Normalizing clinical terms using learned edit distance patterns

RECEIVED 9 February 2015
REVISED 18 June 2015
ACCEPTED 24 June 2015
PUBLISHED ONLINE FIRST 31 July 2015

Rohit J Kate

ABSTRACT

Background Variations of clinical terms are very commonly encountered in clinical texts. Normalization methods that use similarity measures or hand-coded approximation rules for matching clinical terms to standard terminologies have limited accuracy and coverage.

Materials and Methods In this paper, a novel method is presented that automatically learns patterns of variations of clinical terms from known variations from a resource such as the Unified Medical Language System (UMLS). The patterns are first learned by computing edit distances between the known variations, which are then appropriately generalized for normalizing previously unseen terms. The method was applied and evaluated on the disease and disorder mention normalization task using the dataset of SemEval 2014 and compared with the normalization ability of the MetaMap system and a method based on cosine similarity.

Results Excluding the mentions that already exactly match in UMLS and the training dataset, the proposed method obtained 64.7% accuracy on the rest of the test dataset. The accuracy was calculated as the number of mentions that correctly matched the gold-standard concept unique identifiers (CUIs) or correctly matched to be without a CUI. In comparison, MetaMap's accuracy was 41.9% and cosine similarity's accuracy was 44.6%. When only the output CUIs were evaluated, the proposed method obtained 54.4% best *F*-measure (at 92.1% precision and 38.6% recall) while MetaMap obtained 19.4% best *F*-measure (at 38.0% precision and 13.0% recall) and cosine similarity obtained 38.1% best *F*-measure (at 70.3% precision and 26.1% recall).

Conclusions The novel method was found to perform much better than the MetaMap system and the cosine similarity based method in normalizing disease mentions in clinical text that did not exactly match in UMLS. The method is also general and can be used for normalizing clinical terms of other semantic types as well.

Keywords: clinical terms, normalization, edit distance

BACKGROUND AND SIGNIFICANCE

Clinical terms are found in clinical text with several variations. These variations could be because of morphological alternations or differing writing conventions or even because of typographical errors. For example, the term "haemoglobin" may be found in clinical text with an alternate spelling "hemoglobin" or with a typographical error "heamoglobin." Similarly, "Addison's disease" may be found as "Addison Disease," "metastatis" as "metastases," "cyanotic" as "cyanosis," "z-plasty" as "z-plasties," etc., which are examples of morphological variations. Due to variations, clinical terms often do not exactly match the terms in standard terminologies or ontologies, which impedes their automatic coding and normalization process for downstream applications. Although terminology resources such as the Unified Medical Language System (UMLS)[1] provide multiple variations and synonyms for most of the clinical terms, they do not exhaustively cover them. For example, we found that in the dataset of SemEval 2014 Task 7[2] out of total 13 845 disease and disorder mentions whose concepts were known to be present in UMLS, 23.7% of them could not be mapped to any of the concepts in UMLS through exact matching.

The task of matching a given term into a standard terminology is also called *normalization*. Some existing systems that normalize clinical terms do so by applying manually developed string-matching or approximation rules.[3–6] However, manually encoding all possible variations of clinical terms is not only a formidable task but typically also results in limited accuracy and coverage.[7] For example, in the experimental results of this paper we show that MetaMap system,[8] which internally uses such rules for normalization of clinical terms, not only missed mapping many terms to their correct concepts in UMLS but also erroneously mapped many other terms to incorrect concepts.

For normalizing multi-token clinical terms to their concepts in standard terminology, some existing systems use a method similar to the document matching method used in information retrieval.[9] The clinical terms are treated like documents and their tokens as document terms, and then a similarity metric, typically cosine similarity, is used to match them with the standard terminology.[10–12] Although this is a reasonable approximate matching method, it has obvious shortcomings. First, token-based similarity is not always suitable for matching clinical terms. For example, "left ventricular cardiac dysfunction" is very different from "ventricular cardiac dysfunction," but the token-based similarity between the two terms will be very high. Second, it is easily affected by morphological and typographical variations which are very common in clinical terms. For example, the method is unlikely to match "calculous pancreas" to "calculus pancreas" which, in fact, refers to the same concept.

The objective of the research presented in this paper was to design and evaluate a novel method for normalizing clinical terms based on rules that are automatically learned from known variations of clinical terms. To the best of our knowledge no one has presented a method that automatically learns normalization rules for clinical terms.

Our method of learning normalization rules is based on edit distance, which is a measure of typographical similarity between two terms. We used a particular type of well-known edit distance called Levenshtein distance.[13] It measures the minimum number of edit operations of insertions, deletions, and substitutions that are needed

Correspondence to Rohit J. Kate, Department of Health Informatics and Administration, University of Wisconsin-Milwaukee, 2025 E Newport Ave, Milwaukee, WI 53211, USA; katerj@uwm.edu

 For affiliation see end of article.

to convert one term into another term. We call the sequences of such operations edit distance patterns. Our approach first finds edit distance patterns between every pair of terms in UMLS that represent the same concept. Such patterns are then generalized by finding the commonality between them. However, over-generalization is avoided by measuring how often they lead to correct or incorrect normalization of clinical terms within UMLS. The edit distance patterns thus learned and validated become our normalization rules. The entire method does not require any human annotation effort and only utilizes existing resources such as UMLS. Although edit distances and other measures have been used before for normalization by computing similarity between terms and then using ad hoc threshold values to determine if they match,[14,15] our method uses edit distances in a very different way, namely to learn patterns that capture known variations between clinical terms using a resource such as UMLS. Learning methods based on edit distances have been used before for duplicate detection in database records,[16] but to the best of our knowledge, no one has used an approach that learns patterns based on edit distances.

We evaluated our method using the dataset of Task 7 of SemEval 2014,[2] which was built on the dataset of Task 1 of ShARe/CLEF eHealth Evaluation Labs.[17,18] The dataset consists of disease and disorder mentions in clinical text and their mappings to the SNOMED CT[19] part of UMLS. We compared the performance of our method with that of the well-known MetaMap system as well as with a method based on cosine similarity and found that our method performs best. Although our method was evaluated on a dataset of only diseases and disorders due to its availability, the method itself is general enough and can be applied to learn normalization rules for clinical terms of other semantic types as well.

While Subtask A of SemEval 2014 Task 7 was extraction of clinical terms from clinical text and Subtask B was their normalization, the focus of the presented research was only on normalization of clinical terms (i.e., only Subtask B) and not their extraction for which several machine learning based methods have been already developed.[20–22] Since entity extraction process is typically specific to a particular semantic type (e.g., extracting disease and disorder mentions), we also do not consider disambiguation of clinical entities as part of the normalization process. Thus the proposed method expects clinical terms already extracted from text for a specific semantic type which it then maps to a standard terminology. This is how the Subtask B of SemEval 2014 Task 7 was also defined. Also, when we compare our method to MetaMap, we are only comparing against MetaMap's normalization ability out of several of its functionalities.

METHODS

This section describes our novel method for automatically generating normalization rules for clinical terms. In order to learn these rules, the only resource our method needs is a list of clinical terms and their known variations or synonyms. We used UMLS as our resource. In this section, we first describe how the rules are generated, then how they are validated, and finally give an efficient algorithm for the entire process.

Edit Distance Patterns and Their Generalizations

Our method first computes Levenshtein edit distances[13] between every pair of clinical terms that represent the same concept in UMLS (or its subset of a particular semantic type if normalization rules are to be for that semantic type). The Levenshtein edit distance computes the minimum number of edit operations of insertions, deletions, and substitutions needed to convert one term into another. For example, the term "cyanotic" can be converted into term "cyanosis" in minimum of two steps, by substituting the "t" by "s" and the last "c" by "s." Hence the edit distance between them is two. There is a fast dynamic programming algorithm to compute edit distances.[13] That algorithm also gives the sequence of minimum edit distance operations needed to convert one term into another. For the above example, the sequence will be "BEGIN SAME c SAME y SAME n SAME o SUBSTITUTE t/s SAME i SUBSTITUTE c/s END", which is illustrated in Figure 1. In the paper, we will call such a sequence of edit operations an *edit distance pattern*. This is the form in which our method learns normalization rules, and in the paper we will use the two terms alternatively. Note that we have included BEGIN and END symbols at the beginning and at the end of the pattern; their utility will be described shortly. We consider edit distance patterns only at the character level and not at the word level because edit operations at the word level can always be represented using edit distance patterns at the character level.

An edit distance pattern thus generated between two terms can only be used to convert the first term into the second term and as such cannot be applied to any other new term. Hence edit distance patterns need to be generalized before they can be applied to other terms. The generalization should capture the common patterns of variations between clinical terms. Given two edit distance patterns, we define generalization between them as the *longest contiguous common pattern that includes all the edit operations* of insertions, deletions, and substitutions. Thus the generalization process only generalizes over "SAME," "START," and "END" steps occurring outside of the edit operations. For example, the generalization of the two edit distance patterns, one for converting "cyanotic" to "cyanosis" and another for converting "thrombotic" to "thrombosis", will be the edit distance pattern "SAME o SUBSTITUTE t/s SAME i SUBSTITUTE c/s END". This is illustrated in Figure 2. This pattern can apply only to terms that end with "otic" and upon application will convert them to end with "osis." Hence this pattern essentially captures the variation that if a term ends with "otic" then it can be normalized to a term that ends with "osis" rest of the term being the same. Using this generalized edit distance pattern, one will thus be able to normalize, say, "fibrotic" to "fibrosis" even if "fibrotic" may not be mentioned in a resource like UMLS.

The above example also illustrates the utility of "END" symbol, which restricts the edit distance pattern to match only at the end of a term and not somewhere in between. Similarly, "BEGIN" symbol helps to capture variations that are specific at the beginning of terms.

Validating Edit Distance Patterns

Our method further generalizes the generalized edit distance patterns in the same way by finding the commonality between them in order to obtain patterns with even wider applicability. However, not all the generalized patterns thus obtained may be good and some may, in fact, change the meaning of the terms when applied. This is particularly true if the patterns get over-generalized—for example, a pattern that says change every "t" to "s." Hence it is very important to also validate the edit distance patterns. Our method validates them by applying them to other terms in UMLS (or its subset of a particular semantic type) and separately counting how often the converted term is a valid or an invalid variation of the original term. Specifically, when a pattern is applied to a UMLS term and the converted term is also in UMLS with the same concept unique identifier (CUI), then the pair of terms is counted as a *positive* example. However, if the converted term is in UMLS but it has a different CUI (meaning it is a different concept), then the pair of terms is counted as a *negative* example. If the converted term does not match in UMLS then the pair is neither counted as a positive nor as a negative example.

Figure 1: An illustrative example showing the edit distance pattern that converts the term "cyanotic" into "cyanosis."

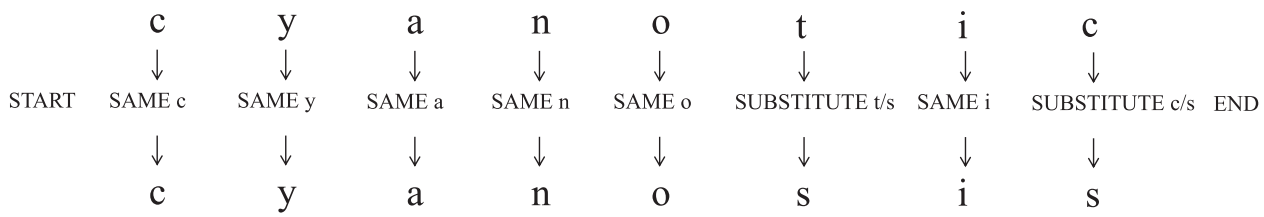
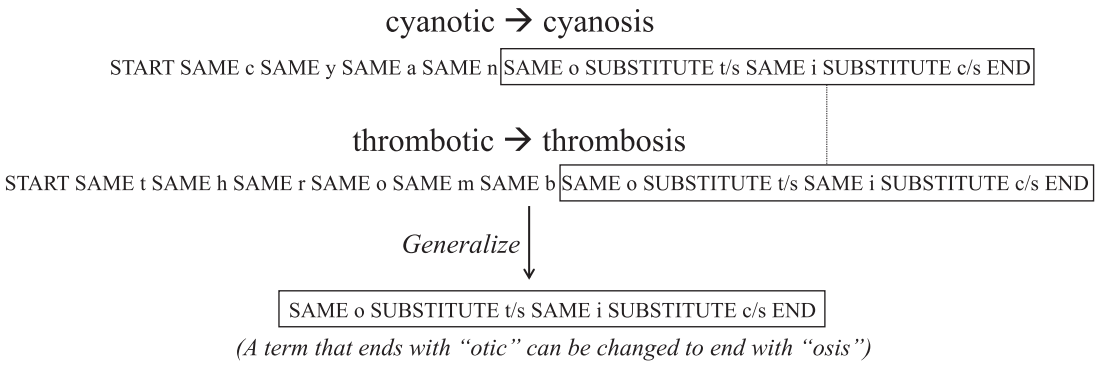


Figure 2: An illustrative example of generalization of two edit distance patterns. Generalization is defined as the longest contiguous common pattern that includes all the edit operations of insertions, deletions, and substitutions.



The number of positives and negatives thus computed for a pattern indicate how accurate the pattern is in generating correct variations of terms for normalization. Additionally, the more the number of positives the more common is the variation that the pattern captures. Patterns with several positives and very few or no negatives are thus very good patterns. We measure goodness of a pattern in terms of its *score,* which is computed as $p/(p+n+1)$, where p is the number of positives and n is the number of negatives found for the pattern. This is a simple form of the well-known m-estimate formula.[23] In our experiments, we use the score of a pattern as a measure of its confidence in normalizing clinical terms.

The Complete Efficient Algorithm

Given the vastness of UMLS, a naïve implementation of the above procedure for finding edit distance patterns, generalizing them, and counting their positive and negative examples will be computationally almost intractable. Hence we developed an efficient algorithm which is described in this subsection and in the Appendix A. Before we begin, we define a term that will be useful in describing the algorithm. We will call the contiguous part of an edit distance pattern from its first edit operation (i.e., insert, delete, or substitute) to its last edit operation as *edit-only pattern.* For example, for the edit distance pattern "START SAME c SAME y SAME n SAME o SUBSTITUTE t/s SAME i SUBSTITUTE c/s END," the edit-only pattern will be "SUBSTITUTE t/s SAME i SUBSTITUTE c/s." Note that an edit-only operation could include "SAME" symbols inside it.

Our algorithm exploits the following observations to make the computation efficient. The first observation is that an edit distance pattern and all its possible generalizations share the same edit-only pattern. This is because of the way we define generalization of two patterns as the longest common contiguous pattern that includes all the edit operations. To exploit this observation, our algorithm, in fact, begins with an edit-only pattern and then specializes it on left and right sides. The second observation is that a positive (or negative) example of an edit distance pattern will also be a positive (or negative) example of all its generalizations. This is because if a pattern applies to a term and converts it into another term, then this will be true for any of its generalizations as well. Hence our algorithm computes positive and negative examples only for specialized patterns from which positive and negative examples of more general patterns are determined. The final observation is that when edit distance pattern is computed between two terms, one can also know if the term pair constitutes a positive example or a negative example for that pattern based on whether the two terms represent the same concept or not. Our algorithm exploits this observation and simultaneously computes edit distance patterns as well as positive and negative examples. The pseudo-code of our algorithm and its full description is in Appendix A.

RESULTS AND DISCUSSION

We implemented the algorithm described in the previous section and ran it on the concepts of UMLS restricted to disease and disorder semantic type in order to generate normalization rules specific to that semantic type. We also restricted to only the concepts that occur in Systematized Nomenclature of Medicine—Clinical Terms (SNOMED CT) as is the case with the SemEval 2014 dataset that we used for evaluation. However, the descriptions (terms representing the concepts) were not restricted to SNOMED CT and could have come from any other source. There were a total of 88 638 concepts and 88 109 of them had at least two descriptions. We restricted patterns to have

at most 10 as the length of their edit-only patterns because patterns longer than that rarely apply to new terms. We also required them to have at least 3 positives, and more positives than negatives. Our program learned a total of 11 832 edit distance patterns, which form our normalization rules. All these patterns and our code are available on our web-site http://www.uwm.edu/~katerj/normalize for download.

A few illustrative patterns are shown in Table 1. The first two patterns picked up the equivalence between American and British spellings, but note that the second pattern is more general than the first pattern and hence also has more positive as well as more negative examples. The first pattern has no negative example—that is, it never led to any incorrect normalization when validated within UMLS, and its score is consequently higher than the score of the second pattern. Thus, although the second more general pattern will be applicable more often than the first pattern, but the system will have higher confidence in normalizing using the first pattern whenever it applies. The third and fourth patterns capture two common typographical variations in the context of other characters. The fifth pattern tells that "of" can be inserted after "hy." The pattern that simply inserts "of" without any context had 205 negative examples and hence was not a good pattern, but interestingly, this pattern has no negative example because of the context of "hy." Such types of rules with contexts are not easy for humans to create but our method is capable of learning them automatically from data. Finally, the last illustrative pattern depicts a very specific conversion—if "alcohol" is present in a clinical term, then it can be changed to "alcoholic" for normalization. For example, it will normalize "alcohol liver cirrhosis" to "alcoholic liver cirrhosis." Although our edit distance patterns are character-based, the last two examples show that these patterns are inherently capable of editing at the word level.

We applied our normalization rules to the dataset of SemEval 2014 Task 7B for normalizing disease and disorder mentions. The normalization rules were first sorted in decreasing order of their scores and were then applied in that order. If a normalization rule applied and the converted term was present in the SNOMED CT part of UMLS, then the corresponding CUI was output with the rule's score as its confidence. If none of the rules worked and if the term was an abbreviation according to a standard list of clinical abbreviations,[24] then its full form was considered. If the rules did not apply even on the full form then the output was given as CUI-less.

Given that the focus of this paper was only on the normalization task (Task 7B), we evaluated the performance of our system independent of the extraction task (Task 7A) by using the gold-standard dataset of disease and disorder mentions already extracted from text. Although we participated in both the subtasks of SemEval 2014 Task 7,[25] in this study we could not compare normalization performance of our system directly with that of the other teams. This is because in that competition the normalization task was not separately evaluated but was evaluated in conjunction with the extraction task.[2] We used the MetaMap[8] system (2013 version) and a method based on cosine similarity[10,11] instead for comparison. We used MetaMap's options to restrict its output concepts to SNOMED CT and of disease and disorder semantic type. Given a term, MetaMap gives a list of UMLS concepts that it determines to be matching the term. This list is given in a decreasing order of its confidence in the matches. For an input term, we take the CUI of the first concept in the list as the normalization output of MetaMap, if the list is empty then the output is taken as CUI-less unless it is an abbreviation in which case its full form is considered.

We implemented a cosine similarity based method which first tokenized all the terms and removed some common morphological suffixes.[26] We did not remove prefixes because they often change the meanings of the terms. The CUI of the UMLS concept description that had the highest cosine similarity score with the input term was regarded as the output of the method. We again restricted to concepts of SNOMED CT and of disease and disorder semantic type although the descriptions of the concepts could be coming from other sources. While the method based on learned patterns as well as MetaMap did not output any CUI if no suitable match was found, the cosine similarity based method would almost always output some CUI even when the similarity score was close to zero which could degrade its performance. To prevent this, we set a minimum cosine similarity threshold of 0.7, which was determined through a pilot experiment using ten percent of the entire data and was found to maximize *F*-measure of matched CUIs. If a term failed to match above the threshold then its full form was considered if it was an abbreviation; otherwise, it was declared CUI-less.

We did not use term-frequency-inverse-document-frequency (TF-IDF) statistic with our cosine similarity method because although TF-IDF is meaningful and effective for document matching,[9] it is not suitable for clinical term normalization for two reasons. First, TF is meant to take into account multiple occurrences of tokens in a document, but clinical terms are very short and rarely contain multiple occurrences of tokens, hence TF is not very useful for clinical term normalization. Second, the purpose of IDF is to down-weigh commonly occurring tokens across documents, but in the context of clinical term normalization the commonly occurring tokens, like "non,"

Table 1: A few illustrative examples of a total of 11 832 automatically learned edit distance patterns.

	Learned Edit Distance Pattern	Positive Examples	Negative Examples	Comments
1	SAME i SAME o INSERT u SAME r SAME space	841	0	Change some American spellings to British ("ior " → "iour ")
2	SAME o INSERT u SAME r	5166	45	Change American spellings to British ("or" → "our")
3	BEGIN SAME h INSERT a SAME e SAME m SAME a SAME t SAME o SAME m	95	0	Variation: "hematom . . . " → "haematom . . . "
4	SAME i INSERT a SUBSTITUTE c/s	25	3	Example: "hyperglycemic" → "hyperglycemias"
5	SAME h SAME y INSERT space INSERT o INSERT f	128	0	Add " of" following "hy"; Example: "exstrophy urinary bladder" → "exstrophy of urinary bladder"
6	BEGIN SAME a SAME l SAME c SAME o SAME h SAME o SAME l INSERT i INSERT c	25	0	"alcohol" → "alcoholic"

"congenital," "acute," "pre," "pain," "benign" etc., are very important and always change the meaning of a clinical term. Hence common tokens should not be down-weighed for clinical term normalization.

Table 2 shows the results. The results of learned patterns, cosine similarity and MetaMap are denoted by "LP," "CS," and "MM," respectively. The first row shows the results obtained on the training and development set (note that this data was not used to train our method which used only UMLS for training). The next row shows results obtained on the test set while the last row shows results on the test set after excluding the mentions that already exactly matched in the training and development set. In all of these datasets, the mentions that already exactly matched in UMLS (also considering their full forms if the terms are abbreviations[24]) were first excluded because those are trivial cases of normalization.

The second major column of Table 2 shows accuracy which is measured as the percentage of terms for which the output was correct (whether CUI or CUI-less). We also separately evaluated the CUIs because this measures the ability of the systems to assign correct CUIs to terms known to be present in the terminology. All methods also give confidence scores to their output CUIs. Our method's confidence is same as the score of the normalization rule that was used. We used these scores to plot precision-recall curve by varying the threshold of confidence and measuring precision (fraction of output CUIs that are correct) and recall (fraction of CUIs in the dataset the system correctly outputs). We also measure best *F*-measure (harmonic mean of precision and recall) along this curve. The third major column of Table 2 shows the best *F*-measures obtained by the three methods over their precision-recall curves, while the next two major columns show precisions and recalls corresponding to the best *F*-measures. The precision-recall curves corresponding to the last row of Table 2 are shown in Figure 3 (the curves are qualitatively similar for the other rows).

As is evident, our method of learned patterns performs better than MetaMap as well as cosine similarity on all the datasets and using all the evaluation measures. Figure 3 clearly shows that our method is capable of giving very high precision on its output CUIs (92.1% precision with 38.6% recall at best *F*-measure). On the other hand, MetaMap and cosine similarity are never so precise at any reasonable recall level, although cosine similarity overall did better than MetaMap. Many mistakes were made by MetaMap because sometimes it would over-approximate terms—for example, it would match "pre renal azotemia" to "renal azotaemia," instead of matching it to "prerenal azotemia," while sometimes it would fail to approximate—for example, it would not normalize "arcus senilus" to "arcus senilis." Our method being a learning method avoids these types of mistakes. While it learns good patterns from UMLS to approximate (e.g., when to change "us" to "is"), it avoids patterns that over-approximate (e.g., a pattern that drops "pre") because such patterns will either have no positive example in the training data or will have several negative examples and hence will get a very low score.

Cosine similarity based method made many mistakes because it is not discriminative enough to know which words are important and which are not. For example, it incorrectly matched "type 2 insulin dependent diabetes mellitus" to "type 2 diabetes mellitus non insulin dependent" with a very high confidence because several words match between the two terms, but it ignored the importance of the word "non", which does not match. Note that because "non" is a very common token in clinical terms, using TF-IDF would have only further lowered its importance. The method similarly matched "chronic headaches" to "chronic cluster headaches" and "pre renal azotemia" to "renal azotemia," etc. Our proposed method avoids such mistakes because of the reasons pointed out in the last paragraph. The cosine similarity method, however, correctly matches terms when they differ in only word order—for example, it correctly matches "cholesterol embolus retinal" to "cholesterol retinal embolus" but our proposed method fails to do so because the conversion between these two terms is not a commonly encountered variation pattern. In future, one may consider combining the scores of cosine similarity and learned patterns to develop a hybrid approach.

While the precision of our method for output CUI is good, its recall is low. But we want to point out that it misses normalizing many of the mentions (resulting in low recall) because normalizing those mentions would require a thorough semantic analysis and/or use of a comprehensive medical knowledge base. For example, in the dataset the mention "neoplastic pleural thickening" should normalize to "tumor of pleura" and the mention "inability to void" should normalize to "difficulty passing urine," but these types of normalizations are beyond character-based edits and hence our method, as well as MetaMap and the cosine similarity method, fail on them.

While Table 2 showed results when the terms that exactly matched in UMLS were first excluded, Table 3 shows results on the test dataset when all the terms are included (total 7997 mentions and 1930 of them CUI-less) in order to give an idea of the relative contribution of normalization methods over exact matching. The first three rows in Table 3 show results when exact matching with UMLS, exact matching with the training and development dataset (shown as TD in Table 3), and their combination (resolving conflicts in favor of TD) are used for normalization. It may be noted that the precision for output CUIs is not 100% with exact matching because of certain amount of ambiguity and some inconsistencies in the dataset. In case a term exactly matched descriptions of multiple CUIs in UMLS, then only the first match was given as the output. Given that there is no confidence score associated with exact matching, there is no threshold to vary,

Table 2: Results comparing performances of the proposed method that uses learned patterns (LP), the method based on cosine similarity (CS), and MetaMap (MM) on different datasets of the disease and disorder mention normalization task of SemEval 2014 Task 7B.

Dataset (mentions, CUIs)	Accuracy (CUI + CUI-less)			Best *F*-measure (CUI)			Precision at best *F*-measure (CUI)			Recall at best *F*-measure (CUI)		
	LP	CS	MM	LP	CS	MM	LP	CS	MM	LP	CS	MM
Training and dev. (4823, 1584)	69.4	38.4	58.3	32.2	24.8	18.5	81.9	32.6	29.7	20.0	20.0	13.5
Test (2777, 992)	68.6	31.1	58.4	44.4	31.5	18.8	86.1	45.2	44.4	29.9	24.2	11.9
Test-exclude(1075, 570)	64.7	44.6	41.9	54.4	38.1	19.4	92.1	70.3	38.0	38.6	26.1	13.0

All the numbers are percentages. From each of the datasets, only mentions with no exact matches in the UMLS were used to obtain the results. The results shown in the last row were obtained after excluding the mentions in the test dataset that exactly matched mentions in the training and development dataset. The proposed method was trained using only UMLS and the "training and development" dataset was not actually used for training.

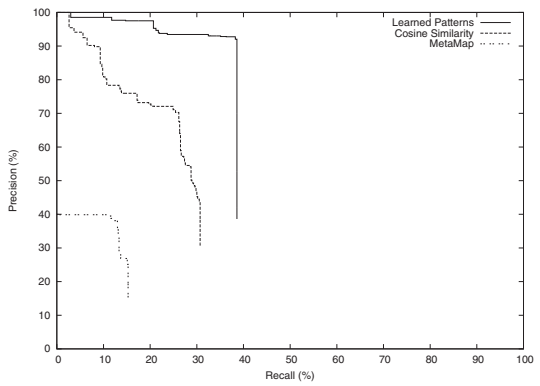
hence for an exact matching system there is only one *F*-measure possible which is shown under the column of best *F*-measure in Table 3.

Rows 4 and 5 in Table 3 show the results when our method, cosine similarity method, and MetaMap are separately used for normalization in cases when exact matching fails. As can be seen (comparing row 4 with row 1 and comparing row 5 with row 3), our method improves accuracies over exact matching while MetaMap and cosine similarity, in fact, lowers the accuracies. This is mainly because those two methods often overly approximate many CUI-less terms and thus end up assigning them CUIs. When considering the output CUIs, our method improved recall by large amount without much affecting the precision, but MetaMap and cosine similarity lowered precision while improving recall.

While our method learned patterns from all the available disease and disorder semantic type terms in UMLS, the method can also learn most of the patterns from a smaller set. Figure 4 shows a learning curve in which we increase the percent of randomly selected UMLS concepts used to learn patterns and measure the best *F*-measure obtained on normalization task using the same test dataset (last row of Table 2). As can be seen, the curve mostly flattens out after around 20% of UMLS concepts showing that the method does not necessarily need the entire UMLS to learn normalization rules. This is because the most common variation patterns for clinical terms are also likely to be present in a smaller subset from which the method can learn them.

In addition to learning normalization rules from a resource like UMLS, our method can also learn normalization rules from a training data of terms found in clinical text and their gold-standard normalized forms. The procedure will be exactly the same, except that it will also consider the terms in the training data and their gold-standard normalized forms for generating edit distance patterns. This will enable the method to learn and adapt to idiosyncrasies specific to the particular genre of the training data or specific to the medical center from where the training data was gathered. This will also enable it to learn common patterns of misspellings if present in the training data which it otherwise cannot learn just from UMLS. However, when we used our method to also learn normalization rules from the training and development data of SemEval 2014 Task 7 in addition to learning them from UMLS, there was almost no impact on the results. This was because our method did not learn any new types of variations of clinical terms from our training data that were not already learned from UMLS. However, we want to point out that this finding is specific to disease and disorder mentions present in the dataset we used, but in general, the method may learn new types of variations from training data which are not learnable from UMLS.

Figure 3: Precision-recall curves for the normalization task on the test dataset of SemEval 2014 Task 7B obtained using the proposed method that uses learned edit distance patterns, using the method based on cosine similarity, and using the MetaMap system.



CONCLUSION

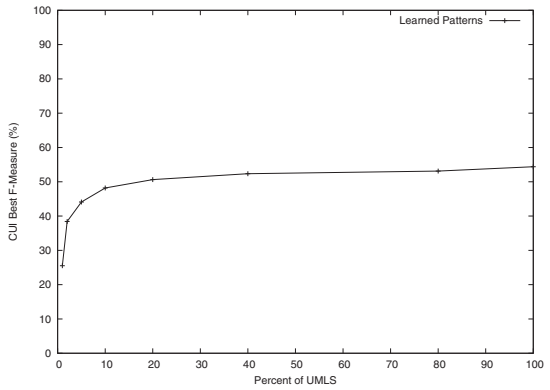
We presented a novel method for automatically learning rules for normalizing clinical terms. The rules are learned from known variations of clinical terms from a resource like UMLS by computing edit distance patterns and then generalizing those patterns. The process of generalization enables the method to capture common variation patterns. The learned rules are then validated within UMLS to estimate their accuracy and to prevent over-generalization. The entire method does not need any human annotation effort. We also presented an efficient algorithm for learning normalization rules that can easily scale to the size of UMLS. Our method is particularly good at capturing morphological and typographical variations. It can also capture variations at the word level. However, because it does not do any semantic analysis, it is not capable of normalizing terms based on their meanings. Using a standard dataset of disease and disorder

Table 3: Results on the test dataset of the disease and disorder mention normalization task of SemEval 2014 Task 7B.

	System or Combination	Accuracy (CUI + CUI-less)	Best *F*-measure (CUI)	Precision at best *F*-measure (CUI)	Recall at best *F*-measure (CUI)
1	UMLS	81.53	83.90	90.71	78.05
2	TD	77.43	81.31	92.79	72.36
3	UMLS + TD	86.16	88.11	92.58	84.06
4	UMLS + patterns	83.02	86.52	90.42	82.94
	UMLS + cosine	70.00	84.19	86.51	82.00
	UMLS + MetaMap	79.49	84.01	88.46	79.99
5	UMLS + TD + patterns	88.53	90.06	92.56	87.69
	UMLS + TD + cosine	85.83	89.06	91.75	86.52
	UMLS + TD + MetaMap	85.47	87.91	90.71	85.28

All the numbers are percentages. The results show performances of exact matching in UMLS, exact matching in the training and development dataset (TD), their combination, and when combined separately with the proposed method that uses learned patterns, with the method based on cosine similarity, and with MetaMap.

Figure 4: Learning curve for the best *F*-measure evaluating the CUI output of the proposed method when it learns normalization rules using increasing number of UMLS terms.



normalization task, we demonstrated that our method works very well and outperforms the MetaMap system and a method based on cosine similarity on this task. Our method is also general enough to learn normalization rules for clinical terms of other semantic types besides diseases and disorders.

CONTRIBUTORS

The author did all the work presented in this paper.

COMPETING INTERESTS

The author has no competing interests to declare.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ACKNOWLEDGEMENTS

We thank the organizers of SemEval 2014 Task 7 for creating and providing the data which was used in this work.

SUPPLEMENTARY MATERIAL

Supplementary material is available online at http://jamia.oxfordjournals.org/.

REFERENCES

1. Bodenreider O. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res.* 2004;32(Suppl 1):D267–D270.
2. Pradhan S, Elhadad N, Chapman W, Manandhar S, Savova G. SemEval-2014 Task 7: analysis of clinical text. In *Proceedings of the Eight International Workshop on Semantic Evaluation (SemEval-2014)*. 2014: 54–62. August 23-24, Dublin, Ireland.
3. Stenzhorn H, Pacheco EJ, Nohama P, Schulz S. Automatic mapping of clinical documentation to SNOMED CT. *Stud Health Technol Inform.* 2009;150:228–232.
4. Lee DH, Lau FY, Quan H. A method for encoding clinical datasets with SNOMED CT. *BMC Med Inform Decis Mak.* 2010;10(1):53.
5. Barrett N, Weber-Jahnke J, Thai V. Automated clinical coding using semantic atoms and topology. In *Proceedings of the 25th International Symposium on Computer-Based Medical Systems (CBMS)*. June 20–22, 2012; Rome, Italy; 2012:1–6.
6. Ramanan S, Adyar C, Nathan S. RelAgent: Entity detection and normalization for diseases in clinical records: A linguistically driven approach. In *Proceedings of the Eight International Workshop on Semantic Evaluation (SemEval-2014)*. 2014;477–481. August 23-24, Dublin, Ireland.
7. Skeppstedt M, Kvist M, Dalianis H. Rule-based entity recognition and coverage of SNOMED CT in Swedish clinical text. In *Proceedings of the International conference on Language Resources and Evaluation (LREC)*. 2012;1250–1257. May 23-25, Istanbul, Turkey.
8. Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. *JAMIA.* 2010;17(3):229–236.
9. Manning CD, Raghavan P, Schütze H. *Introduction to Information Retrieval.* Cambridge University Press; 2008.
10. Tang B, Wu Y, Jiang M, Denny JC, Xu H. Recognizing and encoding discorder concepts in clinical text using machine learning and vector space model. In *Workshop of ShARe/CLEF eHealth Evaluation Lab 2013*. 2013. September 23-26, Valencia, Spain.
11. Zhang Y, Wang J, Tang B, Wu Y, Jiang M, Chen Y, Xu H. A Report for SemEval 2014–Task 7 Analysis of clinical text. In *Proceedings of the Eight International Workshop on Semantic Evaluation (SemEval-2014)*. 2014:802-806. August 23-24, Dublin, Ireland.
12. Leaman R, Doğan RI, Lu Z. DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics.* 2013;29(22):2909–2917.
13. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*. Vol. 10, No. 8, 1966:707-710.
14. Rudniy A, Song M, Geller J. Mapping biological entities using the longest approximately common prefix method. *BMC Bioinformatics.* 2014;15(1):187.
15. Islamaj Dogan R, Lu Z. An inference method for disease name normalization. In *Proceedings of the AAAI 2012 AAAI Fall Symposium on Information Retrieval and Knowledge Discovery in Biomedical Text.* 2012:8–13. November 2-4, Arlington, VA, USA.
16. Bilenko M, Mooney RJ. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.* 2003:39–48. August 24-27, Washington, DC, USA.
17. Pradhan S, Elhadad N, South BR, *et al.* Task 1: ShARe/CLEF eHealth evaluation lab 2013. In *Proceedings of the ShARe/CLEF Evaluation Lab 2013.* 2013:1–6.
18. Pradhan S, Elhadad N, South BR, *et al.* Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *JAMIA.* 2015;22(1):143–154.
19. SNOMED CT. Systematized nomenclature of medicine-clinical terms. *International Health Terminology Standards Development Organization* International release, 2013.
20. Uzuner Ö, South BR, Shen S, DuVall SL. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *JAMIA.* 2011;18(5):552–556.
21. Jiang M, Chen Y, Liu M, *et al.* A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *JAMIA.* 2011;18(5):601–606.
22. de Bruijn B, Cherry C, Kiritchenko S, Martin J, Zhu X. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *JAMIA.* 2011;18(5):557–562.
23. Cestnik B. Estimating probabilities: a crucial task in machine learning. In: *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI 1990)*. 1990:147–149. August 5-10, Stockholm, Sweden.
24. http://en.wikipedia.org/wiki/List_of_medical_abbreviations. Accessed March 19, 2014, Wikipedia.
25. Ghiasvand O, Kate RJ. UWM: disorder mention extraction from clinical text using CRFs and normalization using learned edit distance patterns. In *Proceedings of the Eight International Workshop on Semantic Evaluation (SemEval-2014)*. 2014:828–832. August 23-24, Dublin, Ireland.
26. http://grammar.about.com/od/words/a/comsuffixes.htm. Accessed March 19, 2014, About.com.

AUTHOR AFFILIATION

Department of Health Informatics and Administration University of Wisconsin-Milwaukee Milwaukee, WI, USA katerj@uwm.edu