Similarity Matrix Model for the NTCIR-12 MedNLPDoc Task

Yuichiro Sawai
Nara Institute of Science and Technology, Japan
sawai.yuichiro.sn0@is.naist.jp

Mai Omura
Nara Institute of Science and Technology, Japan
omura.mai.oz5@is.naist.jp

Hiroki Ouchi
Nara Institute of Science and Technology, Japan
ouchi.hiroki.nt6@is.naist.jp

Yuki Nagai
Nara Institute of Science and Technology, Japan
nagai.yuki.nr9@is.naist.jp

Masashi Yoshikawa
Nara Institute of Science and Technology, Japan
yoshikawa.masashi.yh8@is.naist.jp

Ikuya Yamada
Studio Ousia, Japan
ikuya@ousia.jp

ABSTRACT

We participated in the NTCIR-12 MedNLPDoc phenotyping task. In this paper, we describe our approach for this task. The core part of our model is a similarity matrix model in which each element has a local similarity value between n-grams from a disease name and a medical record. We conduct an experiment to evaluate the effectiveness of our method. We report the results of our preliminary experiments and the run submission.

Team Name

matsu

Subtasks

Phenotyping task (Japanese)

Keywords

similarity matrix, neural networks, word representations

1. INTRODUCTION

In this paper, we describe the approach we took for the NTCIR-12 MedNLPDoc phenotyping task. In this task, each participant team is asked to develop a system that assigns diagnostic codes (ICD-10) for given medical records. The details of the task is described in the task overview paper [1].

We tackle this task by taking an approach of scoring a pair of a candidate disease name and a medical record. In our approach, we create a similarity matrix in which each element has local a similarity between n-grams from a disease name and a medical record.

We conduct preliminary experiments to evaluate the effectiveness of our model using the provided training data. We also report our results of the run submission evaluated on the provided test data. We confirm that our model performs better than simple n-gram matching baselines.

2. OUR APPROACH

Each participant in this task is asked to identify the diagnoses of the given medical record in terms of ICD-10 codes. ICD-10 codes are embodied as associated disease names in medical records. Thus, one of the essential components for tackling this task is to match disease names with medical records.

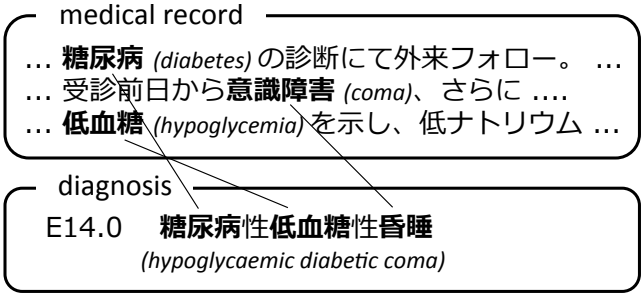

Figure 1: An example of multiple spans contributing to a single ICD-10 code.

One straight-forward approach for this task would be to create a system that identifies spans of words in medical records that could possibly contribute to diagnoses, and create another system that assigns ICD-10 codes for each span to determine the set of diagnoses.

However, in this task, we do not have such a dataset in which spans are annotated and aligned with diagnoses. Instead, we are provided with a dataset in which medical records are paired with sets of diagnoses without alignment information. On top of that, it is often hard to determine a single span that contributes to a certain ICD-10 code, but one ICD-10 code can be attributed to multiple spans in the medical record. For example, in Figure 1, the diagnosis "糖尿病性低血糖性昏睡 (hypoglycaemic diabetic coma)" is derived from three spans ("糖尿病 (diabetes)", "意識障害 (coma)", and "低血糖 (hypoglycemia)") in the medical record. For these reasons, in this paper, we seek an end-to-end approach that directly matches disease names with the entire medical record by a string similarity measure.

A naive choice of string similarity measure would be a simple n-gram matching score:

$$\frac{|N^{dis} \cap N^{rec}|}{|N^{dis}|}, \tag{1}$$

where N^{dis} is a set of word n-grams in the disease name, and N^{rec} is a set of word n-grams in the medical record.

One problem of simple n-gram matching is that each word is treated equally, but in reality, there are certain words (e.g., parts of body) that should be weighted more than other words. For example, "急性 腎 不全 (acute kidney failure)" has

a unigram match score of 0.67 with "急性 肝 不全 (acute liver failure)" if we regard both as being made up of three words. However, these two diseases are quite dissimilar because they are about different parts of body. In fact, the ICD-10 code of the former is "N17.9", while the ICD-10 code of the latter is "K72.0". Thus, it is desirable that the failure of matching "腎 (kidney)" be punished harsher than other words.

Another problem of n-gram matching is that it cannot take paraphrases into consideration. For example, the disease name "卵巣 腫瘍 (ovarian tumor)" has the same unigram match score of 0.5 with "卵巣 のう 腫 (ovarian cyst)" and "卵巣 捻転 (ovarian torsion)". However, the former should have a higher score because "のう 腫 (cyst)" can be paraphrased as "腫瘍 (tumor)".

In this paper, we seek a string similarity measure in which the weight of each word can be learned from a training data and paraphrases can be matched.

2.1 Similarity Matrix Model

In this section, we descirbe our string similarity measure based on a similarity matrix. An overview of our model is shown in Figure 2.

The input to our model is a pair of a candidate disease name and a medical record. The output is the score of how likely the candidate disease name is as the diagnosis for the medical record. Our model creates a similarity matrix where each element has the local similarity value between two n-grams from the disease name and the medical record. Then, the overall score is calculated by combining values in the similarity matrix. When predicting the set of diagnoses for the given medical record, we calculate scores for all possible disease names using this model, and output all the ICD-10 codes that correspond to the disease names with scores higher than a threshold. The threshold is determined so that it achieves the best performance on the development set. In the rest of this section, we formulate the details of our model.

The input to the model is a pair of a candidate disease name S^{dis} and a medical record S^{rec}, where both S^{dis} and S^{rec} are sequences of words. We denote the length of S^{dis} as $|S^{dis}|$ and the length of S^{rec} as $|S^{rec}|$.

The first step of our model is the wide one-dimensional convolution [2] for modeling n-grams in S^{dis} and S^{rec}. Firstly, d-dimensional word embeddings are looked up for each word in S^{dis} and an embedding matrix $\mathbf{E}^{dis} \in \mathbb{R}^{d \times |S^{dis}|}$ is constructed in which i-th column $\mathbf{E}^{dis}_i$ is the word embedding of the i-th word of S^{dis}. Then, convolution over $\mathbf{E}^{dis}$ is performed with window size n^{conv} to obtain an n-gram matrix $\mathbf{C}^{dis} \in \mathbb{R}^{d \times |S^{dis}|}$, in which i-th column $\mathbf{C}^{dis}_i$ is calculated by

$$\mathbf{C}^{dis}_i = \mathbf{W}^c[\mathbf{E}^{dis}_{i-\lfloor \frac{n^{conv}}{2} \rfloor}; ...; \mathbf{E}^{dis}_{i+\lfloor \frac{n^{conv}}{2} \rfloor}], \quad (2)$$

where $\mathbf{W}^c \in \mathbb{R}^{d \times {n^{conv}}d}$ is the convolution weights and [...] denotes vector concatenation. The same procedure is taken for S^{rec}, resulting in matrices $\mathbf{E}^{rec}$ and $\mathbf{C}^{rec}$. The same set of word embeddings and the same convolution weights $\mathbf{W}^c$ are used for both disease names and the medical records.

After n-gram matrices $\mathbf{C}^{dis}$ and $\mathbf{C}^{rec}$ have been calculated, a similarity matrix $\mathbf{M} \in \mathbb{R}^{|S^{dis}| \times |S^{rec}|}$ is constructed by calculating negative squared Euclidean distances for all pairs of n-gram embeddings from $\mathbf{C}^{dis}$ and $\mathbf{C}^{rec}$. Namely, the element in row i and column j of $\mathbf{M}$ is

$$\mathbf{M}_{ij} = -\|\mathbf{C}^{dis}_i - \mathbf{C}^{rec}_j\|^2. \quad (3)$$

$\mathbf{M}_{ij}$ is equal to 0 if n-grams around i-th word in S^{dis} and j-th word in S^{rec} are identical, and less than 0 otherwise.

Then, maximum values of each row of $\mathbf{M}$ are taken, so as to construct a vector $\mathbf{m} \in \mathbb{R}^{|S^{dis}|}$ in which i-th element is

$$\mathbf{m}_i = \max_{0 \le j \le |S^{rec}|} \mathbf{M}_{ij}. \quad (4)$$

$\mathbf{m}_i$ has the real-valued score of how well the n-gram around i-th word in S^{dis} matches S^{rec}.

In order to take the weight of each word in S^{dis} into consideration, a weight vector $\mathbf{w} \in \mathbb{R}^{|S^{dis}|}$ is created. i-th element of $\mathbf{w}$ is calculated by logistic regression with $\mathbf{C}^{dis}_i$ (embedding of the n-gram around i-th word in S^{dis}) as input,

$$\mathbf{w}_i = \sigma((\mathbf{w}^w)^T \mathbf{C}^{dis}_i + b^w), \quad (5)$$

where $\sigma(x) = 1/(1+e^{-x})$ and $\mathbf{w}^w \in \mathbb{R}^d$ and b^w are the weights and the bias of logistic regression.

The overall score s^{mat} by the similarity matrix is defined as the average of $\mathbf{m}$ (Equation 4) weighted by $\mathbf{w}$ (Equation 5),

$$s^{mat} = \frac{\mathbf{w}^T \mathbf{m}}{|S^{dis}|}. \quad (6)$$

Finally, this score is combined with additional real-valued features $\mathbf{f} \in \mathbb{R}^{d_f}$, and the probability p of S^{dis} being one of the diagnoses of S^{rec} is calculated by logistic regression,

$$p = \sigma((\mathbf{w}^l)^T[s^{mat}; \mathbf{f}] + b^l), \quad (7)$$

where $\mathbf{w}^l \in \mathbb{R}^{d_f+1}$ and b^l are the weights and the bias of logistic regression. This probability is used as the final score assigned by the model.

All the parameters (word embeddings, $\mathbf{W}^c$, $\mathbf{w}^w$, b^w, $\mathbf{w}^l$, b^l) are optimized so as to minimize the cross-entropy loss over the entire training data:

$$\mathcal{L} = -\sum_{k=1}^{N} \left(t_k \log p_k + (1-t_k)\log(1-p_k)\right), \quad (8)$$

where N is the size of the training data (pairs of S^{dis} and S^{rec}), p_k is the probability calculated by Equation 7 for k-th sample in the training data, and t_k is a binary label of whether the diagnosis corresponding to S^{dis} in k-th sample is included in the set of the diagnoses of S^{rec} in k-th sample. Gradiens are calculated by backpropagation of errors, and a gradient-based method can be employed to optimize the parameters.

Our similarity matrix model is similar to [6], where they use multiple similarity matrices with varied granularity for paraphrase identification. They use exponentiated Euclidean distance (instead of Equation 3 as similarity measure. They also use dynamic k-max pooling [2] instead of max pooling (Equation 4) and averaged sum (Equation 6) in order to obtain fixed-length vector representation.

3. EXPERIMENTAL RESULTS

3.1 Settings of Preliminary Experiments

Each task participant is provided with a training data which contains 200 pairs of medical records and sets of diagnoses (in terms of ICD-10 codes). The samples in the training data were extracted from the ICD training book [5]. We found out that some of the medical records in this book

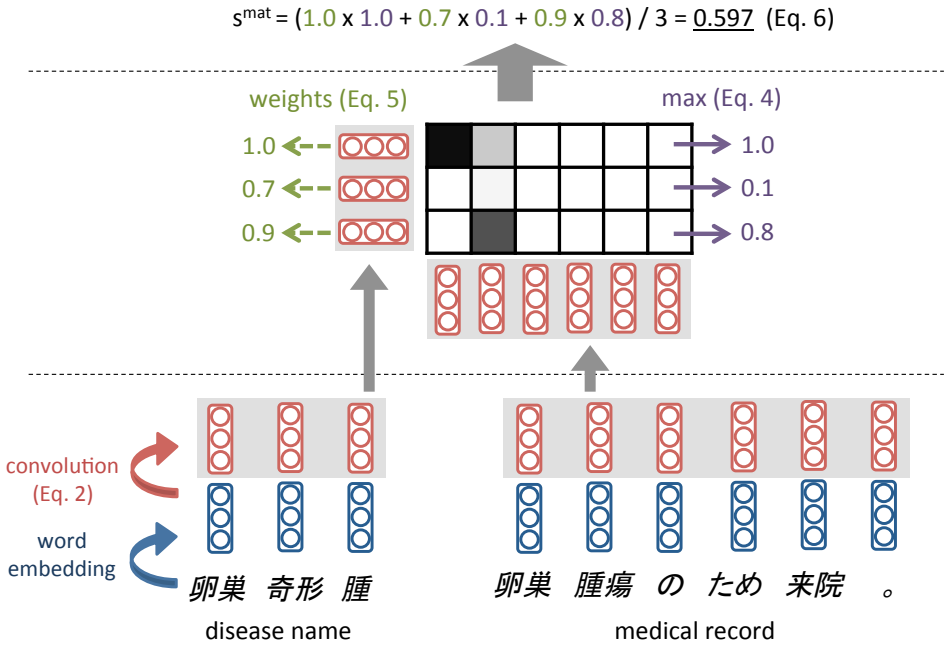

Figure 2: An overview of our similarity matrix model: similarity between a disease name "卵巣 奇形 腫" and a medical record "卵巣 腫瘍 の ため 来院 。" is calculated.

	Train	Dev
# of medical records	203	50
# of ICD-10 codes (diagnoses)	791	209

Table 1: The corpus statistics of the dataset used

were not present in the provided training data. In order to augment the training data, we processed the book with OCR, and added the missing medical records to the original training data. We split the resulting data into training set and development set. The corpus statistics of the resulting dataset is shown in Figure 1.

Candidate ICD-10 codes are extracted from the ICD-10 MEDIS Standard Master[1]. We only use ICD-10 codes associated with at least one disease name, resulting in 7,712 ICD-10 codes and 26,205 disease names. We segment both of the medical records and the disease names into morphemes using MeCab[2].

In order to train our model, we need positive samples and negative samples, which are pairs of disease names and medical records. However, in the provided training data, the medical records are paired with sets of ICD-10 codes. In order to create positive samples, we need to convert ICD-10 codes to associated disease names. Since, one ICD-10 code usually has multiple disease names, we need to choose one of the disease names. Fortunately, in the ICD training book, each diagnostic ICD-10 code is associated with a disease name. When we processed the training book with OCR, we also extracted these disease names. Then, for each diagnostic ICD-10 code, we select the disease name most similar (in terms of edit distance) to the extracted disease name from the ICD-10 MEDIS Standard Master. We chose top-100 disease names with highest model scores as negative samples for each medical record.

We use word 2-gram (Equation 1) as an additional feature to combine with the score of the similarity matrix (in Equation 7).

Word embeddings are initialized with 128-dimensional vectors trained with word2vec [4] tool on medical articles extracted from the Japanese Wikipedia. The bias b^w (in Equation 5) and the feature weights ($\mathbf{w}^l$ in Equation 7) are initialized with 1.

All the parameters are optimized by Adam [3] (with the hyperparameters described in the original paper). Each mini-batch consists of one medical record and all of its positive and negative disease names. We train the model for 50 epochs, and select the epoch at which the model performs best on the development set.

Evaluation metrics are precision, recall, F-measure of predicted ICD-10 codes compared with the correct sets of ICD-10 codes for each medical record. We vary the threshold of the score (from 0 to 1) and report the highest F-measure achieved on the development data.

3.2 Results of Preliminary Experiments

The baselines are methods using n-gram matching scores (Equation 1) as similarity measure to rank disease names. We vary the n-gram size from 1 to 4. As our method, the disease names with scores higher than the threshold (determined on the development set) are used as diagnoses. The results of the baselines are shown in Table 2. The best performance is achieved when we set n to 2.

The results of our method are shown in Table 3. We vary the window size n^{conv} from 1 to 5, and switch the use of word weights and the additional 2-gram feature.

[1] http://www2.medis.or.jp/stdcd/byomei/
[2] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

n	F-measure
1	0.152
2	**0.190**
3	0.165
4	0.170

Table 2: Performance of the baselines

n^{conv}	weights	2-gram feature	
		yes	no
1	yes	0.234	0.184
	no	0.241	0.187
3	yes	0.284	0.286
	no	0.285	0.293
5	yes	**0.294**	0.251
	no	0.259	0.240

Table 3: Performance (F-measures) of our method.

The performance of our method is generally better than the simple n-gram matching baselines. The best performance is achieved when we set n^{conv} to 5 and use word weights and the 2-gram feature. When we set n^{conv} to 1, the performance is worse because the similarity matrix only captures word-level co-occurrence.

Combining the 2-gram feature helps the most when n^{conv} is 1. When n^{conv} is greater than 1, the 2-gram feature has less impact, because the similarity matrix already captures n-grams co-occurence, making the 2-gram feature redundant. When we set n^{conv} to 1 and do not use the 2-gram feature, our method is still better than the 1-gram baseline. We believe this is due to the ability of the similarity matrix to do fuzzy matching of words.

The use of word weights mostly does not help to improve generalization ability. We believe this is because we do not have enough data to exploit the model's ability to assign different weights to different words.

3.3 Results of Run Submission

As the official results for this task, we submitted the prediction by the model that achieved the best F-measure at the time of submission. The hyperparameters of the selected model was $n^{neg} = 10$, $n^{conv} = 3$, and we did not use the 2-gram feature or the word weights. We also did not use the word embeddings pre-trained with word2vec. The F-measure on our development data was 0.245.

The results are shown in Table 4. The evaluation metrics are explained in the task overview [1].

We observe that the test data for run submission is quite different from the training data. In the test data, contextual information of spans plays an important role. For example, negation ("ない") is prevalent in the test data. Even if a disease name strongly matches a medical record, it cannot be a final diagnosis when it is negated from its context. There are also cases where it is clear that a disease name in a medical record is not about the patient,but about the family. In such a case, the disease name should not be coded as a final diagnosis, since it is not the patient's own disease.

Our method cannot consider contextual information such as negation and family history, because our model does not explicitly model spans and their contexts in medical records.

	Precision	Recall	F-measure
SURE	0.237	0.223	0.230
MAJOR	0.313	0.168	0.219
POSSIBLE	0.374	0.109	0.169

Table 4: Results of run submission

Thus, our model scores worse on the test data for this task.

4. CONCLUSION

We participated in the NTCIR-12 MedNLPDoc phenotyping task. To tackle the task, we proposed a method that scores a pair of a candidate disease name and a medical record. The core part of our method is a similarity matrix in which each element has local similarity between n-grams from the disease name and the medical record. We conducted an experiment with the provided dataset and confirmed that our method performed better than the n-gram baseline. One possible future direction is extending our method to incorporate contextual information.

5. ADDITIONAL AUTHORS

Additional authors: Hiroyuki Shindo (Nara Institute of Science and Technology, Japan, email: `shindo@is.naist.jp`)

6. REFERENCES

[1] E. Aramaki, M. Morita, Y. Kano, and T. Ohkuma. Overview of the ntcir-12 mednlpdoc task. In *Proceedings of the 12th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, 2016.

[2] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, 2014.

[3] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*, 2015.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, 2013.

[5] K. Toba and Shindan Joho Kanri Tokyo Network. *ICD Coding Training (2nd Edition; in Japanese)*. Igaku Shoin, 2006.

[6] W. Yin and H. Schütze. Convolutional neural network for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 901–911, 2015.