Normalization of Abbreviations/Acronyms: THCIB at CLEF eHealth 2013 Task 2

Yunqing Xia[1], Xiaoshi Zhong[1]
Peng Liu[2], Cheng Tan[2], Sen Na[2], Qinan Hu[2], Yaohai Huang[2]

[1] Dept. of Comp. Sci. & Tech., Tsinghua National Laboratary of Information Science and Technologies, Tsinghua University, Beijing 100084, China
{yqxia, xszhong}@tsinghua.edu.cn

[2] Canon Information Technology (Beijing) Co. Ltd., Beijing 100080, China
{liupeng, tancheng, nasen, huqinan, huangyaohai}@canon-ib.com.cn

Abstract. This paper describes the THCIB systems that used in the ShARe/CLEF eHealth Lab 2013 task 2. We built a baseline system using open source software, and improve the performance by adding dictionaries. The dictionary is built from training set and web resource using the existing technologies. The experimental results show that adding dictionary of acronym/abbreviation can improve the performance significantly.

Keywords: acronym normalization, abbreviation normalization, dictionary generation, clinical report processing, natural language processing

1 Introduction

The ShARe/CLEF eHealth Lab 2013 task 2 aims to normalize of acronyms/abbreviations (AAs) [1].This task maps the acronyms and abbreviations to UMLS (Unified Medical Language System) CUIs (Concept Unique Identifier), which provide an expansion and a definition of the term [2]. The input is a sentence with annotations of the AA, and the output is the CUIs. For example, the input sentence is “BP 142/70.” and “BP” is annotated as an acronym which means “blood pressure”. Then the target is mapping “BP” to CUI “C0005823”.

In this paper we describe the baseline system and the dictionaries we used to improve the performance. And we also describe the experimental results on the training set and the test set.

The reminder of this paper is structured as follows. In section 2, we present an overview of our baseline system. In section3, we describe how to build the dictionaries. The experiments and analysis of the results are described in section 4. We give the conclusion in section 5.

2 Baseline System

The baseline system for task 2 is implemented using the cTAKES [3]. The cTAKES (Apache clinical Text Analysis and Knowledge Extraction System) is an open source natural language processing system for information extraction from electronic medical record clinical free-text. It can process the clinical text and identify the clinical named entities from various resources including the UMLS [4].

The flowchart of baseline system is shown in Fig. 1.

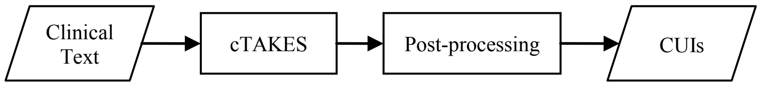

Fig. 1. Flowchart of baseline system.

In the baseline system, the clinical text is processed in following steps: 1) the clinical text is sent to cTAKES; 2) the cTAKES processes the clinical text and maps all concepts in the clinical text to UMLS. The concepts will be saved in an XCAS file. 3) Post-processing the XCAS file, and extract the CUI of each AA which has been annotated. If an AA has no corresponding CUI, a "CUI-less" tag will be given; 4) output the CUIs.

3 Acronym/Abbreviation Dictionaries

The baseline system has very low performance. Experiment on the training set shows that about 80% AAs can't be recognized correctly. An intuitive method to improve the AA recognition performance is adding dictionary as external resource [5]. We build two kinds of dictionaries for task 2.

3.1 CUI Dictionary

The CUI dictionary is a dictionary which maps the AA to the corresponding CUI directly. The CUI dictionary is built using the training set. There are 3660 AAs in the training set. After combination, the CUI dictionary contains 668 entries [5]. An example of CUI dictionary entry is "BP" → "C0005823".

3.2 Full Name Dictionary

The full name dictionary maps the AA to its full name. This dictionary is built using two resources. One is the training set, we extract 668 entries. The other is a web medical dictionary which contains 2180 entries [6]. After the combination, the full name dictionary has 2725 entries. An example of full name dictionary entry is "BP" → "blood pressure".

One AA may map to more than one full name. For example, the AA “HA” may map to “headache” or “herpangina” in different context. For this case, we rank the full name using the co-occurrence of the AA and its full name in a large scale web page corpus which contains more than 1.6 million web pages [7,8]. The full name which has the most co-occurrence with the AA will be reserved.

3.3 Baseline System with AA Dictionaries: The Submitted Run

As the baseline system, the input is a sentence with annotation of AA, and the output is the corresponding CUI. We use a dictionary lookup method [4]. The system works as follows:

1) Look up the AA in the CUI dictionary. If find the AA in the dictionary, output the corresponding CUI;

2) If can’t find the AA in the CUI dictionary, look up the AA in the full name dictionary;

3) If find the full name of the AA, send the full name to the cTAKES; else, send the whole sentence to the cTAKES;

4) Extract the CUI of the AA. If the AA has no CUI, give it a “CUI-less” tag.

4 Experimental Results

4.1 Dataset

The training set contains 200 clinical reports, and totally 3660 AAs. We used all of the training set to build the dictionary and evaluate the performance of the system. The test set contains 100 clinical reports. We will give the evaluation results on training set and test set.

4.2 Evaluation Metrics

Precision is used in this evaluation. Two conditions are setup. One is strict, which means that the recognized words are perfectly matched; the other is relaxed, which means that the recognized words have overlap with the gold standard.

4.3 Internal Results

Results of Baseline System
We use all the training set to evaluate the baseline system. The evaluation results are shown in Table 1.

Table 1. Evaluation results of baseline system.

System	*Accuracy (Strict)*
baseline	0.208

From Table 1, we can find the baseline system has very low performance on the AA normalization.

Results of Baseline System with Dictionaries

Because the dictionary is built using the training set, we use a five-fold cross validation to verify the CUI dictionary. The results are shown in Table 2.

Table 2. 5-fold cross validation using the CUI dictionary.

System	*Accuracy (Strict)*
baseline + CUI dictionary (1/5)	0.739
baseline + CUI dictionary (2/5)	0.711
baseline + CUI dictionary (3/5)	0.800
baseline + CUI dictionary (4/5)	0.703
baseline + CUI dictionary (5/5)	0.739
Average	0.738

After adding CUI dictionary, the average accuracy is 73.8% which is 53% higher than the baseline system. The reason is that the CUI dictionary is built using the training set which has precisely mapping from the AA to the CUI.

We also have a closed test using the whole training set. The results are shown in Table 3.

Table 3. Closed test on training set.

System	*Accuracy (Strict)*
baseline + full name dictionary	0.409
baseline + CUI dictionary	0.885
baseline + CUI dictionary + full name dictionary	0.880

From Table 3, we can find that only adding full name dictionary can improve the accuracy from 0.208 to 0.409. And only adding CUI dictionary can improve the accuracy to 0.885. The accuracy of adding two dictionaries is slightly lower than only using the CUI dictionary. This is caused by the full name dictionary because one full name may map to several CUIs. And current system doesn't consider the disambiguation problem.

4.4 Official Results

Though using full name dictionary will reduce the accuracy slightly, the system will be more robust because it can cover more AAs. So we select this system to process the test set. The official results are shown in Table 4.

Table 4. Official evaluation results on test set.

System	*Accuracy (Strict)*	*Accuracy (Relaxed)*
baseline + CUI dictionary + full name dictionary	0.657	0.685

From Table 4, we can find that the performance on the test set is similar to the performance on the training set. This means that the acronym/abbreviation dictionaries work well on the test set.

5 Conclusion

For the time limitation, our purpose is using the existing technologies to build the baseline system for acronym/abbreviation normalization and verify the performance of the existing technologies. We built a baseline system using OSS for ShARe/CLEF eHealth task 2. In order to improve the performance, we built two kinds of dictionaries as the external resource. One is the CUI dictionary which maps the AA to CUI directly; the other is the full name dictionary which maps the AA to its full name. The CUI dictionary is built using the training set, and the full name dictionary is built using the training set and the web resource. The experimental results show that adding dictionary to the baseline system can improve the performance significantly.

Acknowledgement

This research is supported by Canon Inc. (No. QIM2013). The Shared Annotated Resources (ShARe) project is funded by the United States National Institutes of Health with grant number R01GM090187. We also appreciate the valuable comments from the task organizer.

References

1. Suominen, Hanna, Sanna Salanter, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Danielle Mowery, Johannes Leveling, Lorraine Goeuriot, Liadh Kelly, David Martinez and Guido Zuccon. Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. Proceedings of CLEF 2013. Lecture Notes in Computer Science (LNCS), Springer.

2. Unified Medical Language System (UMLS), http://www.nlm.nih.gov/research/umls/
3. Apache cTAKES, http://ctakes.apache.org/index.html
4. Savova, G., Masanz, J.J., Ogren, P., Zheng, J., Sohn, S., Kipper-Schuler, K., Chute, C.: Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component, evaluation and applications. J Am Med Inform Assoc. 17, 507-513(2010)
5. Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: NYU: Description of the MENE Named Entity System as Used in MUC-7. In proc. of MUC 7, 1998.
6. Medical abbreviations, http://www.abbreviations.com/acronyms/MEDICAL
7. Genetics Home Reference, http://ghr.nlm.nih.gov/
8. Diagnosia, http://www.diagnosia.com/en/