NCBI at 2013 ShARe/CLEF eHealth Shared Task: Disorder Normalization in Clinical Notes with DNorm

Robert Leaman, Ritu Khare, Zhiyong Lu

National Center for Biotechnology Information, Bethesda, Maryland, USA
(robert.leaman, ritu.khare, zhiyong.lu)@nih.gov

Abstract. We describe an application of DNorm – a mathematically principled and high performing methodology for disease recognition and normalization, even in the presence of term variation – to clinical notes. DNorm consists of a text processing pipeline, including the BANNER named entity recognizer to locate diseases in the text, and a novel machine learning approach based on pairwise learning to rank to normalize the recognized mentions to concepts within a controlled lexicon. DNorm achieved the second highest performance in Task 1a (named entity recognition) and the highest performance (strict accuracy) in Task 1b (normalization). A web-based demonstration of DNorm is available at http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/DNorm/

Keywords: conditional random fields, vector space models, cosine similarity, pairwise learning to rank, MetaMap.

1 Introduction

Concept recognition and identification in clinical notes has many applications, including automated identification of patients at a high risk for complications, automated identification of clinical trial eligibility, and automatic error control in electronic medical records. In this article we describe our approach to the ShARe / CLEF eHealth Task 1a (named entity recognition or NER) and Task 1b (normalization) [1]. We use a machine learning approach, including BANNER, a named entity recognizer utilizing conditional random fields and a rich feature approach [2, 3], and DNorm, a method for normalizing disorder mentions that uses a machine learning model learned directly from the training data [4]. The DNorm model is based on pairwise learning to rank (pLTR), and can represent synonymy, polysemy, and relationships that are not 1-to-1.

1.1 Corpus Description

The corpus provided by the organizers consists of clinical notes of 4 different types and is split into two sets [1]. The Training set contains a total of 199 clinical notes from 4 different types, described in Table 1. The Test set contains 100 clinical notes from 3 out of the 4 types present in the Training set, and is described in Table 2. Notes in the Training set range from about 150 bytes to about 13,200 bytes. The notes

in the Training set total about 9,200 lines of text and 5,900 annotations. The minimum note size in the Test set was 0 bytes, and the maximum size was approximately 14,000 bytes. The Test set contained a total of approximately 8,300 lines of text.

Table 1. Count and average size of each type of clinical note in the Training set

Type of Report	Count (%)	Average size (bytes)
Discharge summary	61 (30.7%)	7,561
ECG	54 (27.1%)	285
Echo	42 (21.1%)	2,235
Radiology	42 (21.1%)	1,941

Table 2. Count and average size of each type of clinical note in the Test set

Type of Report	Count (%)	Average size (bytes)
Discharge summary	76 (76.0%)	7,178
ECG	0 (0.0%)	N/A
Echo	12 (12.0%)	2,246
Radiology	12 (12.0%)	1,717

The Test set was not released until one week prior to results submission, therefore the only information about the Test set available during system development was the number of notes. Our team assumed, however, that the Training set would be representative of the Test set. Comparing the Training and Test sets shows that while the average report sizes for each type are relatively similar, the mix of note types included is different. In addition to the Test set not containing any ECG notes, the percentage of discharge summaries is much higher in the Test set than in the Training set. This increases the overall average note length, since discharge summaries are significantly longer than the other note types.

1.2 Lexicon Description

The lexicon was created using the 2012AB release of the UMLS® Metathesaurus. To comply with the annotation guidelines, the concept identifiers (CUIs) were restricted to the 11 recommended disorder semantic types, and the SNOMED-CT source vocabulary. For each restricted CUI, we computed the non-suppressed English synonyms available in the Metathesaurus, and included those terms in the lexicon.

Furthermore, based on our observations of the Training set, we made several major changes to the lexicon. The Training set contained several mentions annotated as "CUI-less" because the corresponding CUIs lied outside the recommended guidelines, e.g., "left ventricular function" and "unable to walk." We identified the "CUI-less" mentions occurring five or more times in the Training set, and appended those mentions to the lexicon, using the concept ID "CUI-less."

We observed from the Training set that adjective forms were freely substituted for the noun form for many words. While stemming handled many of these cases, many anatomical terms were not handled well: for example, "femoral" is the adjective form

of "femur", and occasionally completely different bases were used, such as "optic" as the adjective form of "eye". We therefore extracted a list of about 60 anatomic adjective / noun pairs from UMLS and added a synonym containing the adjective form for every lexicon name containing the noun form.

The Training set contained several abbreviations that are not found in the Metathesaurus. To address this, we used the Taber's dictionary of medical abbreviations[1]. The Taber's dictionary was filtered to include only those entries where the expanded form exact matched with a synonym of any restricted CUI, and the corresponding abbreviation was included in the lexicon. In all, 102 entries were added to the lexicon.

Finally, we observed that several abbreviation mentions in the Training set required disambiguation, e.g., the mention "AR" matches with the concept "aortic regurgitation" (CUI C0003504) as well as the concept "rheumatoid arthritis" (CUI C0003873), and "CAD" matches with the concept "coronary heart disease" (CUI C0010068) as well as "coronary artery disease" (CUI C1956346). We refined the lexicon to include only one sense of an abbreviation in the following manner. We included only those CUIs wherein at least one term demonstrated evidence of the relationship between short and long forms, e.g., the CUI C0003504 contains the term "AR – aortic regurgitation," and the CUI C1956346 contains the term "CAD – coronary artery disease," i.e., each abbreviation letter matches with the corresponding word's first letter in long form. After applying this pattern rule, some terms still required disambiguation e.g., "MI" matches with "myocardial infarction" as well as "mitral incompetence." We resolved these cases by preferring the sense that appears more frequently in the Training set.

2 Methods

We create two separate systems based on our previous research on disease name recognition and normalization [5 - 7], both of which are described in this section. The first is an application of MetaMap, and is used as a baseline rather than to create our submission for the task. The second system is an adaptation of DNorm to clinical notes, which has previously been applied to the NCBI Disease Corpus [8, 9]. DNorm is a methodology for locating and identifying diseases and disorders mentioned in biomedical text. DNorm uses a pipeline architecture, with modules to perform named entity recognition, abbreviation resolution, and concept normalization (grounding). In this study, we adapt DNorm to clinical notes by dropping the abbreviation resolution module and introducing a post-processing module for boundary revision.

2.1 Sentence segmentation

We segmented each clinical note into sentences using the built-in Java class BreakIterator and manually created rules to correct its output. Examples of the rules we implemented include removing a sentence break after the period in "Dr." and consider-

[1]http://www.tabers.com/tabersonline/view/Tabers-Dictionary/767492/0/Medical_Abbreviations

ing a double newline to be a sentence break. Applying the sentence segmenter to the Training set resulted in about 9,900 sentences.

2.2 MetaMap Baseline

We developed a baseline system using the MetaMap application developed by the National Library of Medicine [10]. MetaMap is a highly configurable system for biomedical named entity recognition and UMLS normalization. Given a textual passage, MetaMap identifies the candidate UMLS concepts and the corresponding spans of the mentions. For this study, we used the MetaMap JAVA API to programmatically access the MetaMap with the following settings. The source vocabulary was limited to the SNOMED-CT, and the semantic categories were restricted to the 11 disorder semantic types as specified in the annotation guidelines.

The baseline system uses the sentence segmentation module described in Section 2.1, the MetaMap API, and a post-processing module. Given a clinical report as the input, the sentence segmenter splits the report into chunks and each chunk is fed into the MetaMap API to obtain the candidate CUIs and spans. For each sentence, the post-processing module validates the candidates in the following manner. The overlapping candidates are resolved using the longest span (or specific mention) criteria, e.g., "breast cancer" is preferred to "cancer." The candidates that require disambiguation, e.g., "heart failure" maps to multiple CUIs, are resolved using the word sense disambiguation module of the MetaMap. In addition, the module filters some generic mentions, e.g., "allergies," "condition," "disease," "finding," etc.

2.3 Named Entity Recognition

The system used to create our submission operates in three steps: named entity recognition, described in this subsection, followed by normalization and boundary revision, which are described in the following two subsections. We used the BANNER named entity recognizer, an open source NER system based on linear-chain conditional random fields and a rich feature set. We used a dictionary feature with diseases from the UMLS Metathesaurus, as in previous work [3]. To reduce overfitting and increase the training performance, we set the labeling model to IO and the order to 1. We created a model that employed different labels for continuous and discontinuous mentions. Mentions tagged by the model as continuous were returned directly, but tokens labeled with the discontinuous mention tag were joined into a single discontinuous mention. This significantly reduced the confusion between continuous and discontinuous mentions, and allowed either 0 or 1 discontinuous mentions to be represented for each sentence. While this is clearly not a complete solution, we found that the majority of sentences with disjoint mentions only contain one.

2.4 Normalization with DNorm

DNorm is a technique for finding the best name from a controlled vocabulary such as SNOMED-CT for a given mention. It first converts both the mention and the names

from the controlled vocabulary to a TF-IDF vector space. It then uses a regression model learned directly from the training data to score each name in the controlled vocabulary against the mention provided as query, and returns the top ranked name.

Vector Space Model. Mentions output by BANNER are tokenized by using whitespace and punctuation as boundaries. Punctuation, whitespace and stop words from the English stop words set in Lucene are removed. Digits are retained, and each token is converted to lower case and stemmed with the Porter stemmer.

We convert the mentions and names to vectors by first defining a set of tokens containing the tokens from all mentions from the Training set and all names from the controlled vocabulary. We then convert both mentions and names to TF-IDF vectors within the space defined by this token set [11]. The TF of each element in the vector is calculated as the number of times the corresponding token appears in the mention or name. The IDF for each element in mention and name vectors is calculated from the number of names in the lexicon that contain the corresponding token:

$$IDF = \log \frac{count(number\ of\ names\ in\ lexicon)}{count(number\ of\ names\ in\ lexicon\ containing\ the\ token) + 1}$$

To correct for the varying lengths of each mention or name, all vectors are normalized to unit length.

Candidate Generation with Ranking. Given the vector space model for mentions and names, normalization can be seen as a ranking task between tuples containing one vector representing a mention (m) and one vector representing a lexicon name (n). Finding the best name can be seen as a scoring task mapping from $\langle m, n \rangle$ onto the set of real numbers. Cosine similarity has typically been used for this purpose, but cosine similarity is not robust to term variations not present in the lexicon. Instead, we can learn a scoring function by introducing a weight matrix, W:

$$score(m, n) = m^T W n = \sum_{i,j=1}^{|\mathcal{T}|} m_i W_{ij} n_j$$

This model allows us to learn both positive and negative correlations between tokens, and is capable of representing synonymy and polysemy. Since our vectors are already unit-length, it is also equivalent to cosine similarity when $W = I$, the identity matrix.

Training DNorm with Pairwise Learning to Rank. We use the training data to learn weights that will result in a higher score for matching pairs $\langle m, n^+ \rangle$ than for mismatched pairs $\langle m, n^- \rangle$. We express this constraint as $score(m, n^+) > score(m, n^-)$, and therefore choose W so that $m^T W n^+ > m^T W n^-$. This is a pairwise learning to rank (pLTR) approach, following [12]. We initialize W to the identity matrix I and optimize via stochastic gradient descent (SGD) [13]. In SGD, a training instance is selected and classified according to the current parameters of the model. If the instance is classified incorrectly, then the parameters are updated by taking a step in the direction of the gradient. We use the ranking loss [14], so that if

$m^T W n^+ - m^T W n^- < 0$, then W is updated as $W \leftarrow W + \lambda(m(n^+)^T - m(n^-)^T)$. The learning parameter λ controls the size of the change to W.

Many concepts have multiple names. Instead of iterating through all combinations of $\langle m, n^+, n^- \rangle$, we instead iterate through all combinations of $\langle m, c^+, c^- \rangle$, where c^+ is fixed as the annotation for m, and c^- is any other concept from the lexicon. Since we intend the best-matching name for c^+ to be ranked higher than the best-matching name for all other concepts, we determine n^+ and n^- as:

$$n^+ = \underset{n \in names(c^+)}{\text{argmax}} \; score(m, n)$$

$$n^- = \underset{n \in names(c^-)}{\text{argmax}} \; score(m, n)$$

2.5 Boundary Revision

We implemented a boundary revision module which uses feedback from the normalization to optimize the NER span tagged. This module considers adding or removing tokens on the left and the right of the span, and uses a manually-constructed set of rules to decide whether to accept the change or not. The boundary revision module adds one token to the left or to the right if the normalization score of the new mention is at least 0.05 above the score for the current mention. Alternatively, the boundary revision module will also add one token to the left if the resulting mention is an exact match for any name in the lexicon. Tokens are not removed from the right, as this tends to delete headwords. Tokens are removed from the left, however, if the best concept for the new mention is the same as the best concept for the old mention, and the difference between the two scores is at least 0.3, which is relatively large.

The boundary revision module also implemented some rule-based post-processing to correctly handle both NER and normalization of several consistent patterns that BANNER was not able to learn. One example is "w/r/r," which is an abbreviation for concepts "wheezing" (CUI C0043144), rales (CUI C0034642), and ronchi (CUI C0035508), though we also observed this abbreviation to be written as "r/w/r" or "r/r/w."

3 Results

We used the official task evaluation measures. These consist of the strict f-measure and overlapping f-measure to evaluate named entity recognition, and strict accuracy and relaxed accuracy for evaluating normalization. We used the definitions provided in the task definition, and used the official scoring script for system evaluation during development. Precision, recall, and F1 measure are defined as follows:

$$p = \frac{tp}{tp + fp} \; r = \frac{tp}{tp + fn} \; f = \frac{2pr}{p + r}$$

where tp is defined as the number of spans that the system returns correctly; for the strict measure, the span returned must match on both the left and the right side, the overlapping measure only requires the spans to have some text in common. Both

measures are micro-averaged. The strict accuracy measure for normalization is defined as follows:

$$strict\ accuracy = \frac{count(correct\ span\ \&\ correct\ concept)}{count(reference\ spans)}$$

This is equivalent to the standard definition for recall if a true positive is taken to be both the span matching exactly and the concept being correctly identified. Mentions marked as "CUI-less" are evaluated as if "CUI-less" were their concept. In other words, the system must return "CUI-less" or the concept will be marked incorrect. The relaxed accuracy is defined as follows:

$$relaxed\ accuracy = \frac{count(correct\ concept)}{count(correct\ span)}$$

Because relaxed accuracy only measures the ability to normalize spans that are correct, it is possible to obtain very high values for this measure by simply dropping any mention with a low confidence span.

3.1 Official Evaluation Results

Our team is listed as TeamNCBI in the official task results. TeamNCBI.1 corresponds to DNorm without boundary revision and TeamNCBI.2 corresponds to DNorm with boundary revision.

Table 3. Official evaluation results for Task 1a (NER), Strict

System	Precision	Recall	F-measure
DNorm, without boundary revision	0.768	0.654	0.707
DNorm, with boundary revision	0.757	0.658	0.704

Table 4. Official evaluation results for Task 1a (NER), Relaxed

System	Precision	Recall	F-measure
DNorm, without boundary revision	0.910	0.796	0.849
DNorm, with boundary revision	0.904	0.805	0.852

Table 5. Official evaluation results for Task 1b (Normalization)

System	Strict	Relaxed
DNorm, without boundary revision	0.587	0.897
DNorm, with boundary revision	0.589	0.895

4 Discussion

Several aspects of the annotations contributed to our results. First, the annotators were instructed to annotate all disorders mentioned, even if not a current concern or not experienced by the patient, and also only annotate disorders that are referenced textually, rather than disorders requiring some inference. These instructions favored an

NER approach based on local textual inference, such as the conditional random field with rich feature set approach used by BANNER. In addition, the annotators were requested to annotate spans that were an exact match for the concept being annotated. In particular, negation is ignored and anaphoric references are not annotated.

There were two primary difficulties we found with our approach based on localized textual inference. First, discontinuous mentions posed a significant difficulty. In addition, there were some annotations that appeared to require inference from the remainder of the clinical note. For example, "aspiration" is sometimes mapped to "pulmonary aspiration" (CUI C0700198) and sometimes to "aspiration pneumonia" (CUI C0032290). Another example is "complications," which was mapped to "complications of treatment" (CUI C0679861) and also to "late effect of complications of procedure" (CUI C0160815). It was not entirely clear, however, whether such examples indicated that the context should be considered or were merely reflections of the difficulty in maintaining annotation consistency. Our methods attempted to learn the most frequent sense based on the localized text, and did not consider the broader context of the clinical note.

5 Conclusion

In conclusion, we have successfully applied our DNorm method for finding disorder mentions to clinical notes. The method uses a pipeline approach to text processing, primarily based on localized textual inference, and learns term variations directly from the training data by applying a learning algorithm based on pairwise learning to rank. We believe that this method may be widely applicable. For future work, we intend to improve our ability both to infer the presence of discontinuous mentions and to condition our normalization inferences on the context present in the remainder of the clinical note.

Acknowledgements

The authors are grateful to the ShARe project (Shared Annotated Resources: Noemie Elhadad, Wendy Chapman, Martha Palmer) for providing the corpus. The authors would like to thank Chih-Hsuan Wei for his help preparing the demonstration website. This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

References

1. Suominen, H., Salantera, S., Velupillai, S. et al.: Three Shared Tasks on Clinical Natural Language Processing. Proceedings of the Conference and Labs of the Evaluation Forum. (2013) To appear.
2. Leaman, R., Gonzalez, G.: BANNER: an executable survey of advances in biomedical named entity recognition. Pac. Symp. Biocomput. pp. 652-663 (2008)

3. Leaman, R., Miller, C., Gonzalez, G.: Enabling Recognition of Diseases in Biomedical Text with Machine Learning: Corpus and Benchmark. Proceedings of the 2009 Symposium on Languages in Biology and Medicine, pp. 82-89 (2009)
4. Leaman, R., Islamaj Dogan, R., Lu, Z.: Disease Name Normalization with Pairwise Learning to Rank. Under consideration
5. Névéol, A., Kim, W., Wilbur, W.J., Lu. Z.: Exploring two biomedical text genres for disease recognition, In Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, pp. 144-152. (2009)
6. Névéol, A., Li, J., Lu. Z.: Linking multiple disease-related resources through UMLS, Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, pp. 767-772. (2012)
7. Islamaj Dogan, R., Lu, Z.: An Inference Method for Disease Name Normalization, Proceedings of the AAAI 2012 Fall Symposium on Information Retrieval and Knowledge Discovery in Biomedical Text, pp. 8-13. (2012)
8. Islamaj Doğan, R., Leaman, R., Lu, Z.: NCBI Disease Corpus: a Richly Annotated Corpus for Disease Name Recognition and Normalization. Under consideration
9. Islamaj Doğan, R., Lu, Z.: An improved corpus of disease mentions in PubMed citations. Proceedings of the ACL 2012 Workshop on BioNLP, pp. 91-99 (2012)
10. Aronson, A.R.: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proceedings of the AMIA Symposium, pp. 17-21 (2001)
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retreival. Cambridge University Press (2008)
12. Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y.J., Chapelle, O., Weinberger, K.: Learning to rank with (a lot of) word features. Inform. Retrieval 13, 291-314 (2010)
13. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. Proceedings of the International Conference on Machine Learning, pp. 89-96 (2005)
14. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Smola, A.J., Bartlett, P.L., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers, pp. 115-132. MIT Press (2000)