Clinical Entity Recognition Using Cost-Sensitive Structured Perceptron for NTCIR-10 MedNLP

Shohei Higashiyama
higashiyama@ai.cs.kobe-u.ac.jp

Kazuhiro Seki
seki@cs.kobe-u.ac.jp

Kuniaki Uehara
uehara@kobe-u.ac.jp

Graduate School of System Informatics
Kobe University

ABSTRACT

This paper reports on our approach to the NTCIR-10 MedNLP task, which aims at identifying personal and medical information in Japanese clinical texts. We applied a machine learning (ML) algorithm for sequential labeling, specifically, structured perceptron, and defined a cost function for lowering misclassification cost. On the test set provided by the organizers, our approach achieved an F-score of 77.00 for the de-identification task and 79.14 for the complaint and diagnosis task.

Team Name

KobeU

Subtasks

De-identification (Japanese)
Complaint and diagnosis (Japanese)

Keywords

Clinical natural language processing, named entity recognition, information extraction, structured perceptron, cost-sensitive learning

1. INTRODUCTION

Our research group at the computational intelligence laboratory in Kobe University (KobeU) participated in the NTCIR-10 MedNLP task [5], which targets information extraction (IE) from clinical texts written in Japanese. Specifically, we took part in two subtasks, the de-identification task and the complaint and diagnosis task (C&D task). Both subtasks can be considered as named entity recognition (NER) tasks to identify predefined entities, such as proper names and technical terms, and can be formulated as a sequence labeling problem.

The MedNLP task requires the participants to extract personal and medical information from fictional medical history summary reports. These medical reports are annotated by the MedNLP task organizers and are provided for participants as the sample set.

The purpose of this study is to implement a simple and general ML-based system which is not specific for clinical domains and to evaluate it on the MedNLP dataset as a first step toward more effective clinical text processing systems. To this end, we implemented our system based on structured perceptron [1], which is comparable in performance to modern ML algorithms, such as Support Vector Machines (SVM), despite the simplicity of the algorithm. In addition, we used a cost function in the perceptron framework for achieving higher performance. The cost function is a type of cost-sensitive learning method which lowers the expected cost of misclassification.

2. METHOD

2.1 Task Formulation

The NTCIR-10 MedNLP de-identification task intends to identify personal information about patients, such as ages and personal names, appearing in clinical texts. It is a typical NER task recognizing entities and classifying them into predefined semantic classes.

The de-identification task can be seen as classifying each word in a sentence into one of the labels consisting of a semantic class (e.g., `AGE` and `GENDER`) and a chunk IOB tag, where I, O, and B mean inside, outside, and beginning of an entity, respectively. For example, if a word "64" in "64 years old" is assigned with a label "`B-AGE`", it means that the "64" is recognized as the beginning of an entity with a semantic class "`AGE`" . For word segmentation, we used the Japanese morphological analyzer MeCab [4] (version 0.994).

The complaint and diagnosis task (C&D task) consists of two parts: firstly, patients' symptoms and diagnosis by physicians are extracted and, secondly, their modality are determined (e.g., is a medical condition present or absent in patients?). Although the latter part of the C&D task can be solved as a multiclass classification problem, we formulated the overall C&D task as classifying each word in text into a class defined as a pair of complaint (or diagnosis) and modality (e.g., *complaint and negation*) so as to solve both de-identification and C&D tasks in the same framework. Therefore, we dealt with both subtasks as NER whose semantic classes represent personal information or complaint.

2.2 Structured Perceptron and Cost-Sensitive Learning

For the MedNLP task, we applied structured perceptron [1], which can be used for structured prediction including sequence labeling. Despite its simplicity, structured perceptron is reported to have close performance to SVM which has been successfully applied to various classification problems. In this section, we describe the learning and prediction algorithm on an ordinary and a cost-sensitive structured perceptron.

Let $\mathcal{X}$ be a set of instances and let $\mathcal{Y}_{\boldsymbol{x}}$ be a set of possible label sequences for an instance $\boldsymbol{x} \in \mathcal{X}$, where $\boldsymbol{x}$ denotes a word sequence (i.e., sentence) in the training or test data. Also, $\boldsymbol{y} \in \mathcal{Y}_{\boldsymbol{x}}$ denotes a possible label sequence of $\boldsymbol{x}$. Note that $\mathcal{Y}_{\boldsymbol{x}}$ is equivalent to the direct product $\mathcal{L}^n$, where n is the length of $\boldsymbol{x}$ and $\mathcal{L}$ is a set of labels, such as `B-AGE` and `O`.

The learning on structured perceptron can be considered as finding the weight vector $\boldsymbol{w} \in \mathbb{R}^d$ so that the discriminative function f predicts the correct label sequences of instances. The discriminative function $f : \mathcal{X} \to \mathcal{Y}$ is defined as

$$f(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}, \boldsymbol{y}) \rangle \ ,$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product of two arguments and $\Phi(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$ is the feature vector of $\boldsymbol{x}$ and $\boldsymbol{y}$.

The prediction $\hat{\boldsymbol{y}}$ for $\boldsymbol{x}$ is the output of f as in

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathcal{Y}}{\operatorname{argmax}} \ f(\boldsymbol{x}, \boldsymbol{y}) \ . \tag{1}$$

During learning on the training data, we receive a training instance $\boldsymbol{x}_t$ on each round t, and output its prediction $\hat{\boldsymbol{y}}_t$ by Eq. (1). Then, $\boldsymbol{w}$ is updated by Eq. (2) if the prediction $\hat{\boldsymbol{y}}_t$ is different from the correct label sequence $\boldsymbol{y}_t$:

$$\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t + \Phi(\boldsymbol{x}_t, \boldsymbol{y}_t) - \Phi(\boldsymbol{x}_t, \hat{\boldsymbol{y}}_t) \ , \tag{2}$$

where $\boldsymbol{w}^t$ is the weight vector on round t. The learning is iterated through all the training instances T times. Label sequences of test instances can be predicted by Eq. (1) in the same manner as training instances.

In addition to using structured perceptron, we exploited information on distance between a correct and a candidate label sequence of each training instance during learning based on cost-sensitive learning of an ML framework for lowering misclassification cost. Cost-sensitive approaches were, for example, applied to semantic role labeling and exploited on the study by Johansson et al. [3] which used passive-aggressive [2].

The cost-sensitive learning algorithm on structured perceptron updates the weight vector $\boldsymbol{w}$ by using $\tilde{\boldsymbol{y}}_t$ defined below instead of $\hat{\boldsymbol{y}}_t$ in Eq. (2).

$$\tilde{\boldsymbol{y}}_t = \underset{\boldsymbol{y} \in \mathcal{Y}}{\operatorname{argmax}} \ f(\boldsymbol{x}_t, \boldsymbol{y}) + \alpha \rho(\boldsymbol{y}_t, \boldsymbol{y}) \tag{3}$$

In Eq. (3), $\rho : \mathcal{Y} \times \mathcal{Y} \to \mathbb{N} \cup \{0\}$ is the cost function which returns a larger value for larger distance between $\boldsymbol{y}_t$ and $\boldsymbol{y}$, and α is a parameter taken to have a positive real number. Here, we define the cost function ρ by

$$\rho(\boldsymbol{y}_1, \boldsymbol{y}_2) = \sum_{i=1}^{|\boldsymbol{y}_1|} \delta(y_1^{(i)}, y_2^{(i)}) \ ,$$

where $|\boldsymbol{y}|$ is the length of the vector $\boldsymbol{y}$ and the function $\delta : \mathcal{L} \to \{0, 1\}$ is defined as follows:

$$\delta(y_1, y_2) = \begin{cases} 0 & (y_1 = y_2) \\ 1 & (y_1 \neq y_2) \end{cases} \ .$$

In the cost-sensitive learning framework, the weight vector can be updated so as to reserve margin $\alpha\rho(\boldsymbol{y}_t, \tilde{\boldsymbol{y}}_t)$ by using $\tilde{\boldsymbol{y}}_t$ instead of $\hat{\boldsymbol{y}}_t$, that is,

$$\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t + \Phi(\boldsymbol{x}_t, \boldsymbol{y}_t) - \Phi(\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t) \ .$$

2.3 Features

We used the following features in the experiments:

- tokens in the window of size two around the current token and
- the part-of-speech (POS) tag, the subtype of POS tag, the lemma and the furigana of the current token.

As the latter features, we used the output of MeCab for each sentence.

3. EVALUATION

3.1 Parameter Setting

We determined the optimal value of parameter α in Eq. (3) and the number of iterations T using the sample set as follows.

1. We used the 90% of the sample set as the learning set and the remaining 10% as the validation set.
2. Varying the value of α and increasing the value of T, we learned a model for particular α and T on the learning set and evaluated it on the validation set.
3. The values of α and T which resulted in the best F-score were considered as the optimal.

As a result, the optimal α and the number of iterations T were set to 30 and 20, respectively. By using the cost function, both precision and recall on the validation set improved by around 4 points, compared with the method without the function. We used these values for producing our official runs on the test set submitted to the MedNLP organizers.

3.2 Evaluation Using the Test Set

Table 1 shows the performance of our system using the test set. Table 1 (a) shows the overall performance and Table 1 (b) shows the performance of each entity class. The performance was measured by precision, recall, F-measure ($\beta = 1$), and accuracy.

Recall was always lower than precision for all classes of both tasks, and especially lower in the family and the suspicion classes, which led to degraded F-scores. In addition, the lower performance for the total on the C&D task than 2-way indicate difficulty of modality classification.

3.3 Discussion

We evaluated our system using the sample set for error analysis. We used a 5-fold cross-validation method in the evaluation, then analyzed the results on the validation set for each fold.

As compared with the performance on the test set, the performance on the validation sets was worse by several points for the C&D task, and almost equivalent for the de-identification task. The reason of the former is the fewer training instances, and that of the latter was that the target entities for the de-identification task have much in common as we discuss shortly.

3.3.1 Analysis on de-identification task

Despite the smaller number of positive instances of entity classes for the de-identification task than that for the C&D task, the performance for the former classes was relatively

Table 1: Results of both de-identification (De-ID) task and compliant and diagnosis (C&D) task. The "2-way" is a result of recognition of compliant/diagnosis or not. The "total" is a result including classification of modality classes. P, R, F and A indicate precision, recall, F-measure ($\beta = 1$), and accuracy, respectively.

(a) Overall performance.

subtask	P	R	F	A
De-ID	82.09	76.39	79.14	99.38
C&D (2-way)	82.37	72.29	77.00	95.48
C&D (total)	74.72	65.58	69.86	94.50

(b) Performance of each entity class.

subtask	tag	P	R	F
De-ID	a (age)	80.65	78.12	79.37
	h (hospital)	72.73	63.16	67.61
	t (date time)	84.56	81.56	83.03
	x (sex)	100.00	50.00	66.67
C&D	c-positive	72.87	67.04	69.83
	c-family	66.67	36.36	47.06
	c-negation	82.35	68.02	74.50
	c-suspicion	55.00	36.67	44.00

high on the whole. The reason is that a large portion of these entities fit typical patterns. For example, over 70 percents of the instances of the age class in the sample set match a simple regular expression,

[1-9]?[0-9] 歳 [時頃 (ごろ)]?[ー〜(から)(より)(まで)]? .

For misclassified cases, we found two major types of errors across all classes in this task: (1) recognition of incorrect boundaries of entities; and (2) undetection of entities (false negatives).

Specifically, the most frequent errors on the age class was found to be the first type, such as "４７歳" for a correct boundary "２７歳〜４７歳" (*27 to* 47 years old) and "１０代" for "１０代前半" (*early* 10s). Because words or expressions co-occurring with or including ages themselves as numerical values are limited, it may be effective to fix system outputs by rule-based post-processing.

On the other hand, most errors on the hospital class was the second type (i.e., false negatives), such as "同院" (the hospital) and "総合病院" (general hospital). The reason is that these words rarely appeared in the sample set in contrast to frequently appearing words, such as "当院" (our hospital) and "近医" (local hospital), which were correctly detected.

As for the time class, both types of errors were often observed. A large portion of boundary errors were recognizing narrower scopes for entities than their correct ones, e.g., "１０月２９日" against a correct boundary "１０月２９日夕刻まで" (*until the evening on* Oct. 29). Many false negatives were found to be expressions using slashes, such as "７／２０". More formal expressions, such as "７月２０日", are more often used in the sample set.

For dealing with the errors of the hospital and the former type of the time, constructing and using dictionaries composed of expressions which often constitute or co-occur with those type of entities may be beneficial. For the latter type of the time, rule-based post-processing may be effective, similarly to the age class.

Input:	薬剤性肺炎 の可能性を考え
	⇓
Output:	薬剤/性/肺炎/の/可能/性/を/考え

Figure 1: An example of a parsed sentence including a suspicion entity by MeCab. The underlined part in the input sentence indicates an entity annotated with the suspicion class. The parts segmented by slashes in the output indicate words segmented by the parser.

3.3.2 *Analysis on compliant and diagnosis task*

In addition to the two types of errors discussed for the previous task, there were mainly two types of errors in detecting compliant entities: (3) misclassification of the modality classes; and (4) misdetection of non-entities (false positive).

The most frequent errors were undetection of entities through all classes, and this type of errors frequently observed in the positive and the negation classes. In order to reduce such false negatives and improve recall, we plan to use external knowledge resources such as public dictionaries in future work.

The second most frequent errors were misclassification of entities whose boundaries were correctly recognized. They accounted for a major portion of errors on the three classes except the positive class. Especially, the low performance on the family and the suspicion classes was due to misclassification in addition to undetection which occur similarly as the other modality classes.

For these modality classes, it was found that there exist typical keywords which often co-occur with entities. Entities of the family class co-occur with family relation names. In particular, most of them in the sample set co-occur in itemized sentences, such as "父：心筋梗塞" (Father: cardiac infarction). Entities of the negative class and the suspicion class occur ahead of expressions of negations, such as "なし" (be absent), and expressions of uncertainty, such as "考えられる" (be concerned), "疑いがある" (be suspected), and "可能性がある" (could be).

However, our system could not exploit these keywords because of the limited window size of two around the current token, and entities often occur at a distance from keywords, especially in the suspicion class. For example, Figure 1 shows an input sentence containing a suspicion entity "薬剤性肺炎" and its parsed output by the MeCab morphological analyzer. Two out of three tokens constituting the entity (i.e., "薬剤" and "性") are more than two tokens away from the uncertainty keywords (i.e., "可能", "性" and "考え").

To improve classification performance for modality classes, specifically recall, it is crucial to increase the window size to, for example, sentence boundaries. Alternatively, it may be effective to take advantage of dependency parsing.

The other causes of the observed errors were incorrect boundary errors and misdetection errors. The reasons require a further study.

4. CONCLUSIONS

This paper described our system to extract personal and medical information from clinical texts. We implemented a simple system based-on structured perceptron as a first step toward more effective Japanese clinical text processing systems, and analyzed its performance and issues for achieving the goal. Although the result on the MedNLP dataset indicates that classification of clinical entities into their modality classes is difficult, our analysis revealed that the terms and expressions in clinical texts have useful patterns and characteristics which could be exploited for more accurate extraction.

For developing more advanced systems, we plan to use knowledge resources, such as a lexicon in the clinical domain, and to explore more useful features embedded in machine learning approaches.

5. REFERENCES

[1] M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on EMNLP*, pages 1–8, 2002.

[2] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *The Journal of machine learning research*, 7:551–585, 2006.

[3] R. Johansson and P. Nugues. Dependency-based semantic role labeling of propbank. In *Proceedings of the 2008 conference on EMNLP*, pages 69–78, 2008.

[4] T. Kudo, K. Yamamoto, and Y. Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 conference on EMNLP*, pages 230–237, 2004.

[5] M. Morita, Y. Kano, T. Ohkuma, M. Miyabe, and E. Aramaki. Overview of the NTCIR-10 mednlp task. In *Proceedings of NTCIR-10*, pages 1–2, 2013.