthfi et al., 2014), however we could not use their corpus since both corpus is not a PRONER corpus. Related work on bootstrapping PRONER by (Zhang et al., 2020) applies positive unlabeled learning to recognize product named entity in a low-resource setting. (Putthividhya and Hu, 2011) used bootstrapping method to expand brand dictionary for PRONER. (Liao and Veeramachaneni, 2009) proposed a semi-supervised learning on news data using high confidence label and low confidence label. Our work is similar to (Liao and Veeramachaneni, 2009), the difference is in the high confidence label selection. We used high confidence label that is predicted at least twice instead off all high confidence label.

3 Materials and Methods

We used three labels: product (PRO), brand (BRA), and type (TYP), following the labels in (Zhao and Liu, 2008). We used Beginning Inside Outside (BIO) encoding and we allow a token to have multiple labels as shown in Example 1.

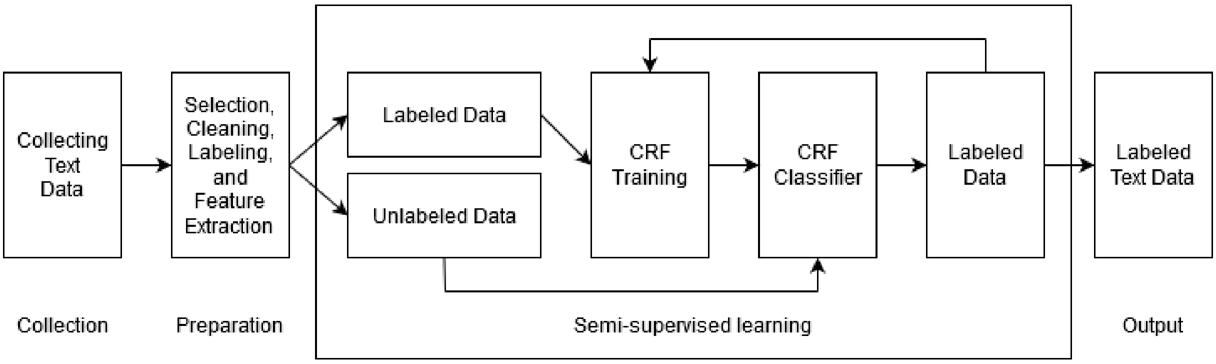

Figure 1: Indonesian PRONER corpus development

Example 1:
pake (use)/O shampoo/B-PRO ciment/I-PRO|B-TYP thermique/I-PRO|I-TYP nya ('s)/I-PRO kerastase/I-PRO|B-BRA
(use kerastase's ciment thermique shampoo)

Figure 1 shows the overview of our proposed corpus development process. The process starts from collecting post data from various threads on `https://forum.femaledaily.com/`. The preparation process start with selecting all post data that contains at least one product entity and then cleaning it from unrelated token such as emoji, link, and image. The labelling process is done by one person and on 15% of the data, while the 85% other data is left unlabeled. We used the standard CRF classifier and we manually designed the feature space from both train and test data. The feature space is listed below:

- Beginning and end of sentence boolean feature.
- Lowercased current token, previous token, and next token and their orthographic information.
- Brand and product category list lookup in window size 1 using edit distance features for brand and brand abbreviation, and item category such as respectively. The edit distance feature is similar to the one used in (Tsuruoka and Tsujii, 2003).
- A list of token that indicates a list of products or brand or type such as hyphen, star symbol, number followed by dot, and comma.
- A list of token that usually precedes a PRO NE and all of its variation such as "pakai" (use), "pake" (use), and "dari" (from).
- Features of neighbors in window size 2. For example, if the previous token is in the brand list, then this token would have a feature "-1:neigbourFeatures: Brand".
- Joint features. This feature is a combination of individual features from a token and its neighbor. For example, if a token is in the brand list and the previous token indicates a list, then this token would have a feature "jointFeatures: ListIndication+Brand".

The CRF classifier is trained on the labeled data and then used to label all unlabeled data. If a sequence of tokens in a sentence has been classified as BRA/PRO/TYP with high confidence [1] and there is at least one other identical occurrence with the same label, we replace the label of the same sequence or any of its subset with low confidence[2] label to be the same label. All sentences that contain such replaced label is then added to the training data and re-train the classifier. The sentences that contain high confidence label sequence and replaced label is then written to the output file. This semi-supervised learning process is repeated until less than 50 high confidence labels are found. Finally, the rest of the data is written to the output file.

[1] The marginal probability of each token in the sequence is ≥ 0.99 for more accurate label.

[2] The average marginal probability of all token in the sequence is ≤ 0.8 for more quantity.

4 Experiments

There are 2,275 manually annotated sentences and 16,853 unlabeled sentences. We randomly split the manually annotated data into 1,890 sentences as training data and 385 sentences as testing data. First we evaluate the baseline by using manually labeled data the training and we evaluate the effect of brand list lookup features on the baseline. The result is shown in Table 1 and it shows that there is no label bias effect from this feature and it improves performance on all label. Afterward, we run the semi-supervised learning. Figure 2 illustrates how the classifier evolve after each iteration and Table 2 shows the classifier performance after semi-supervised learning. The semi-supervised learning slightly improves the classifier and the performance on each label declines at least once during the process. Finally we used both manually labeled and automatically labeled data as training data to evaluate the quality of the corpus and the result is shown in Table 3. We randomly selected 150 sentences from the automatically labeled data and performed error analysis on it. We found 162 labeling error cases on the random 150 automatically labeled data. The most frequent errors were found on TYP label, which we found 35 entities unlabeled, 11 labels on the wrong expression, and 17 entity boundaries missed. The other significant errors are 29 missed PRO entities, 24 unlabeled BRA entities, and 15 PRO entity boundaries missed.

Feature	P/R/F1 (PRO)	P/R/F1 (BRA)	P/R/F1 (TYP)
Without brand list lookup	73.80 / 63.05 / 68.00	89.95 / 79.05 / 84.15	48.36 / 36.81 / 41.80
With brand list lookup	77.32 / 64.74 / 70.47 (3.52 / 1.69 / 2.47)	91.06 / 85.83 / 88.37 (1.11 / 6.78 / 4.22)	52.14 / 36.31 / 42.81 (3.78 / –0.50 / 1.01)

Table 1: Effect of brand list lookup feature on baseline.

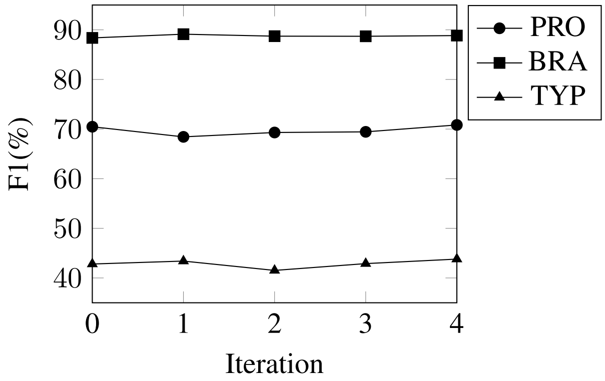

Figure 2: F1 score after each iteration

PRO NE	Precision%	Recall%	F1-score%
PRO	80.25% (2.93)	63.38% (–1.36)	70.83% (0.36)
BRA	91.14% (0.08)	86.65% (0.82)	88.84% (0.47)
TYP	52.05% (–0.09)	37.81% (1.50)	43.80% (0.99)

Table 2: Performance of classifier after semi-supervised.

PRO NE	Precision%	Recall%	F1-score%
PRO	81.05% (3.73)	62.37% (–2.37)	70.49% (0.02)
BRA	91.59% (0.53)	87.26% (1.43)	89.37% (1.00)
TYP	52.77% (0.63)	37.81% (1.50)	44.05% (1.24)

Table 3: Performance of corpus.

References

Andry Luthfi, Bayu Distiawan, and Ruli Manurung. 2014. Building an Indonesian named entity recognizer using Wikipedia and DBPedia. In *Proceesing of 2014 International Conference on Asian Language Processing (IALP).*

Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. 2000. Rule-Based Named Entity Recognition for Greek Financial Texts. In *Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries (COMLEX 2000).*

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*.

Hanchu Zhang, Leonhard Hennig, Christoph Alt, Changjian Hu, Yao Meng, and Chao Wang. 2020. Bootstrapping Named Entity Recognition in E-Commerce with Positive Unlabeled Learning. In *ECNLP-3 @ ACL 2020 : The Third Workshop on e-Commerce and NLP*.

Ika Alfina, Septiviana Savitri, and Mohamad Ivan Fanany. 2017. Modified DBpedia Entities Expansion for Tagging Automatically NER Dataset. In *Proceeding of 9th International Conference on Advanced Computer Science and Information Systems 2017 (ICACSIS 2017).*

Ji-Hwan Kim and P. C. Woodland. 2000. A Rule-based Named Entity Recognition System for Speech Input. In *Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH 2000.*

Jun Zhao and Feifan Liu. 2008. Product Named Entity Recognition in Chinese Text. *Language Resources & Evaluation*, 42(2):197-217.

Wenhui Liao and Sriharsha Veeramachaneni. 2009. A Simple Semi-supervised Algorithm for Named Entity Recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2003. Boosting Precision and Recall of Dictionary-Based Protein Name Recognition. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*.