ShARe/CLEF eHealth 2013 Normalization of Acronyms/Abbreviations Challenge

Jon D. Patrick, Leila Safari, Ying Ou

Health Language Laboratories, School of Information Technologies
The University of Sydney, NSW, Australia

jonpat@it.usyd.edu.au, lsaf7301@uni.sydney.edu.au,
yiou6374@uni.sydney.edu.au

Abstract.

Objective: Abbreviations and acronyms are widely used in the clinical documents. This paper describes using of a machine learner to automatically extract spans of abbreviations and acronyms from clinical notes and map them to the UMLS (Unified Medical Language System) CUI (Concept Unique Identifier).

Tasks: A Conditional Random Field (CRF) machine learner was used to identify abbreviations and acronyms. Firstly, the training data was converted to the CRF format. The different feature sets were applied with 10-fold cross validation to find the best feature set to create the machine learning model. Secondly, the identified spans for abbreviation/acronyms were mapped to the UMLS (Unified Medical Language System) CUIs. Thirdly, a rule based engine was applied for disambiguation of terms with multiple abbreviations or acronyms.

Approach: A novel supervised learning model was developed that incorporates a machine learning algorithm and a rule-based engine. Evaluation of each step included precision, recall and F-score metrics for span detection and accuracy for CUI mapping.

Resources: Several tools which were created in our laboratory were used, including a Text to SNOMED CT (TTSCT) service, Lexical Management System (LMS) and Ring-fencing approach. Also a set of gazetteers which had been created from the training data was employed.

Results: A 10-fold cross validation on the training data showed 0.911 of precision, 0.887 of recall and a F-score of 0.899 for detecting the boundary of abbreviation/acronyms and an accuracy of 0.760 for CUI mapping while the official results on the test data showed strict accuracy of 0.447 and relaxed accuracy of 0.488 which is the third team out of the five participating teams. A supervised machine learning method with mixed computational strategies and rule based method for disambiguation of expansions seems to provide a near-optimal strategy for automated extraction of abbreviation/acronyms.

1 Introduction

Clinical notes usually contain a large number of abbreviations and acronyms without mention of their definition. Also, they often have multiple expansions related to the context in which they have been used. For instance “BS”, may have two different expansion of “Bowel Sound” or “Breath Sound”. The context which the “BS” is used may help unlock the meaning of this abbreviation. In addition, the type of clinical document may affect the encoding of an abbreviation or acronym. For instance, the expansion of an abbreviation may be different if it is used in a discharge summary compared to a radiology report. Although the context and the document type may create some criteria for interpretation of the ambiguous abbreviations or acronyms in the clinical notes, correctly expanding them is still a challenging task for NLP systems. The main purpose of the current work is to map abbreviations and acronyms from clinical documents to UMLS (Unified Medical Language System)[1] CUIs (Concept Unique Identifier) based on the guidelines provided for ShARe/CLEF eHealth Task2[2]. The document types which have been provided for training include discharge summaries, ECG reports, echo reports, and radiology reports. As explained previously the same abbreviation and acronym may have different expansions and consequently different CUIs in different document types. A Conditional Random Field machine learner (CRF) [3] has been used to identify the spans of the provided text which are an abbreviation or acronym and then created a rule based engine to map these spans to the UMLS CUIs.

This paper is organized as follows: Section 2 covers the related works. Section 3 presents the methods which have been used to identify spans of abbreviation/acronyms and mapping them to CUIs. Section 4 explains experimental results followed by a discussion and conclusion.

2 Related works

Successful NLP techniques have been invented for Named Entity recognition and concept extraction in the general domain while the same tasks are more challenging in the clinical domain. Extracting concepts like drug names, diagnosis and symptoms has attracted several researchers. Patrick et.al [4], developed a novel supervised learning model that incorporates two machine learning algorithms and several rule-based engines to automatically extract medication information related to drug names, dosage, mode, frequency, duration and reason for administration of a drug from clinical records producing an F-score of 85.65%. The Mayo Clinic developed information extraction system [5] to process and extract information from free-text clinical notes including named entities such as diseases, signs/symptoms, anatomical sites and procedures. Attributes related to the named entities – context, status and relatedness to patient – were also extracted from the text.

Named entity recognition or concept extraction and classification tasks usually involve detecting and interpreting abbreviations and acronyms which makes them more complicated in both the general and medical domains. In [6], four methods including a rule based method and decision tree classifiers were used for detecting abbreviations and then decoding the detected abbreviations using a simple inventory. Their best detection method reached to 91.4% of recall and 80.3% of precision. In addition, the authors of [7], compared the performance of three existing clinical NLP systems including MetaMap, MedLEE AND cTAKES in handling abbreviations. Based on their evaluation, the systems achieved suboptimal performance in abbreviation identification with F-scores ranging from 16.5% to 60.1% while MedLEE was the best system with more than 60 per cent of F-score for detecting all abbreviations and more than 70 per cent for identifying clinically relevant abbreviations. They concluded that, identification of clinical abbreviations is a challenging task and the existing clinical NLP systems need incorporation of more advanced abbreviation recognition modules.

The authors of [8] focused on improving the quality of biomedical acronym sense inventories or acronym disambiguation by improving existing approaches which employ sense inventories of acronym long form expansions from the biomedical literature. They used subsequent application of a semantic similarity algorithm and evaluated their approach on a reference standard developed for only ten acronyms. 78% of long forms mapped to concepts in the UMLS while synonymous long forms identified with a sensitivity of 70.2% and a positive predictive value of 96.3%.

3 Methods

3.1 Challenge Requirements

The main objective of the Task2-ShARe/CLEF eHealth challenge is to identify and map acronyms and abbreviations found in clinical texts to a unique identifier from a controlled vocabulary. Based on the annotation guidelines provided for Task2 an acronym or abbreviation is defined as a substitute for a concept in the text. So, the task consists of two main steps: firstly, identifying and annotating acronyms and abbreviations in the clinical documents and secondly, assigning to acronyms and abbreviations a concept unique identifier from the defined medical terminology, UMLS. Some of the annotations may not match UMLS concepts and should be assigned the value "CUI-less". Also, based on the guidelines some of the abbreviations and acronyms like measurement units, non-medical acronyms or abbreviations etc. have to be excluded from the final results.

3.2 Corpus Description

The dataset for Tasks 2 consists of de-identified clinical free-text notes from the MIMIC II database, version 2.5 (mimic.physionet.org). A set of 200 notes are provided for the training task and 100 notes are provided for testing. Notes were authored in the ICU setting and note types include discharge summaries, ECG reports, echo reports, and radiology reports. For this task, the focus is normalization of pre-annotated acronyms/abbreviations to UMLS concepts. Annotators were instructed to annotate all acronyms/abbreviations that were contained in narratives and not contained in a list. Annotators were multiple nursing students trained for this task, followed by an open adjudication step [9].

3.3 The Classification Strategy

Figure 1 demonstrates the main work-follow for extracting abbreviation/acronyms from the clinical documents and mapping them to the UMLS CUIs.

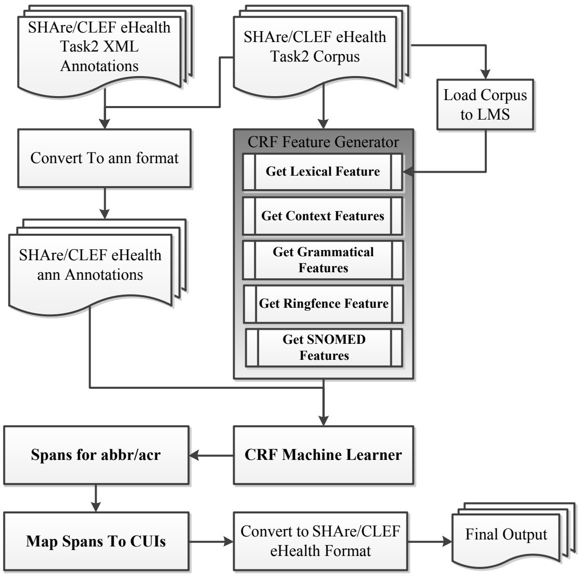


Fig. 1. Workflow of mapping abbreviations/acronyms to UMLS CUIs in ShARe/CLEF eHealth Task2

A Conditional Random Field machine learner (CRF) was used to identify “Abbreviation/ Acronyms” in this work. Firstly, the training data is converted to the CRF format. The CRF format is like a spread sheet in which each column represents a feature and the last column represents the output tag. BIO tagging notion [10] was used. The token tags with class information were converted to B-ENTITY, I-ENTITY and O to represent the beginning of an entity (Abbreviation-Acronym here), inside an entity (not at the beginning) and not a member of an Abbreviation-Acronym structure respectively. The boundary of an Abbreviation-Acronym structure begins with a B label and ends with either an O label or another B label, indicating a new Abbreviation-Acronym structure.

Different feature sets have been applied with 10-fold cross validation to find the best model. Moreover, five categories of features include Context features, Lexical features, Grammatical features, Ring-fence feature and SNOMED CT (Systematized Nomenclature Of Medicine Clinical Terms) features which have been used to construct the machine learning model. The base line feature set were the tokens with a context window of 5 words, that in addition to a token itself 2 tokens before and 2 tokens after the token were also included in the computation. Then the other features were applied sequentially to find out the optimum feature set. In the feature selection process firstly the CRF feature generator was added to train the model and compute the results and record the performance. If the performance increased the F-score with adding a feature, this feature was thought to be useful and it was retained, otherwise, it was removed from the feature set.

To be able to use the tools which have already been developed in our laboratory there was a need to do some pre-processing tasks on the corpus and annotations which the challenge organizer has provided for the Task2. One of these tasks was converting the annotations from XML or pipeline format to our .ann format.

In addition, the whole corpus was loaded into our Lexicon Management System (LMS). The LMS takes care of all new lexical knowledge generated by experts and automatic agents (Knowledge Discovery) and feeds it into the verification process or any other process that needs this information as a Knowledge Reuse process. So, by using the LMS the lexical features of the whole tokens in the training corpus were prepared to feed to the CRF feature generator. LMS initially categorizes the types of all tokens in the corpus as “Known”, “Unknown” or “Unseen”. “Known” means the primary characteristics have been defined for tokens, “Unknown” means the tokens are not resolved yet and “Unseen” means the tokens are un-reviewed. LMS enables checking each “Unseen” and “Unknown” token and add any information about that token to make it known. Spelling corrections, expansions and semantic categories can be set to make a token as known. Moreover, the lexicon is not a simple list of words but an organization of the words into semantic groups and the form of different representations of words. The following semantic groups are defined in the LMS as the words class of the tokens in the corpus or the whole lexicon:

- **Compound Words**: In a great deal of clinical terminology, productive forms of words are regularly used. An example is the word vesicle which has the combining form vesico-. The convention will be that the combing form is shown with the hyphen in the LMS, and the canonical form of the compound will include the hyphen, e.g. vesico-ureturic. Compound words are usually defined by two words separated by a non-letter character, typically a hyphen or slash. The hyphen carries the usual morphological interpretation, but the slash is still to be resolved.
- **Neologisms:** These are the words constructed to represent new forms typically used in names of organizations or products, e.g. Bayview, HealthCare. This excludes drug names which although neologisms are not included in this category but listed separately.
- **Abbreviations**: Shortened forms of words that are not acronyms. e.g. using “backgrd” instead of “background”.
- **Acronyms:** Words which are formed from the first letters of a phrase. The letters are usually in uppercase and should be preserved in their orthographic form.
- **Automatic:** The words that have been processed and categorized by direct computational methods without manual intervention.
- **Named Entity**: A set of classifications of different entity types like drug names, equipment, person names, locations, etc.

Using the above facilities in the LMS valid properties such as spelling corrections and expansion of abbreviation/acronyms were assigned and also assigned semantic classes to tokens to resolve unknown and unseen tokens. Finally, all the properties of the known tokens were extracted from the LMS and applied as one or a set of features in the machine learning model.

3.4 Abbreviation/Acronym Annotation Experiment

To find out the best feature set to feed to the CRF machine learner five categories of features have been used in the experiments. They were Context features, Lexical features, Grammatical features, Ring-fence features and SNOMED features.

- **Context Features**. Context features provides the content information for a token. The surrounding words usually convey useful information about a token which help in predicting the correct tag for each token. This feature has been used with a window of five, meaning that in addition to the token itself, the 2 tokens before and the 2 tokens after the target token are used as features for predicting the output tag.
- **Orthographic Feature**s. Includes the case tag with the values “Lower” for the tokens with all lowercase characters, “Upper” for the tokens with all uppercase characters and “Title”, for the tokens which start with an uppercase character but follow with the lowercase ones.
- **Lexical Features**. Include the expansions of abbreviations/acronyms and correction of misspelt words. As explained above, the LMS provides most of the required lexical features. In addition the lowercase form of tokens has been used as another feature.
- **Grammatical Features**. Include Lemma, part of speech (POS) and chunk features. The GENIA Tagger has been used to produce these features from the training set. By applying the lemma form of the words a more general description of the words has been possible. Also, as a low level grammatical information the POS tags of the words will help in determining the boundaries of instances. The chunk feature in a similar way assists in determining expression boundaries.
- **Ring-fence Feature**. The existence of complex and compound abbreviations and acronyms like “R Tib/FIb XR”, “Abd U/S” or “PERC G/G-J TUBE PLMT” necessitate a solution to welding these forms together. The ring fencing method which was originally invented in this laboratory to identify complex patterns like scores and measurements was used for this task. The basic idea is to put a fence around a group of tokens and dose not let the tokenizer separate them into smaller chunks but rather to make them an indivisible token. To accomplish this task a process of running a Trainable Finite State Automata (TFSA) [11] on intended phenomena over the text is required.
- **SNOMED Features**. The final features which we utilized in our experiments were the results from the TTSCT service provided in this laboratory on the training corpus [12]. TTSCT takes free text and identifies text segments equivalent to SNOMED CT concepts. The algorithm utilizes a dynamic programming search en-

gine to match different parts of the text with SNOMED CT description terms. To maximize the Recall different generalization techniques are applied and the algorithm also detects negations and excludes negated concepts. The run time of the algorithm is in polynomial order (O(n3)) and the F-score is around 70% [12]. By applying TTSCT the three features of SNOMED CT term, SNOMED CT concept id and also SNOMED CT top category are available to be used in the feature generator engine. For instance, for the token "headache" in the corpus it gives 3 features of "Headache" as a term, "25064002" as concept id and "Clinical Finding" as SNOMED CT top category.

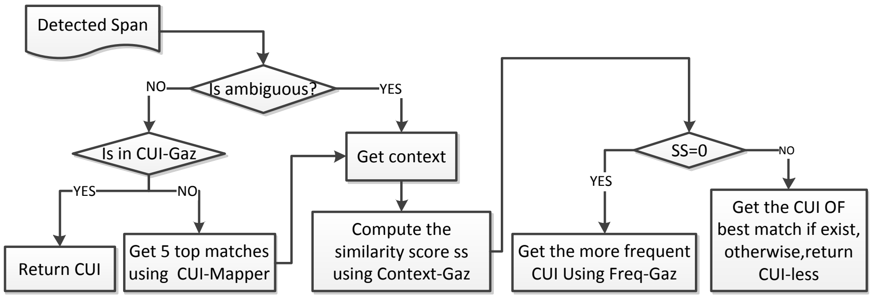
As mentioned above to find out the best feature set in the feature selection process each feature is sequentially added to the CRF feature. The feature was retained in the feature set if it increased the F-score otherwise it was removed from the feature set. So, from the above features which were tested in the experiment only orthography features did not help in increasing the performance of the machine learner and consequently it was removed from the feature set.

3.5 Mapping the Spans to CUIs

After correctly finding the spans of abbreviation/acronyms in the text, the next step was to map them to the UMLS CUIs. To accomplish this task a simple java program (CUI_Mapper) was developed which employed the APIs provided by the UMLS web site [1] to map a piece of text to the CUIs. But the problem was that it returned a list of possible matching CUIs for an abbreviation/acronym. Another barrier was that even in the gazetteer created from the training data there were multiple CUIs for some abbreviation/acronyms as they come from various document types (discharge summaries, ECG reports, etc.). For disambiguation of the expansions and CUIs for the spans a rule based engine was created. It relies on the context features and also frequency of the usage of CUIs. In addition several gazetteers were created from the training data, including:

1- **CUI-Gaz**: which maps a reference to an abbreviation/acronym to a CUI.
2- **Expansion-Gaz:** which maps a reference to an abbreviation/acronym to its expansion form. This gazetteer is created from the silver standard annotations of the abbreviation/acronyms provided by the challenge organizer for task 2.
3- **More-Expansion-Gaz:** In addition to the abbreviations and acronyms which were annotated in the silver standard, there were some more abbreviations or acronyms which were either incorrectly annotated or may excluded from the final results according to the exclude list which is defined in the guidelines. This set of abbreviations and acronyms which were identified using the LMS, helped in improving results in our experiments.
4- **Freq-Gaz**: which maps a CUI to its frequency of usage in the whole corpus.
5- **Context-Gaz:** which maps ambiguous abbreviation/acronyms to surrounding tokens. We have applied tokenisation and lemmatization tasks on the surrounding texts of the ambiguous spans from the training data to prepare lists of surrounding tokens for them.

Moreover, a list of all ambiguous or multi-expansion abbreviations/acronyms has been created. Figure 2, shows the main steps of the mapping algorithm:



Detected Span
Is ambiguous?
NO
YES
Is in CUI-Gaz
YES
NO
Return CUI
Get 5 top matches using CUI-Mapper
Get context
Compute the similarity score ss using Context-Gaz
SS=0
YES
NO
Get the more frequent CUI Using Freq-Gaz
Get the CUI OF best match if exist, otherwise,return CUI-less

Fig. 2. Main steps of CUI mapping

According to the Figure2, there are different paths to follow to match a correct CUI or CUI-less for a span:

1. If the span is not in the ambiguous list but there is a match for it in the CUI-Gaz simply returns the matched CUI from the CUI-Gaz.
2. If the span is not in the ambiguous list and there is not a match for it in the CUI-Gaz :
   (a) Get the 5 top matches using the CUI-Mapper;
   (b) Get the context feature of the span;
   (c) Compute similarity scores (SS) among the span's context features and 5 top matches based on the number of common words among them and return the CUI with the most similar match.
3. If the span is in ambiguous list, get context features for the span from the corpus and then compute the similarity score SS between this context and each of the candidate expansions for the span in the Context-Gaz.
   (a) If the value of SS is zero for all candidate expansions it means there was not any similarity among the contexts of the detected span and the contexts of the previously annotated spans in the training set. So, return the more frequent CUI using CUI-Gaz and Freq-Gaz.

   (b) If the value of SS is non-zero at least for one of the expansions it means the algorithm was able to find a similarity among the context of the detected span and the context of the previously annotated spans in the training set. So, match the span with the expansion with higher value of SS and return its CUI using Expansion-Gaz and CUI-Gaz.

4. If there is not any match in the step 2, or there is not any similarity or frequency score for the span in the step 3, return CUI-Less.

4 Results and Discussion

Table 1 represents CRF results for detecting the spans for abbreviation/acronyms for different feature sets based on 2-fold cross validation, which was submitted to the challenge. As the number of features increases the model is elaborated and the results improve. CRF takes advantage of Context, that is, words around the target word and itself (M1). Model 1 is used as the baseline model. Then the lowercase of the tokens with window of five was added to construct model M2 which improved the F-Score from 0.665 to 0.709. Adding the only orthography feature in our experiment, case feature, to construct model3 decreased the F-Score, so this feature was removed from the feature set. In model M4, the expansion form of the known tokens was added from the created gazetteers of the training data as a feature and achieved about a 13 per cent improvement in F-Score in comparison with the model M2. In the model M5 the UMLS CUI was applied as another feature. A gazetteer was created to map the known abbreviation/acronyms from the training set to the CUIS. Applying this feature improved the F-score for about half per cent. Adding the ring-fence tag to the feature set improved the F-Score by another half per cent. But more improvement should be possible by defining more patterns to the ring-fencing algorithm to capture complex abbreviations and acronyms.

By adding the 3 features CID (SNOMED concept id), termtag (SNOMED Term) and cattag (SNOMED top category) from TTSCT, there was a slight increase in the F-Score for termtag but a slight decrease for the other 2 features. CID is similar to CUI not much improvement was expected in using CID. Consequently, the CUI was kept in the final model as a feature. But it was surprising that applying SNOMED CT top category decreased the results. However in the next experiment this feature helped to improve the results (model M13).

Applying the chunk or grammatical features using the GENIA Tagger (model 10), include Lemma, Part of speech and Chunk features caused a decrease in the results and these features should be removed from the final feature set.

Table 1. CRF results with 2-fold cross validation for 16 different feature sets for BIO token tagging

Model to detect abbreviation/acronym spans	TP	FP	FN	P	R	F	NUM
M1 = bag of word with window(5)	1988	334	1672	0.856	0.543	0.665	3660
M2 = M1+ lower case with window(5)	2193	331	1467	0.869	0.599	0.709	3660
M3 = M2+ case feature	2050	184	1610	0.918	0.560	0.696	3660
M4 = M2+ expansion feature	2838	327	822	0.897	0.775	0.832	3660
M5 = M4+ CUI	2895	361	765	0.889	0.791	0.837	3660
M6 = M5+ ring tag	2897	360	763	0.889	0.792	0.838	3660
M7 = M6 + CID	2887	371	773	0.886	0.789	0.835	3660
M8 = M5 + termtag	2886	367	774	0.887	0.788	0.835	3660
M9 = M8 + cattag	2883	368	777	0.887	0.788	0.834	3660
M10 = M 9 + lemma+ postag+ chunk feature	2888	384	772	0.883	0.789	0.833	3660
M11= M 10 with more expansions	2933	393	727	0.882	0.801	0.840	3660

M12= M6+more expansions	2927	389	733	0.883	0.800	0.839	3660
M13= M11+ SCT top category	**2937**	**392**	**723**	**0.882**	**0.802**	**0.840**	**3660**
M14 = M13 + expansion with window(5)	2920	397	740	0.880	0.798	0.837	3660
M15 = M14 with window 3 for expansion	2934	397	726	0.881	0.802	0.839	3660
M16 = M13 window 3 for SCT category	2926	399	734	0.880	0.799	0.838	3660

In addition, there were many abbreviations and acronyms in the training set which were not annotated. They were all extracted from the training set using the LMS and created the More-Expansion-Gaz as explained before, their expansion was applied in model M11 and improved the F-score from 0.833 to 0.840. Applying this on model M6 (model M12) confirmed the effects of applying more expansions in improving the results. So this feature was retained in the final model. Finally, applying the SNOMED CT top category in model M13 lead to another improvement as expected and M13 became the best model with a precision of 0.882, recall of 0.802 and the best F-Score of 0.840 for submission to the Task2 of the challenge. In models 14-16 the expansion forms and SNOMED top category with different window size was tried but they failed to improve the results.

After finding the optimum feature set to construct a model for CRF machine learner, model M10, other experiments were conducted. In a separate process, errors in the annotations provided for the ShARe/CLEF eHealth Task2 corpus were sought by performing validation on the ShARe/CLEF eHealth Task2 training data with a 100% train and test strategy. The results from 100% train and test using model M10 is illustrated in table 2:

Table 2. CRF results for 2 experiments on model 10

Experiment	**TP**	**FP**	**FN**	**P**	**R**	**F**	**NUM**
100% train and test	3545	63	115	0.982	0.969	0.975	3660

The number 115 for the false negatives reflects that there are some annotation errors in the training corpus or the machine learner has seen the tokens before but insufficient context was provided to capture those tokens as abbreviation/ acronyms. Also, the number of 63 false positives illustrates some more annotation errors in the training set. The errors of both categories were corrected manually as a first step, so that the model would not learn from the incorrect examples. In addition, more investigation on the false positives and false negatives revealed that there are some more abbreviations and acronyms which needed to be ring-fenced together as the machine learner was able to tag only part of them. So, some new patterns were defined in the ring fence training set, and also merged more expansions from the LMS with the expansions of annotations in the next steps. Considering these 3 steps as a pre-processing task the experiment was repeated with 10-fold cross validation with approximately the same feature sets. The final results of 10-fold cross validation are shown in table 3 which illustrates approximately the same effect to the 2-fold experiment by adding new features but with higher scores. The best score was recorded for Model M12 with the precision of 0.911, recall of 0.887 and F-score of 0.899.

Table 3. CRF results with 10 fold cross validation for 14 different feature sets with BIO token tagging

Model to detect abbreviation/acronym spans	**TP**	**FP**	**FN**	**P**	**R**	**F**	**NUM**
M1 = bag of word with window (5)	2354	321	1346	0.880	0.636	0.738	3700
M2= M1+ lower case of tokens with window(5)	2592	334	1108	0.886	0.700	0.782	3700
M3=M2+ case feature	2489	217	1211	0.920	0.673	0.777	3700
M4=M2+ expansion feature	3078	319	622	0.906	0832	0.867	3700
M5=M4+ CUI	3178	319	522	0.909	0.859	0.883	3700
M6=M5+ ring tag	3210	281	490	0.919	0.868	0.893	3700
M7 = M6 + cid	3229	290	471	0.918	0.873	0.895	3700
M8 = M7 + termtag	3236	297	464	0.916	0.875	0.895	3700
M9 = M8 + cattag	3244	306	456	0.914	0.877	0.895	3700
M10 = M 9 + lemma	3257	301	443	0.915	0.880	0.897	3700
M11= M 10 + POS tag	3278	321	422	0.911	0.886	0.898	3700
M12= M11+chunk feature	**3280**	**319**	**420**	**0.911**	**0.887**	**0. 899**	**3700**
M13= M11+ expansion with window(5)	3275	323	425	0.910	0.885	0.897	3700
M14 = M11 + cattag with window(5)	3269	322	431	0.910	0.883	0.897	3700

The CUI mapping algorithm on the detected spans of 2765 out of 3660 showed accuracy of 0.760 on the training data. Finally, by applying the optimum model (M12) and the CUI mapping algorithm on the official test data the final results of 0.447 for strict accuracy and 0.488 for relaxed accuracy were published by the challenge organizer in which our team was third team out of five.

5 Conclusion

A machine learning model has been introduced that was designed to participate in the ShARe/CLEF eHealth Task2 challenge. The model was based on the CRF machine learner for detecting the spans of abbreviation and acronyms and a rule-based engine to map the detected spans to the UMLS CUIs. Evaluation results showed 0.911 of precision, 0.887 of recall and 0.899 of F-score for span detection experiment based on 10-fold cross validation of the training data and an accuracy of 0.760 for CUI mapping while the official results on the test data showed a strict accuracy of 0.447 and relaxed accuracy of 0.488 in the CUI mapping in which our team was third team out of five. The results demonstrated that the performance of this system for detecting spans of abbreviation/acronyms is promising but more accurate rules are required for CUI mapping. In addition, further improvements should be possible by adding new features to the model and also enhancing the performance of TTSCT and ring-fencing algorithms. In addition, not all the features which the LMS provides for lexical verification have been used. All these tasks will be our focus of interest in future work.

6 Acknowledgments

This work is supported by the Shared Annotated Resources (ShARe) project funded by the United States National Institutes of Health: R01GM090187. We also would like to give a special thanks to Dr. Stephen Crawshaw and other members in the Health Information Technologies Research Laboratory for their valuable contributions.

7 References

[1] "Unified Medical Language System". [cited 15 March 2013].

[2] H. Suominen, S. Salantera, & S. Velupillai, et al. "Three Shared Tasks on Clinical Natural Language Processing". *Proceedings of CLEF 2013*.

[3] "CRF++. Yet Another CRF toolkit.". [cited 15 Mar 2013].

[4] J. Patrick, & M. Li, "High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge". *J Am Med Inform Assoc 2010*. vol. 17, pp. 524-527, 2010.

[5] G. K. Savova, K. Kipper-Schuler, J. D. Buntrock, & C. G. Chute, "UIMA-based Clinical Information Extraction System
", in *LREC 2008 workshop: towards enhanced interoperability for large HLT systems: UIMA for NLP* 2008.

[6] H. Xu, P. D. Stetson, & C. Friedman, "A Study of Abbreviations in Clinical Notes". *AMIA Annu Symp Proc*. vol., pp. 821-825, 2007.

[7] Y. Wu, Joshua C. Denny, T. Rosenbloom, R. A. Miller, D. A. Giuse, & H. Xu, "A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries". *AMIA Annu Symp Proc*. vol., pp. 997-1003, 2012.

[8] G. B. Melton, S. Moon, B. McInnes, & S. Pakhomov, "Automated identification of synonyms in biomedical acronym sense inventories", in *Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents*, Association for Computational Linguistics: Los Angeles, California. pp. 46-52, 2010.

[9] "https://sites.google.com/site/shareclefehealth/". [cited.

[10] F. Sha, & F. Pereira, "Shallow parsing with conditional random fields", in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, Association for Computational Linguistics: Edmonton, Canada. pp. 134-141, 2003.

[11] J. Patrick, & M. Sabbagh, "An Active Learning Process for Extraction and Standardisation of Medical Measurements by a Trainable FSA", in *Computational Linguistics and Intelligent Text Processing*, A. Gelbukh, Editor.Springer Berlin Heidelberg, 2011: pp. 151-162.

[12] J. Patrick, Y. Wang, & P. Budd, "An automated system for conversion of clinical notes into SNOMED clinical terminology", in *Proceedings of the fifth Australasian symposium on ACSW frontiers - Volume 68*, Australian Computer Society, Inc.: Ballarat, Australia. pp. 219-226, 2007.